The Blessings of Explainable AI in Operations & Maintenance of Wind Turbines

Joyjit Chatterjee, B.Tech, PGDip

Department of Computer Science and Technology
University of Hull

A thesis submitted in partial fulfilment for the degree of
Doctor of Philosophy in Computer Science

September 2021

In loving memory of my Grandmother,
and dedicated to my Mom, Dad and Grandfather . . .
मेरी प्यारी दादी की याद में ,
और मेरे माता , पिता एवं दादाजी को समर्पित ।

Declaration

I declare that the material in this thesis consists of original work which has solely been undertaken by the author. The different strands of work in this thesis have been presented and published in several leading international conferences and journals, both in artificial intelligence as well as wind energy.

An overview of the thesis and the research plan was published in Chatterjee (2020). The literature review in Chapter 2 was published as a review article in Chatterjee and Dethlefs (2021b). The work in Chapter 4 on applying Deep Learning for explainable anomaly prediction in wind turbines was published in Chatterjee and Dethlefs (2020e) and Chatterjee and Dethlefs (2019b); another article on this topic on utilising Reinforcement Learning techniques for offshore vessel transfer planning has been published in Chatterjee and Dethlefs (2020b). The work in Chapter 5 on identifying temporal causal relationships in SCADA data has been published in Chatterjee and Dethlefs (2020c) and Chatterjee and Dethlefs (2020a). The work in Chapter 6 on Natural Language Generation for Operations & Maintenance of wind turbines was published in Chatterjee and Dethlefs (2019a) and Chatterjee and Dethlefs (2020d). Finally, the contents of Chapter 7 on facilitating automated Question-Answering in the wind industry was published in Chatterjee and Dethlefs (2021a) and Chatterjee and Dethlefs (n.d.). All supplementary materials accompanying this thesis in various chapters through GitHub (source code, O&M templates data, domain-specific knowledge graph etc.) developed by the author are provided as open-source resources under the MIT license [1].

Joyjit Chatterjee, B.Tech, PGDip
September 2021

[1] MIT Open Source License: https://opensource.org/licenses/MIT

List of Publications

Journal Papers

- **Chatterjee, J.**, Dethlefs, N., "Automated Question-Answering for Interactive Decision Support in Operations & Maintenance of Wind Turbines", Expert Systems with Applications (in submission).

- **Chatterjee, J.**, Dethlefs, N., "Scientometric Review of Artificial Intelligence for Operations & Maintenance of Wind Turbines: The Past, Present and Future", Renewable and Sustainable Energy Reviews, Elsevier, 144:1-31, July 2021.

- **Chatterjee, J.**, Dethlefs, N., "Temporal Causal Inference in Wind Turbine SCADA Data Using Deep Learning for Explainable AI", IOP Journal of Physics: Conf. Series, September 2020.

- **Chatterjee, J.**, Dethlefs, N. Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines, Wind Energy, Wiley, April 2020.

Preprints

- **Chatterjee, J.**, Dethlefs, N., "XAI4Wind: A Multimodal Knowledge Graph Database for Explainable Decision Support in Operations & Maintenance of Wind Turbines", arXiv Preprint, December 2020.

Book Chapters

- **Chatterjee, J.**, Dethlefs, N., "Deep Reinforcement Learning for Maintenance Planning of Offshore Vessel Transfer", Developments in Renewable Energies Offshore, CRC Press, Taylor & Francis, October 2020.

Conference and Workshop Papers

- **Chatterjee, J.**, “Explainable AI for Intelligent Decision Support in Operations & Maintenance of Wind Turbines”, Proceedings of the European Conference on Artificial Intelligence (ECAI)’s Doctoral Consortium, Santiago, Spain, August 2020.
- **Chatterjee, J.**, Dethlefs, N., “The Promise of Causal Reasoning in Reliable Decision Support for Wind Turbines”, 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) at Fragile Earth: Data Science for a Sustainable Planet Workshop (FEED Workshop, San Diego, California, USA, August 2020.
- **Chatterjee, J.**, Dethlefs, N., “A Dual Transformer Model for Intelligent Decision Support for Maintenance of Wind Turbines”, International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, July 2020.
- **Chatterjee, J.**, Dethlefs, N., “Natural Language Generation for Operations and Maintenance in Wind Turbines”, 33rd International Conference on Neural Information Processing Systems (NeurIPS) at Tackling Climate Change with Machine Learning Workshop, Vancouver, Canada, December 2019.
- **Chatterjee, J.**, Dethlefs, N., “Transparent Deep Learning and Transductive Transfer Learning: A New Dimension for Wind Energy Research”, WindEurope Offshore 2019, Copenhagen, Denmark, November 2019.
- **Chatterjee, J.** and Dethlefs, N., “A Deep Learning Approach Towards Prediction of Faults in Wind Turbines”, Extended Abstract in Northern Lights Deep Learning Workshop (NLDL), Tromso, Norway, January 2019.

Abstracts

- **Chatterjee, J.** and Dethlefs, N., "Transparency, Interpretability and Data Availability: Key Challenges for Tackling Climate Change with AI", Workshop on Data Science in Climate and Climate Impact Research, ETH Zurich, Switzerland, August 2020.

Acknowledgements

This thesis is the end product of three years of research in the area of my interest, and an aspiration to make a small positive contribution to the future of Operations & Maintenance of wind turbines. It is integral to mention that the thesis was developed during the COVID-19 pandemic, which has been a difficult time that affected so many lives – I have lost my own family members during this pandemic which has been an irreparable loss. Pursuing research and completing the thesis successfully during this time would not have been possible without the valuable support of many individuals.

I would like to thank my principal supervisor Dr. Nina Dethlefs for introducing me to the fascinating world of Deep Learning and Explainable AI, and enlightening me on its potential for real-world useful applications. You have been my friend, philosopher and guide in this long journey, and without your valuable guidance, coordination and support, completing this thesis would not have been possible. I would also acknowledge and thank my co-supervisor Dr. Alex Turner for the comprehensive and constructive feedback on my thesis as a technical expert during the panel meetings, which have been an invaluable source of learning. I extend my gratitude to Professor Yiannis Papadopoulos for chairing my panel meetings over the last three years and providing valuable guidance and insights on shaping my research direction and in developing this thesis.

I acknowledge the valuable support from the Offshore Renewable Energy Catapult (ORE Catapult) for providing me data from the Levenmouth Demonstration Turbine to conduct my research. This data has helped me understand the real-world challenges in the wind energy domain and develop AI models to tackle some of these challenges. I would thank Dr. Conaill Soraghan, the then Team Leader, O&M Systems at ORE Catapult, and the data and digitalisation team, past and present at ORE Catapult for their valuable insights and support beyond the data, whenever I have needed it. I am also thankful to ENGIE for making the data from the La Haute Borne wind farm openly available for researchers, which has been useful in this thesis. I thank the Skillwind project for the openly available maintenance action manual in the wind energy domain, which was also utilised in this thesis. I am grateful to the Aura Innovation Centre and the University of Hull for supporting my research with a PhD.

scholarship, without which this journey would not have been possible. I acknowledge the Nudge Global Impact Challenge organised by Nudge, Netherlands which helped me pave my impact plan with a view to create a positive change in the society in line with the theme *Prosperity for All – the Great Reset*. I also acknowledge Clym Stock-Williams, Wind Energy Consultant at TNO for your valuable feedback on my research plan during the Nudge Global Impact Challenge 2020.

I am thankful to the Department of Computer Science & Technology and the Faculty of Science & Engineering, and their administrative and technical staff for their continued help and support over the years. In particular, I would acknowledge Helen El-Sharkawy, Pam Doyle, Lynn Morrell, Sarah Ord and Andrew Hancock. Thank you to other academics I have had the pleasure to speak with during my PhD., in particular Professor John Murray and Dr. Peter Robinson for your support and leadership in the department, Dr. Yongqiang Cheng for your valuable guidance and support and organising the wonderful PGR seminars, Dr. Darryl Davis for your valuable feedback and interesting questions during the Christmas workshops and Professor James Gilbert for supporting me as a part of the Aura Innovation Centre. I would like to also thank the highly supportive people at the Doctoral College, particularly Dr. Briony McDonagh, Dr. Nigel Shaw, Dr. Karen Turner, Jackie McAndrew, Alicia Milson, Arlene Goodwin for all guidance and help during my postgraduate research study. I also acknowledge Dr. Janine Hatter for your valuable support in completing the PGTS.

To all my friends, past and present at the Big Data Analytics Research Group in the Milner lab, thank you for the many lunches we had, Christmas workshops, long insightful discussions and positive thoughts. Also, I acknowledge the wonderful support from my friends at the Turing and Babbage labs. In particular, I would like to thank Koorosh Aslansefat, Eva Sousa, Dr. Annika Schoene, Matthew Hartley, Albert Dulian, George Lacey, Joshua Bee, Luana Mincarelli, Onatkut Dagtekin, Lydia Bryan-Smith and Dr. Ric Colasanti for your friendship and the confidence reposed in me, which never made me feel that I was 7,000 km away from my home in India.

I would like to thank my family for their continued support and trust in me throughout my PhD., and the love and affection at all times, without which it would not have been possible to complete this thesis during a life-changing global pandemic. Finally, I acknowledge the valuable comments and constructive suggestions from my examiners, Dr. James Carroll and Dr. Yongqiang Cheng, which helped to significantly improve the quality and readability of this thesis.

Abstract

Wind turbines play an integral role in generating clean energy, but regularly suffer from operational inconsistencies and failures leading to unexpected downtimes and significant Operations & Maintenance (O&M) costs. Condition-Based Monitoring (CBM) has been utilised in the past to monitor operational inconsistencies in turbines by applying signal processing techniques to vibration data. The last decade has witnessed growing interest in leveraging Supervisory Control & Acquisition (SCADA) data from turbine sensors towards CBM. Machine Learning (ML) techniques have been utilised to predict incipient faults in turbines and forecast vital operational parameters with high accuracy by leveraging SCADA data and alarm logs. More recently, Deep Learning (DL) methods have outperformed conventional ML techniques, particularly for anomaly prediction. Despite demonstrating immense promise in transitioning to Artificial Intelligence (AI), such models are generally black-boxes that cannot provide rationales behind their predictions, hampering the ability of turbine operators to rely on automated decision making. We aim to help combat this challenge by providing a novel perspective on Explainable AI (XAI) for trustworthy decision support.

This thesis revolves around three key strands of XAI – DL, Natural Language Generation (NLG) and Knowledge Graphs (KGs), which are investigated by utilising data from an operational turbine. We leverage DL and NLG to predict incipient faults and alarm events in the turbine in natural language as well as generate human-intelligible O&M strategies to assist engineers in fixing/averting the faults. We also propose specialised DL models which can predict causal relationships in SCADA features as well as quantify the importance of vital parameters leading to failures. The thesis finally culminates with an interactive Question-Answering (QA) system for automated reasoning that leverages multimodal domain-specific information from a KG, facilitating engineers to retrieve O&M strategies with natural language questions. By helping make turbines more reliable, we envisage wider adoption of wind energy sources towards tackling climate change.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AD-DSTCN	Attention-based Dilated Depthwise Separable Temporal Convolutional Network
ADASYN	Adaptive Synthetic Sampling
AI	Artificial Intelligence
AM	Argument Mining
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
AST	Abstract Syntax Tree
BART	Bidirectional and Auto-Regressive Transformer
BERT	Bidirectional Encoder Representations from Transformers
BSTS	Bayesian Structural Time-Series
C4	Colossal Clean Crawled Corpus
CART	Classification and Regression Trees
CBM	Condition-Based Monitoring
CBR	Case-Based Reasoning
CNN	Convolutional Neural Networks

DAE	Denoising Autoencoders
DC GAN	Deep Convolutional Generative Adversarial Networks
DNN	Deep Neural Networks
DQN	Deep Q-learning Network
DRL	Deep Reinforcement Learning
DSL	Domain-Specific Language
DWT	Discrete Wavelet Transform
ELM	Extreme Learning Machine
EMD	Empirical Mode Decomposition
ENGIE	ENGIE La Haute Borne onshore wind farm
EOS	End of sentence
FDI	Fault Detection and Isolation
GDA	Gaussian Discriminative Analysis Models
GLUE	General Language Understanding Evaluation benchmark
GMM	Gaussian Mixture Models
GP	Gaussian Processes
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
GRU	Gated Recurrent Units
GUI	Graphical User Interface
HPC	High Performance Computing
IMF	Intrinsic Mode Functions
IoT	Internet of Things

KB	Knowledge Bases
KG	Knowledge Graphs
KNN	K-Nearest Neighbours
KS-test	Kolmogorov-Smirnov test
LC-QuAD	Large-Scale Complex Question Answering
LDPC	Low-density Parity Check
LDT	Levenmouth Demonstration Turbine
LIME	Local Interpretable Model-Agnostic Explanations
LMedS	Least Median of Squares
LSTM	Long Short-Term Memory
MCKD	Maximum Correlated Kurtosis Deconvolution
MemNN	Memory Networks
MIMO	Multi-Input-Multi-Output
ML	Machine Learning
MLM	Masked Language Modelling
MLP	Multilayer Perceptron
MTTF	Mean Time to Failure
NLG	Natural Language Generation
NLIKB	Natural Language Interfaces to Knowledge Bases
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NWP	Numerical Weather Prediction
O&M	Operations & Maintenance

PCA	Principal Component Analysis
POD	Platform for Operational Data
PUI	Product Use Information
QALD-7	7th Question Answering over Linked Data Challenge
QA	Question-Answering
QC	Question-Cypher query
RAMOBoost	Ranked Minority Oversampling in Boosting
RBFNN	Radial Basis Function Neural Networks
ReLu	Rectified Linear Unit
RL	Reinforcement Learning
RMS	Root Mean Square
RNN	Recurrent Neural Networks
ROI	Return on Investment
RUL	Remaining Useful Life
SARSA	State–Action–Reward–State–Action
SCADA	Supervisory Control & Acquisition Data
Seq2Seq(Att)	Sequence-to-Sequence model with attention
Seq2Seq	Sequence-to-Sequence model
SMOTE	Synthetic Minority Over-Sampling Technique
SVM	Support Vector Machine
SVR	Support Vector Regression
t-SNE	t-Distributed Stochastic Neighbor Embedding
T5	Text-To-Text Transfer Transformer

TPU	Tensor Processing Unit
VAE	Variational Autoencoders
XAI	Explainable AI
xDNN	Explainable Deep Neural Networks
XGBoost	eXtreme Gradient Boosting

Chapter 1

Introduction

> It always seems impossible until it's done.
>
> Nelson Mandela

In the global efforts towards combating climate change and transitioning to a low-carbon economy, there has been a significant growth in developments and deployments of wind turbines globally (Hara, 2016; Pang et al., 2020a). With this fast-paced move, the worldwide capacity of wind power generation has continued to evolve rapidly (Arshad and O'Kelly, 2019; Kaldellis and Zafirakis, 2011; Stetco et al., 2019). Wind energy has several advantages – including being clean and non-polluting, leading to the creation of many jobs, and is cost-effective in the long term (Fauzan et al., 2017). However, wind turbines are complex engineering systems and regularly suffer from operational inconsistencies and failures (Bach-Andersen et al., 2015), leading to unexpected downtimes and significant maintenance costs for turbine operators.

With the presence of several electrical and mechanical sub-components in turbines (gearbox, generator, pitch system etc.), there is frequent prevalence of complex and dynamic loads which lead to inconsistent operational behaviour (Chatterjee and Dethlefs, 2020e; de Novaes Pires Leite et al., 2018). As turbines continue to be deployed in harsh environments under challenging operating conditions (such as offshore) (Hara, 2016; Röckmann et al., 2017), this situation becomes even more challenging, owing to the higher average wind speeds and challenging conditions at sea. Operations and Maintenance (O&M) plays an intrinsic role in tackling such unexpected failures and inconsistent behaviour of turbines, and can ensure timely diagnosis of impending faults. According to Carroll et al. (2016), O&M activities account for around 30% of the levelized cost of energy involved in wind power generation. Condition-Based Monitoring (CBM) (García Márquez et al., 2012; Kuseyri, 2015; Maldonado et al., 2020) is a key aspect in O&M, and has been widely utilised in the wind industry towards monitoring vital operational parameters, especially through application of signal processing techniques and numerical physics-based models (Qiao and Lu, 2015). As developments in computational intelligence and statistical techniques rapidly continue to proliferate, the advent of Artificial Intelligence (AI) techniques, fuzzy logic, Bayesian networks, evolutionary algorithms etc. has made data-driven decision making much promising in domains spanning healthcare, transportation, energy, insurance, banking, e-commerce etc. (Baum et al., 2018; Janev et al., 2020; Tannahill and Jamshidi, 2014).

Wind turbines consist of a variety of sensors which regularly measure vital operational parameters from the turbine and its different sub-components (e.g. gearbox oil temperature, active and reactive power, pitch angle etc.), meteorological information (e.g. wind speed, air pressure etc.) as well as historical logs of alarms with descriptions of their causes in the form of natural language phrases. This is stored in the form of Supervisory Control & Acquisition (SCADA) data (Chatterjee and Dethlefs, 2020d; de Novaes Pires Leite et al., 2018; Tautz-Weinert and Watson, 2017). By optimally leveraging the full potential of such information, O&M of wind turbines can potentially be made much simpler and cost-efficient.

1.1 Data-driven decision making for Operations & Maintenance

The utilisation of field data, also referred to as Product Use Information (PUI) towards data-driven decision making in planning and scheduling maintenance activities for wind turbines has been an area of significant interest to the wind industry (Nabati and Thoben, 2017). The PUI in the case of turbines generally consists of historical records of alarms and SCADA signals for CBM and performance assessment/analysis, providing rich information about the status of the turbine and its sub-components (referred to as *assets*).

It is integral to note that the domain of O&M in the wind industry is vast and diverse, spanning different focus areas for CBM and performance assessment, including time-series forecasting of vital operational parameters (e.g. wind power, wind speed etc.) (Foley et al., 2010), prediction of faults in turbine sub-components (Zhao et al., 2017) and determining maintenance strategies to fix/avert operational inconsistencies (de Novaes Pires Leite et al., 2018). Figure 1.1 summarises the different strands of O&M tasks commonly prevalent in the wind industry, along with their specific applications.

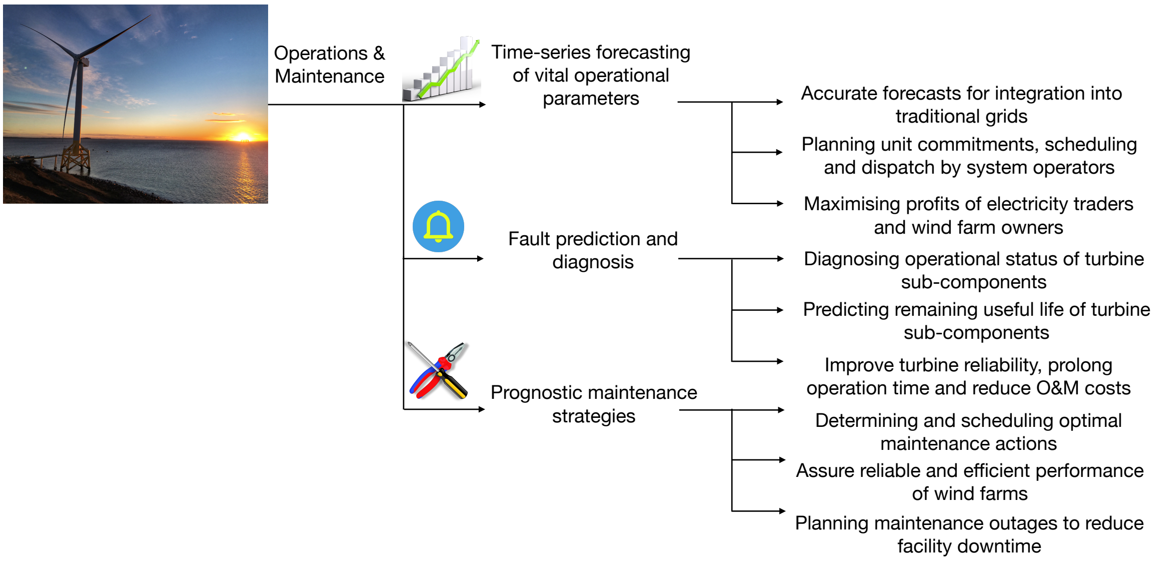

Operations & Maintenance
Time-series forecasting of vital operational parameters
Fault prediction and diagnosis
Prognostic maintenance strategies
Accurate forecasts for integration into traditional grids
Planning unit commitments, scheduling and dispatch by system operators
Maximising profits of electricity traders and wind farm owners
Diagnosing operational status of turbine sub-components
Predicting remaining useful life of turbine sub-components
Improve turbine reliability, prolong operation time and reduce O&M costs
Determining and scheduling optimal maintenance actions
Assure reliable and efficient performance of wind farms
Planning maintenance outages to reduce facility downtime

Fig. 1.1 Summary of various focus areas for wind turbine CBM and performance assessment. The vast and diverse domain of O&M in the wind industry can clearly be inferred.

Several studies in the past have focused on utilising data-driven approaches for decision support in the wind industry using SCADA data, primarily applying conventional Machine Learning (ML) techniques such as decision trees, Support Vector Machines (SVMs) and probabilistic models for anomaly prediction during CBM (Abdallah et al., 2018a; Ge et al., 2017; Xiao et al., 2019). Deep Learning (DL) techniques have also witnessed significant interest, and have mostly outperformed traditional ML techniques based on their predictive performance (Carroll et al., 2018; Chatterjee and Dethlefs, 2020e; Helbing and Ritter, 2018c; Hongshan et al., 2018; Ibrahim et al., 2016; Stetco et al., 2019). More sophisticated DL architectures, such as Recurrent Neural Networks (RNNs) have seen limited attention in the wind industry for anomaly prediction (Lei et al., 2018), with most studies applying Long Short-Term Memory Networks (LSTMs) for performance assessment in predicting vital operational parameters (such as active power, wind speed etc.) (Zhang Jinhua et al., 2019). Despite providing high accuracy in predictions of faults as well as vital operational parameters, existing studies do not focus on inculcating transparency in their decisions and automatically determining strategies towards averting (or quickly fixing) the failures, which is vital for effective decision making. Another interesting vein of research which has recently gained prominence is in combining multiple sources of information (e.g. vibration and SCADA data) into AI-based unified anomaly prediction models (Turnbull et al., 2020). Such techniques have shown immense promise in detection of anomalies with greater consistency compared to traditional approaches which only rely on turbine data sources in isolation.

As described above, some promising advances have been made in existing literature towards utilising AI models for time-series forecasting and fault diagnosis in the wind industry, especially using SCADA data. However, there has been limited attention on developing AI models which are human-intelligible and transparent i.e. that are capable of providing interpretations on *why* and *how* the model makes (or does not make) certain decisions. For complex engineering systems such as wind turbines operating in real-world challenging conditions, there is an urgent need to develop trustworthy decision making techniques for O&M. More importantly, such techniques need to be utilised in conjunction with existing fault prediction methodologies to ensure reliable and efficient decision making in the wind industry. Towards accomplishing explainable decisions, we focus on two key areas in this thesis – **(1)** explainable anomaly prediction in turbine sub-components **(2)** determining appropriate O&M actions to fix/avert the predicted faults.

1.2 Research Problem

The wind industry is presently reluctant to adopt AI-based decision making approaches for O&M, especially deep learners – primarily due to the lack of trust in decisions made by the black-box natured learning models and sometimes the issue of obtaining sufficient amounts of relevant domain-specific information for CBM (Chatterjee and Dethlefs, 2020e; Leahy et al., 2019; Stetco et al., 2019). SCADA data, especially those containing a labelled history of alarm records are often difficult to acquire for pursuing research, given their generally commercially sensitive nature and in cases wherein new turbines are deployed that have not witnessed long periods of operation (Chatterjee and Dethlefs, 2021b). Additionally, it is challenging for turbine engineers and technicians to manually annotate rapidly changing information on operational events and alarms. These resources are rarely fully and efficiently utilised by wind turbine operators and maintenance service providers (Koltsidopoulos Papatzimos, 2020). Conventional ML algorithms (such as decision trees and probabilistic models) provide added transparency in their decisions and are generally simpler to interpret, but are mostly significantly outperformed by DL techniques when the data utilised has a temporal nature (Sezer et al., 2019), which is the case with SCADA data spanning multivariate time-series. Such challenges make it integral to develop deep learners which can predict faults with high accuracy as well as provide transparent decisions, under the constraints of limited data availability in the wind industry. In particular, to ensure trustworthy decisions during O&M, there is a pressing need to develop specialised DL models which can discover novel insights during the learning process (for instance, by predicting the causal relationships in

SCADA features which lead to failures in turbines) as well as quantify the importance of vital parameters that lead to the models' decisions. We hypothesise that discovery of such knowledge can help inculcate transparency and trustworthiness in the generally black-box natured neural networks when these are utilised in the wind industry for CBM. This leads to the first research question (RQ) of this thesis:

RQ1: *How can we develop Deep Learning models tailored for explainable decision support in the wind industry in the face of limited data?*

While existing studies have utilised SCADA data for decision support, they mostly neglect other vital information available including historical alarm records and failure logs. These records, usually referred to as *event descriptions* contain detailed human-intelligible information outlining historical alarms in the form of natural language phrases, types of the affected sub-component (e.g. pitch system, yaw system, gearbox etc.) alongside the exact time-stamp during which the fault occurred in the turbine (Chatterjee and Dethlefs, 2020e; Leahy et al., 2017). The task of generating *informative* natural language messages from turbine SCADA data represents a data-to-text generation problem, wherein, given sequential input data to AI models (continuous valued SCADA features), generation of outputs in the form of descriptions of alarms and human-intelligible maintenance action messages can potentially play an integral role in providing transparent decisions. Such models have also seen immense success in other domains for data-to-text generation, including weather forecasting, spatial navigation etc. (Gong et al., 2019). It is hypothesised that the wind industry can derive significant benefits by leveraging NLG techniques for data-driven decision making towards shortening the time frames for analyses and fixing faults, speeding the pathways towards achieving solutions based on past failures and their context etc. This leads to the second research question of this thesis:

RQ2: *How can we utilise NLG techniques to enhance human-intelligibility in AI models?*

To realise the full potential of recent advances in computational intelligence towards O&M of turbines, wind farm operators need to consider all forms of *useful* data available. Besides SCADA data, there are a variety of other resources available to the wind industry, including maintenance manuals, work orders, turbine sub-component images with details of their prognosis, satellite images from wind farm sites etc. Such multimodal knowledge representations have been utilised in other domains (such as finance and healthcare) towards data-driven decision making in complex scenarios, and have been successful in discovering novel insights and generating informative reports for decision support e.g. through utilisation of domain-specific ontologies (Berant et al., 2013; Hohenecker and Lukasiewicz, 2020). A particularly notable aspect in generation of informative reports in other safety-critical applications like healthcare (Daniel et al., 2019; Goodwin and Harabagiu, 2016) etc. has been through facilitation of interactive decision support via automated question-answering (Chakravarti et al., 2020; Hong et al., 2019) by combining AI models with domain-specific facts – providing automated reasoning during the models' prediction making process. There is a pressing need to develop AI models in the wind industry which can provide transparent decisions beyond quantifying importance of vital SCADA features and predicting brief alarm messages/maintenance actions. In particular, an ambient interface for human-computer interaction is intrinsic for supporting O&M personnel who may not always possess know-how of information retrieval mechanisms like ontologies and knowledge of AI models. To truly benefit from the blessings of explainability in AI models, data-driven decision support systems need to be better accessible to non-specialist users of AI, who can potentially benefit from taking appropriate and timely actions based on their domain-specific expert knowledge in the wind industry. It is hypothesised that facilitation of AI-driven interactive decision support in the wind industry and generation of informative and comprehensive O&M reports by leveraging industry relevant facts (e.g. maintenance manuals, historical alarm logs, SCADA data etc.) can help engineers and technicians better manage (fix/avert) faults and operational inconsistencies in turbines. This leads to the final research question of the thesis:

RQ3: *How can we leverage domain-specific prior knowledge to achieve more informative decisions with AI models?*

The 3 RQs described above present a common challenge of incorporating trust and transparency into fault diagnosis and prognosis models. Explainable AI (XAI) (Barredo Arrieta et al., 2020) is a specialised domain of AI models that can potentially ensure responsible, trustworthy and dependable decision making. XAI can additionally contribute towards improved performance of traditional black-box natured AI models, as explanations can point to various pitfalls and issues in datasets and feature behaviours, while also instilling the blessings of trust and confidence in utilising AI for O&M by engineers and technicians. XAI has seen very limited applications in the wind industry (Chatterjee, 2020), and this research gap motivates us to direct this thesis on XAI for supporting O&M in the wind industry.

Thesis Objectives

Based on the outlined research questions, this thesis has three key objectives – all contributing to the central theme of tackling the challenges in transparency and interpretability during data-driven decision support for the wind industry through XAI. We systematically pursue research in line with these objectives to investigate and evaluate the RQs discussed above. The key research objectives (ROs) of this thesis are:-

- **RO1:** *To develop a tailored AI model for accurate, transparent and scalable anomaly prediction in wind turbines.*
- **RO2:** *To generate O&M strategies in natural language with event descriptions of alarms and suggested maintenance actions to avert/fix the failures.*
- **RO3:** *To leverage multimodal knowledge representations in generating informative O&M reports by facilitating interactive decision support.*

By helping instil trustworthiness and better human-intelligibility in AI models utilised for O&M through the ROs described above, we envisage that unexpected failures in turbines can be better managed (averted or quickly fixed), potentially contributing to reduced turbine downtimes and lower O&M costs for wind farm operators. We are optimistic that these objectives can play a small, positive role in helping make wind energy a more attractive source of renewable energy for wider adoption by organisations globally towards tackling climate change.

1.3 Contributions found in the thesis

There are a number of novel contributions made in this thesis – these are directly based on the research questions and objectives in Section 1.2. The key contributions are described below:-

1. **Explainable anomaly prediction in wind turbines and offshore vessel transfer planning**
    - A novel hybrid AI learning model has been developed for explainable anomaly prediction in an offshore wind turbine, utilising a combination of LSTM for accurate prediction of faults and a XGBoost-based decision tree model to achieve transparency in identifying the key SCADA features which contribute to the fault. The LSTM-XGBoost model is able to predict faults in the offshore wind turbine with an accuracy of 96.6% and F1 score of 0.92.
    - The proposed anomaly prediction model has further been extended to facilitate anomaly prediction in an unseen dataset for an onshore wind farm lacking labelled historical failure data to train the AI model, through transfer learning. The model achieves an accuracy of 65.42% and F1 score of 0.719 in predicting faults in the new target domain.

- Deep Reinforcement Learning (specifically, Deep Q-learning) has been utilised to develop an offshore vessel maintenance transfer planning system based on the types of predicted faults and their priorities, weather conditions etc.

2. **Causal inference and natural language generation for explainable decision support**

    - A temporal causal inference technique based on Attention-based Dilated Depthwise Separable Temporal Convolutional Networks (AD-DSTCN) has been developed to discover novel insights and hidden confounders in SCADA data giving rise to the faults predicted by AI models, and generate intuitive visualisations of these relations in the form of temporal causal graphs.
    - A two stage data-to-text generation system has been developed using a Transformer NLG model to generate human-intelligible natural language phrases for event descriptions of alarms and suggest maintenance actions to avert/fix the faults. The proposed model is able to predict alarm messages with an accuracy of 96.76% and F1 score of 0.978. The model correctly generates relevant maintenance action strategies in 75.35% cases.

3. **Leveraging multimodal knowledge representations for interactive decision support**

    - We propose the development of a novel AI-based expert system to facilitate interactive decision support in O&M of wind turbines. The essence of our methodology lies in developing an ambient interface for the automated retrieval of domain-specific information from a Knowledge Graph (KG).
    - We utilise multimodal domain-specific information in the wind industry (like SCADA features, alarm messages etc.) and integrate it with industry-relevant facts extracted from a repository of documents relating to O&M in wind turbines.
    - The multimodal KG is integrated with a Transformer formal language generation model to develop a full-fledged Question-Answering (QA) system for the wind industry. In this system, engineers can perform information retrieval from the KG in natural language without any need for specialised skills and understanding of graph query languages. Given natural language queries, the Transformer model automatically generates the corresponding Cypher code for querying the KG with an accuracy of 89.75%. The generated code is in turn utilised to retrieve

corresponding maintenance actions and insights from the KG database, providing an efficient interface for automated reasoning.

1.4 Structure of the thesis

The chapters in this thesis are arranged in a logical manner to facilitate coherent understanding of different strands of XAI for O&M of turbines, and can thereby be read linearly. However, the chapters are written in a manner so that they can also be read independently. The thesis is organised as follows:-

Chapter 2 provides an in-depth review of existing literature, summarising the key challenges in O&M of wind turbines, CBM techniques for decision support applied in the past, and the present state of the art. This chapter also discusses literature on various AI algorithms which are utilised in the later chapters. In Chapter 3, detailed information is provided regarding the datasets from real-world operational turbines utilised in this thesis, along with their exploratory data analysis and the steps for pre-processing. The data described in this chapter are used for experimentation in the forthcoming chapters, and are clearly labelled and referenced wherever applicable. Chapter 4 explores the promise of DL for data-driven decision support in turbines towards explainable anomaly prediction and offshore vessel transfer planning. In Chapter 5, temporal causal inference is performed in SCADA data to aid explainability in decisions made by the black-box natured DL models. Chapter 6 explores the application of NLG techniques for O&M. In Chapter 7, the development of an automated QA system based on multimodal domain-specific O&M information is discussed for facilitating interactive decision support. Finally, Chapter 8 concludes the thesis. This chapter discusses the strengths and limitations of the thesis, and provides possible directions for future work.

Chapter 2

Literature Review

> Research is to see what everybody else has seen, and to think what nobody else has thought.
>
> Albert Szent-Gyorgyi

This chapter is based on an article previously published by the author in the journal of *Renewable and Sustainable Energy Reviews* (Chatterjee and Dethlefs, 2021b).

2.1 Introduction

Condition-Based Monitoring (CBM) is a vital aspect of Operations and Maintenance (O&M), and plays an integral role in identifying operational inconsistencies prevalent in various sub-components of wind turbines (Stetco et al., 2019). CBM techniques span areas pertinent to fault detection, fault diagnosis and fault prediction/prognosis (Leahy et al., 2019), and facilitate early detection of degradation or any potential incipient failures through condition-based maintenance before they can cause significantly costly failures in the turbines. Besides this important role, CBM also helps to keep healthy turbines in continued operation by reducing outage events which can occur due to redundantly scheduled maintenance activities (Kuseyri, 2015). Alongside CBM for turbine control, diagnosis of failures and their prediction, there are some other types of focus areas relevant to O&M planning and performance assessment/analysis of turbines necessary to minimise energy costs (Charabi and Abdulwahab, 2020). These areas pertain to forecast and prediction-making for vital operational parameters in operations of turbines (such as wind speed, power factor, torque, power etc.) (Merizalde et al., 2019). All these different types of activities play a vital role in ensuring efficient O&M, especially for wind power systems deployed offshore, given the nature of the multifaceted systems and the harsh conditions in the environments wherein they generally operate (Lin and Liu, 2020).

The key focus of existing studies in the last decade has mostly spanned the utilisation of signal processing techniques and physics-based numerical models for the purpose of CBM pertaining to health monitoring of turbines, especially by utilising vibration data. More recently, there has been an increasing interest in adopting data-driven decision making solutions for CBM (Maldonado et al., 2020; Stetco et al., 2019) and performance assessment/analysis of turbines (Charabi and Abdul-wahab, 2020; He and Kusiak, 2018). Artificial Intelligence (AI) techniques have been successfully utilised in the wind industry for decision making, by learning from Supervisory Control & Acquisition (SCADA) data which are regularly generated through various sensors in turbines (Peharda et al., 2017; Tautz-Weinert and Watson, 2017; Yang and Jiang, 2011). While in some domains such as healthcare and finance, AI has played a pivotal role as a game-changer for data-driven decision making (Tan et al., 2016; Wang and Blei, 2019), the wind industry has not seen as significant benefits from applying AI, especially Deep Learning (DL) techniques. This is likely due to the lack of clear perspective in utilising such models and the limited trust and confidence in their decisions. Given the multitude of directions which span the AI domain, including supervised learning, unsupervised learning, reinforcement learning techniques etc., a comprehensive analysis of existing literature in utilising AI for data-driven decision support is integral for the wind industry.

To analyse and discover patterns prevalent in existing publications, bibliometrics is a family of statistical analysis techniques which has been commonly utilised in the domain of library and information sciences. For such analysis specific to scientific literature, scientometrics, a sub-field of bibliometrics has witnessed growing interest. While several domains including medicine and finance (Kokol et al., 2020; Merigo and Yang, 2014) have seen applications of bibliometrics (and scientometric techniques), there has been limited application of such analytical techniques in the wind industry. The wind energy domain is highly complex – with the prevalence of widely varying techniques for CBM and performance assessment/analysis, variations in data utilised and differences in the nature of tasks performed. This phenomenon is depicted in Figure 2.1, with network visualisation of publications in the domain of data-driven decision making for the wind industry in the last decade.

Our goal in this chapter is to optimally harness statistical computing techniques for scientometrics, to comprehensively evaluate and analyse the applications of AI prevalent

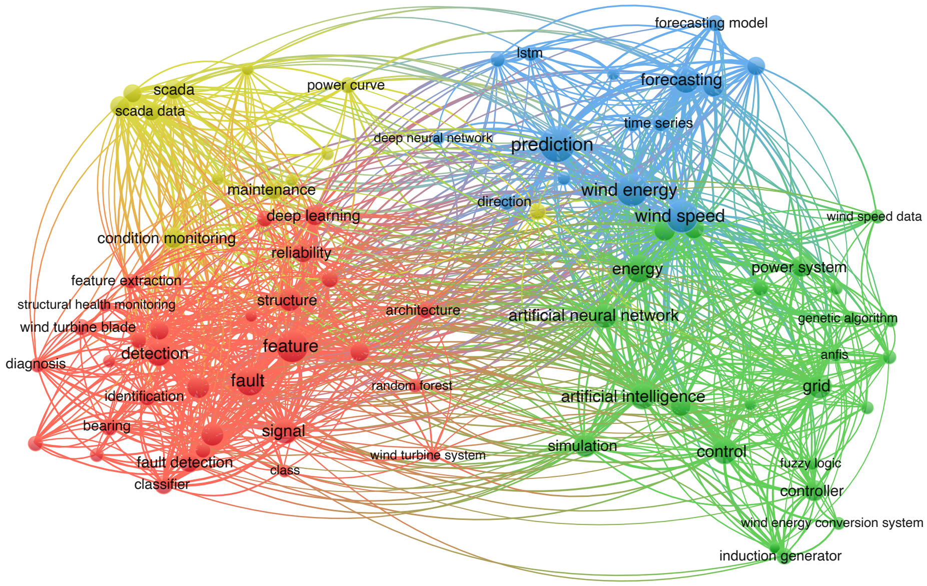


Fig. 2.1 Network visualisation outlining the complex domain of data-driven decision making in the wind industry based on publications from 2010-2020. The various colours depict specific clusters related to each keyword, wherein, a strong association exists amongst terms from similar clusters.

in the wind industry for data-driven decision making, with a primary focus on CBM but also considering and including relevant O&M publications in the area of performance assessment/analysis wherever relevant. For this purpose, we utilise Bibliometrix (Aria and Cuccurullo, 2017), a specialised statistical computing library in R, alongside Citespace (Chen, 2006), a Java application generally utilised to analyse scientific literature and VOSviewer (Eck and Waltman, 2009), a specialised software tool used to visualise complex bibliometric networks for deriving insights on the prevalent conceptual and thematic organisation of existing literature in data-driven decision making for O&M of turbines. Alongside these specialised tools for our literature review, we use Datawrapper (Lorenz et al., 2012), a data analytics and visualisation tool available as a web application for development of insightful visualisations. Through the proposed technique, the systematic literature review conducted can help derive novel insights and knowledge on the evolving nature of AI in the wind energy domain, providing a knowledge taxonomy for varying research themes in this domain. This chapter will also provide a perspective on pressing issues for data-driven decision making in the wind industry at present, such as lack of transparent decisions made by AI models, deploying models for the purpose of real-time decision support and data availability and quality ensurance, along with possible strategies to overcome these major challenges.

This chapter is organised as follows: Section 2.2 describes the bibliographic dataset utilised and its pre-processing. Scientific mapping and analysis for deriving insights on AI for data-driven decision making is discussed in Section 2.3. A perspective on the prevailing challenges which the wind industry is presently facing (and will likely continue to face) in the future is presented in Section 2.4. A likely viable roadmap for adopting AI in the wind industry in the next few years is provided in Section 2.5. Finally, Section 2.6 concludes the chapter and discusses the path for focusing on Explainable AI in this thesis in the later chapters.

2.2 Dataset description and pre-processing

As an initial step for retrieving the data required for performing our literature review, we queried the Web of Science Database [1] for all papers which pertain to CBM in the wind industry prevailing in the last decade (from 2010 to present), leading us to an initial record consisting of 818 publications. From these records, we eliminated the review articles from consideration as we were mainly interested in identifying literature pertaining to specific topics rather than focusing on broad perspectives to ensure robustness. The search for the dataset was refined further, by only retaining the publications from either journals or conference proceedings which are in English language. Note that a few non-relevant publications which did not have any relationship to wind turbine CBM were manually eliminated at this stage. Finally, the overall dataset obtained consisted of 734 records which pertain to O&M of wind turbines, and focus on several techniques (e.g. signal processing and vibration analysis etc.), besides the studies applying AI.

With the key motivation for our literature review being identification of the publications which are specifically focused on applying AI to data from wind turbines, we specified an additional logical criteria to segregate the publications which utilise AI techniques for CBM as well as towards performance assessment/analysis. This led us to 422 records, which mainly consist of papers relevant to CBM but also include few instances of performance assessment/analysis tasks in O&M. The logical criterion which were utilised for retrieval of the publications are described in Table 2.1. We exported all these records into plain text files, which can further be utilised to perform scientific mapping and analysis, which is discussed further in the following sections of this chapter. The dataset utilised for the scientometric review is publicly available on GitHub[2] .

[1] Web of Science Database: https://apps.webofknowledge.com/

[2] Scientometric Review Data: https://github.com/joyjitchatterjee/ScientometricReview-AI

Table 2.1 Summary of the logical queries utilised alongside the number of records retrieved for performing scientometric analysis

Domain	Logical Query (Inclusion criteria)	No. of Retrieved Records after Filtering
All papers relating to CBM	("wind turbine" AND "condition monitoring") OR ("wind energy" AND "condition monitoring")	734
Papers specifically relating to AI for O&M (Includes publications pertaining to CBM and performance assessment/analysis)	("wind turbine" AND "machine learning") OR ("wind energy" AND "machine learning") OR ("wind turbine" AND "deep learning") OR ("wind energy" AND "deep learning") OR ("wind turbine" AND "artificial intelligence") OR ("wind energy" AND "artificial intelligence") OR ("wind turbine" AND "AI") OR ("wind energy" AND "AI")	422

2.3 Performing scientific mapping and analysis

2.3.1 Signal Processing and Vibration Analysis techniques prevalent in the past

We consider the period from 2010 to 2015 across the publications as the past for performing the systematic literature review [3]. This period was utilised after carefully considering the mostly prevalent consensus pertaining to current literature, wherein, any period which falls beyond the last five years is generally considered out of scope (Bloomberg and Volpe, 2016). We were primarily interested in identifying traditional methods (other than those based on AI) applied for CBM and performance assessment/analysis in the past. We observed that after 2015, there was very limited focus on applying signal processing and vibration analysis methods as the wind industry experienced a dynamic shift towards applying AI for data-driven decision making. This was another key reason which led us to consider the period till 2015 as the past, and we will consider the AI-based publications subsequently in this chapter when analysing the present. Considering the past literature is integral as it can help derive insights for prediction and analysis of future trends on the basis of strengths

[3] Note that our scientometric review will mainly focus on papers published in this period. However, for ensuring thoroughness, we will also provide mentions of some notable studies relevant to the scope of O&M in this chapter that may fall outside of this time period.

and weaknesses which existing studies have demonstrated. An interesting observation for this period was that the majority of influential publications had a primary focus on CBM for health monitoring of turbines and their sub-components, with a negligible focus in the area of performance assessment/analysis tasks (such as for forecasting turbine power). Thereby, our discussion and analysis of the past would be mainly directed towards CBM for facilitating a comprehensive analysis, but in relevant cases, we would also include few papers which focus on performance assessment/analysis.

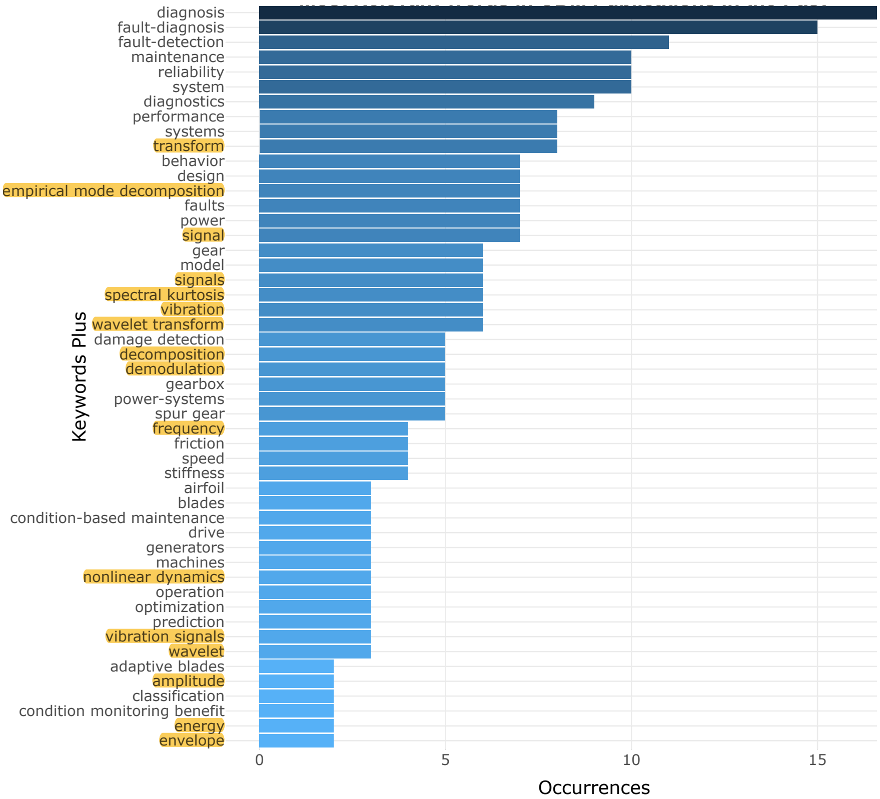


Fig. 2.2 Word frequency for the top-50 words pertaining to wind turbine CBM in the past. The words with relevance to the signal processing domain are highlighted.

From the total records of 734 publications, 311 records were part of this analysis period. The frequency of most common words which were prevalent in CBM-based studies in this period of analysis are shown in Figure 2.2. It should be noted that these words were determined automatically based on the *Keywords Plus* metric [4] (Zhang et al., 2015), which considers the author-provided keywords in the publications alongside the titles and abstracts, as well as the references for identifying relevant content which could have been potentially overlooked if only utilising the author listed keywords. This method thereby helps expand

[4] Keywords Plus: http://interest.science.thomsonreuters.com/content/WOKUserTips-201010-IN

the search scope for analysis of all relevant papers during this time period. Of all identified keywords, not all keywords were relevant to the domain of signal processing and vibration analysis according to their semantic definition e.g. classification. The relevant keywords were manually highlighted as per their occurrence in prevailing domain-specific papers. For instance, the keyword *frequency* was considered to be within scope of the signal processing domain as it occurred in the abstract of a relevant paper (Feng et al., 2012), in which the authors performed vibration signal processing. We adopted a similar approach in highlighting other such keywords according to relevant publications. As can clearly be visualised, there are multiple keywords which fall in the domain of signal processing (such as empirical mode decomposition, wavelet transform etc.) as well as vibration analysis (such as amplitude, vibration signals etc.). To provide further discussion on the prevalence of these techniques in the past, we will focus on some of the papers which are most relevant in this period.

Vibration analysis techniques have been widely utilised for diagnosing faults in turbine structures and their sub-components, with an especially common focus on monitoring rotational parts (Liu, 2013). Zimroz et al. (2012) have previously utilised data consisting of Root Mean Square (RMS) of vibration acceleration signal and generator power obtained via a professional monitoring system to perform vibration analysis and diagnose abnormalities in turbine bearings under conditions of non-stationary load/speed, by decomposing the relevant data for CBM into different sub-ranges of loads. They also utilised these parameters as features for determining operational inconsistencies through statistical signal processing. In a similar study, Liu (2013) proposed the application of physics-based techniques during vibration analysis for statistically estimating the total wind force which prevails in the wind turbine's blade-cabin-tower system, which was described through a mathematical framework, although there was no utilisation of data for demonstrating the method's applicability. Their paper focused on deriving kinetic equations and identification of the natural frequency in the coupling system by utilising Fourier transform alongside other probabilistic methods. Their proposed technique can be integral in the identification of random wind vibrations and the effects which they cause as per the analysed spectrum, helping support fault diagnosis.

Given the non-stationary and non-linear nature of most SCADA features/time-series parameters (Chatterjee and Dethlefs, 2020e), there has been a significant focus on applying statistical signal processing techniques to such metrics. Several studies (Feng et al., 2012; Li, 2010; Yang et al., 2011) have utilised vibration signals and applied Empirical Mode Decomposition (EMD) methodology towards prediction of incipient faults in the mechanical and electrical sub-systems of wind turbines, by performing decomposition of these signals into multiple Intrinsic Mode Functions (IMFs). Feng et al. (2012) have previously utilised an ensemble EMD method, wherein, they performed demodulation analysis of the gearbox vibration signals while accounting for the IMFs generated. By comparison of the demodulated signal envelope's amplitude and instantaneous frequency with ideal theoretical values across the prevailing Fourier spectra, gearbox operational abnormalities, including specific wear and chipping faults can be identified. There have been some variations of this technique utilised for short-term forecast of vital operational parameters in turbines, such as wind speed and wind power. For instance, Zheng et al. (2013) utilised historical wind farm data consisting of wind speed, wind direction and output power of turbines. The authors utilised EMD to decompose the wind power into multiple IMF components and one residue, besides utilising a Radial Basis Function Neural Network (RBFNN) as a prediction model. This study also utilised statistical control algorithms such as Kalman filtering for the purpose of eliminating noise.

There have been some studies which have utilised signal processing techniques and applied them to vibration signals towards estimating Spectral Kurtosis (SK) in the frequency domain for CBM. The kurtogram can play an integral role in this technique, by helping determine non-stationary instances in the signals which potentially lead to defects in any turbine sub-component. The SK methodology plays a vital role to facilitate the extension of the general concept of kurtosis (which is, by nature a global value) to a function of frequency that can indicate impulsiveness in the signals (Saidi et al., 2015). In a notable study within this domain, Saidi et al. (2015) proposed a specialised square envelope methodology based on SK to diagnose skidding in high-speed shaft bearings by performing degradation analysis towards run-to-failure testing. The paper utilises real-world vibration signals from high-speed shaft bearings, and showcases that the SK's maximum value can help provide an indication of the prevailing damage's severity, while the SK's square root is able to extract the signal's transient components. The authors in this study performed experimental runs across different types of cases which pertain to normal zone, degradation zone and failure zone, and the study outlines the immense capability of the SK to diagnose faults across critical parts of rotating sub-components present in turbines.

Owing to the fact that vibration signals are generally subject to high background noise effects, Jia et al. (2015) proposed an improved technique in comparison to the conventional SK method for detecting faults in the rolling-element bearings, by application of Maximum Correlated Kurtosis Deconvolution (MCKD), which can play a useful role in clarifying the periodic fault transients across noisy signals, providing more suitable signals for diagnosing incipient faults. While these techniques are simple to apply and do not require historical fault data for development of the fault prediction models, there are no performance metrics (such as accuracy) presented in these studies, making the predictions incapable of being validated and thereby, such methodologies less robust. In addition, such techniques cannot be applied for determining vital metrics during O&M of the turbine, like Mean Time to Failure (MTTF), Remaining Useful Life (RUL) etc.

Other than the conventional Fourier transform technique, there have been some studies which have utilised wavelet transform towards CBM for analysing vibration signals in the frequency domain (Yanping Guo et al., 2010). Yanping Guo et al. (2010) have, for instance, utilised the Discrete Wavelet Transform (DWT) methodology to determine prevailing faults in the gear based on characteristics of the vibration acceleration signal. In comparison to traditional methods for signal processing, the DWT can play a better role in characterising time-varying segments of the signals and their energy distributions, which can be integral in making fault identification much easier. There has been a similar study by Yang and An (2013), wherein, the authors proposed a hybrid technique which combines EMD with wavelet transform methodology. In this paper, wavelet transform supports analysis of the vibration signals and EMD plays its part in contributing to better decomposition of the prevalent signal into its corresponding IMF components. This process helps in facilitation of more thorough predictions for the instantaneous frequency of the signal, as it can address the challenges of aliasing resulting from interference due to presence of high-frequency components in the transformed signal. Despite being simple for analysis of signals towards detecting faults, such techniques utilising wavelet transform can often be highly computationally expensive and intensive during fine-grained analysis in comparison to AI algorithms, and also generally require careful consideration and choice of parameters for shifting, scaling etc. to achieve any potential success (Sovic and Sersic, 2012).

The hierarchical composition of signal processing and vibration analysis techniques utilised in the past is shown in Figure 2.3. The treemap outlines combinations of different possible keywords in this area (e.g. *empirical mode decomposition* co-occurred as a keyword along with other keywords like *system*, *design*, *behaviour* etc., the *spectral kurtosis* keyword co-occurred alongside others such as *transform*, *spur gear*, *drive* etc. in the majority of cases for the papers in this domain). This visualisation is in line with the reviews presented above, and provides a low-level analytical view. As can clearly be seen, e.g. EMD has been widely utilised towards prediction of faults, forecasting turbine power output and design of turbine control systems etc. Another prevalent method is SK, which has been successfully utilised for modelling SCADA signals characterising turbine sub-components, especially the gearbox. Techniques focusing on demodulation have been utilised widely during signal processing in O&M for vibration signals. Besides this, other relationships outline the notable

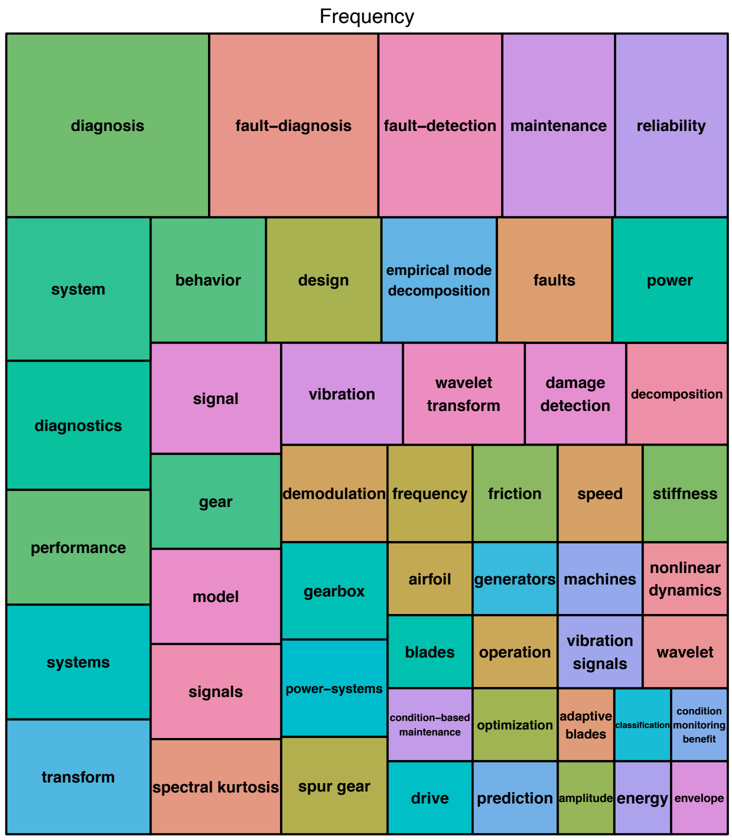


Fig. 2.3 Treemap describing hierarchical composition and structure of prevailing signal processing and vibration analysis methodologies during the past

entities in applying signal processing and vibration analysis prevalent in the past. For the purpose of detecting damage in turbine sub-components, there have been multiple focus angles – such as fault prediction, optimising operations and preventing friction etc. A further fine-grained analysis is shown in Figure 2.4, with the depiction of various clusters for the prevalent techniques alongside their common applications.

It is integral to mention that while signal processing and vibration analysis techniques have mostly dominated the past for CBM, there have been some important studies which have demonstrated the promise of utilising conventional AI methodologies prevalent in the past for the purpose of performance assessment and analysis of turbines. Clifton et al. (2013a) utilised aerostructural simulation data from a turbine and applied regression trees for forecasting wind turbine power output, by accounting for parameters such as wind speed, turbulence and shear. Their technique demonstrated immense promise and success towards forecasting the performance of turbines at new sites, by simply utilising wind resource assessment data which is generally available to turbine operators easily. There have been

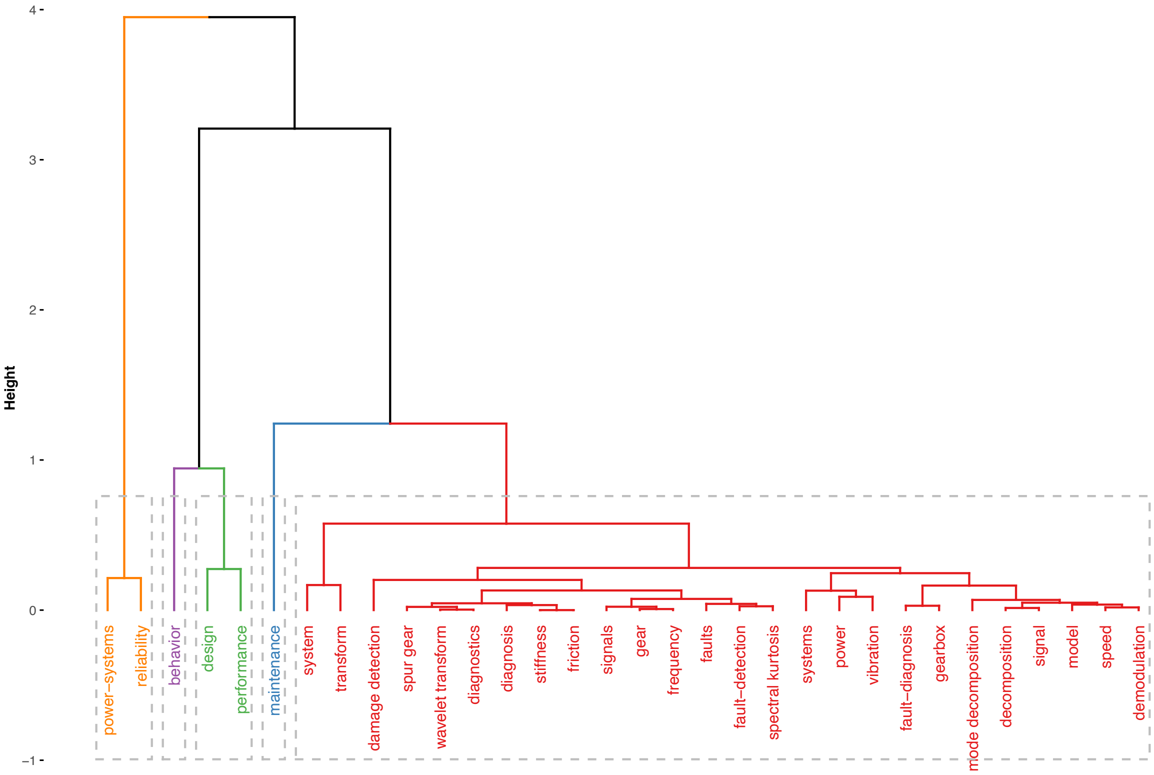


Fig. 2.4 Dendogram outlining clusters and the relevant context towards application of CBM techniques during the past

several other studies which have modelled turbine power outputs by using conventional AI methodologies, like time-series clustering for power forecasts across periods of normal and anomalous operational conditions (Pravilovic et al., 2014), Support Vector Machine (SVM) enhanced Markov modelling (Yang et al., 2015), Numerical Weather Prediction (NWP) models and Gaussian Processes (GPs) (Chen et al., 2014) etc. Some attempts have been made for achieving improved results in power forecasting through application of hybrid models. For instance, Soleymani et al. (2015) utilised a dataset from a real-world wind farm and applied probabilistic approximation techniques in conjunction with conventional AI optimisation techniques to develop a hybrid modified firefly algorithm, which can generate predictions of turbine power outputs, while also taking into consideration the confidence intervals for the predictions. Such techniques are moreover, simple in terms of their application and interpretation, owing to their dependence on probabilistic and statistical inference mechanisms during the prediction process. However, as later discussions would make it evident, these techniques are often significantly outperformed by other approaches in recent times, especially those utilising DL for time-series forecasting.

It is interesting to note that there have been very limited applications of AI techniques in the past towards prediction of faults in turbines and their sub-components. It is thereby integral to carefully consider the trade-off for installation of additional sensors in the turbine for vibration analysis and utilisation of SCADA data for achieving optimal results during data-driven decision making.

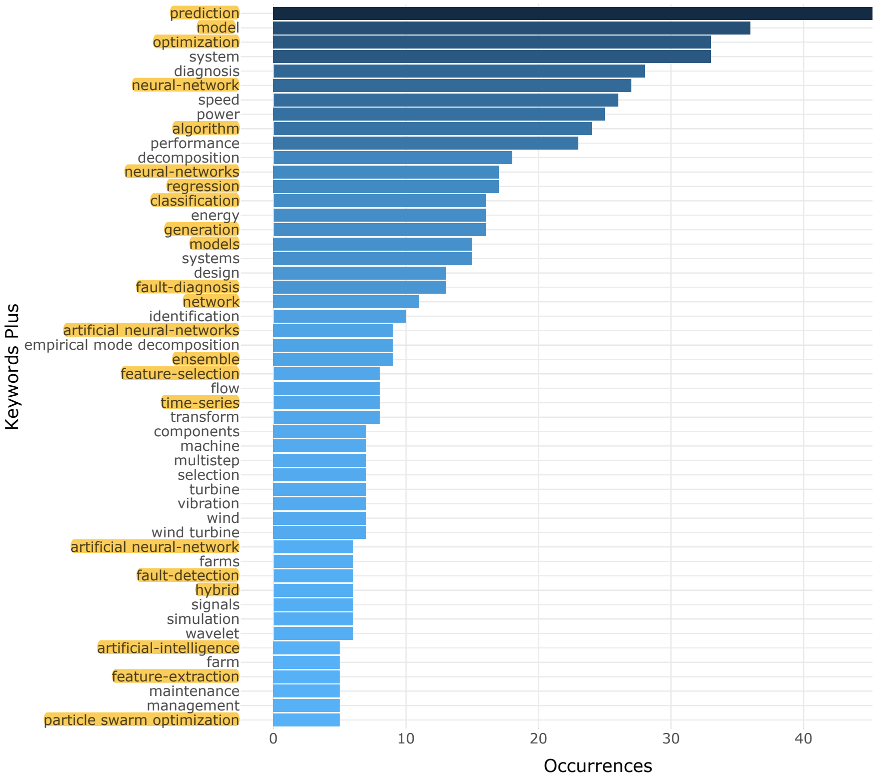


Fig. 2.5 Word frequency for the top-50 words pertaining to wind turbine CBM at present. The words with relevance to AI techniques are highlighted.

2.3.2 Rise of AI in the wind energy domain at present

Numerical modelling and signal processing have been the most popular techniques in the past for CBM, wherein, there has been a particularly dominant focus on leveraging vibration signals for the purpose of turbine health monitoring with promising results. There have been some instances wherein AI techniques have been utilised with SCADA data for the purpose of performance assessment/analysis. However, these are very rare compared to the focus which has existed on utilisation of vibration data towards CBM. Data-driven decision making techniques that utilise historical SCADA data for training AI algorithms

(Tautz-Weinert and Watson, 2017) are often much simpler and cheaper to apply. Also, as present-day wind turbines are generally fitted with multiple sensors which regularly measure vital parameters that are part of SCADA data as standard components, there are rarely any additional requirements for installation of measuring/instrumentation systems and devices (Yang et al., 2014).

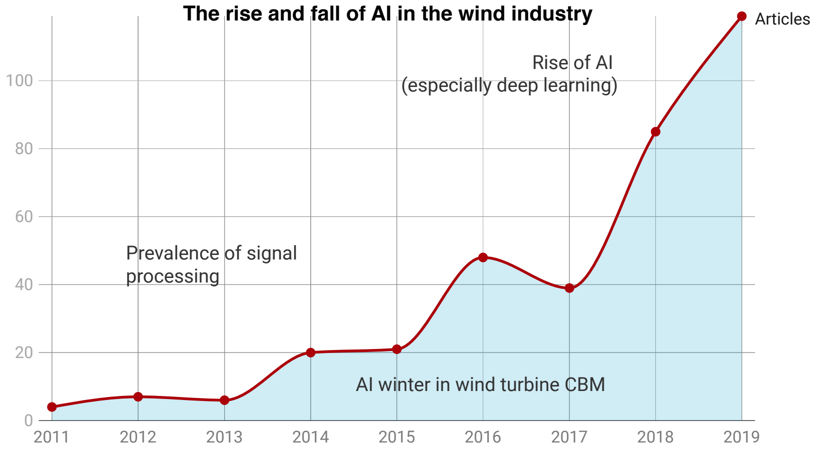


Fig. 2.6 Visualisation of the evolving nature of AI in the wind industry. Post-2017, the significant rise in interest towards utilising AI for CBM is clearly evident.

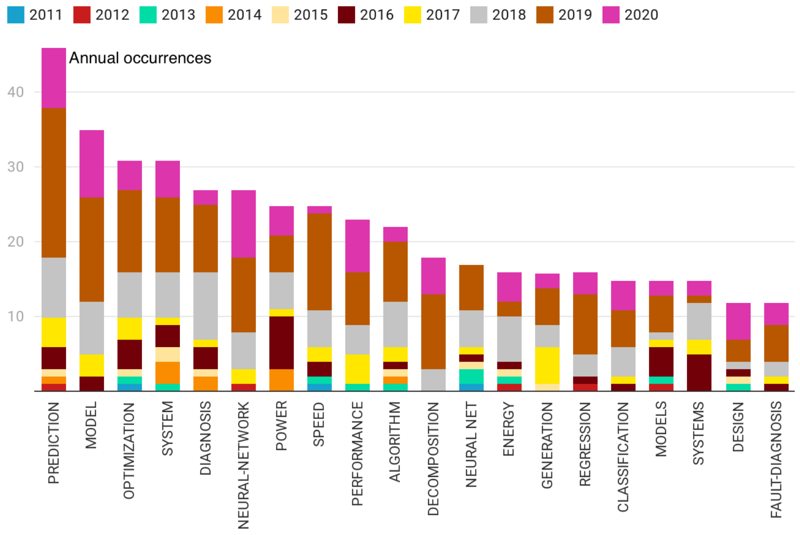


Fig. 2.7 Word dynamics of keywords across CBM publications per year. The growth in utilisation of AI techniques (including neural networks) is enunciated.

Post-2015, there has been a rapid growth in applying ML techniques towards data-driven decision support in the wind industry, in particular towards utilising SCADA datasets for data-driven decision making. An interesting observation in this period is that there was a significant growth in studies focusing on performance assessment/analysis tasks for O&M, while CBM methods utilised for turbine health monitoring still continued to remain popular. However, a critical change noticeable is the dynamic shift from utilisation of vibration data to a greater focus on SCADA datasets. Figure 2.5 outlines the top-50 notable keywords prevalent in CBM publications at present, clearly enunciating the rise in dominance of neural networks. This phenomenon has directly led to a significant rise in the number of publications which utilise AI for CBM and performance assessment of wind turbines, especially from 2017 onwards, which is evident from the growth in annual publications as per Figure 2.6. It is interesting to note that in the year 2017, there was the prevalence of an AI winter in the wind industry, which caused significant reduction in interest for application of AI towards O&M tasks, which is possibly due to reduction in funding and/or lack of vital resources, such as quality data. During 2015 to 2020, there have been a variety of applications in CBM for which AI techniques have been utilised, with the thematic evolution for these concepts illustrated in Figure 2.8. Evidently, conventional ML algorithms spanning variational approximations, Bayesian inference, maximum likelihood estimation techniques etc. have been the dominating forces during the period of early evolution of AI in the wind energy domain. It is particularly notable that unsupervised linear transformation methods like Principal Component Analysis (PCA) have seen utilisation for tasks such as feature extraction and dimensionality reduction. There has also been prevailing interest in the area of novelty detection (e.g. identification of abnormal events) for health monitoring of turbines and their sub-components. During the later half of this period (2017 onwards), we can see the thematic rise in CBM using DL techniques, which especially includes feedforward neural networks aimed at regression (e.g. prediction of turbine power output time-series values and for short-term prediction of wind speed), diagnosing faults, optimising operations and system design of turbines etc. The word dynamics for most frequent words pertaining to CBM focusing on AI publications at present are outlined in Figure 2.7. Clearly, the diversity of applications for AI require careful consideration and analytical understanding of the present, which is discussed below with a specific focus on various different types of relevant algorithms.

Figure 2.9 depicts the treemap enunciating the hierarchical composition and corresponding focus areas for AI techniques used in CBM and performance assessment of turbines at present in the wind energy domain, wherein, the outlined keywords co-occurred together in the majority of publications during this time period (for instance, *neural-network* prevailed as a keyword alongside *performance*, *regression*, *decomposition*, *classification* etc). A variety

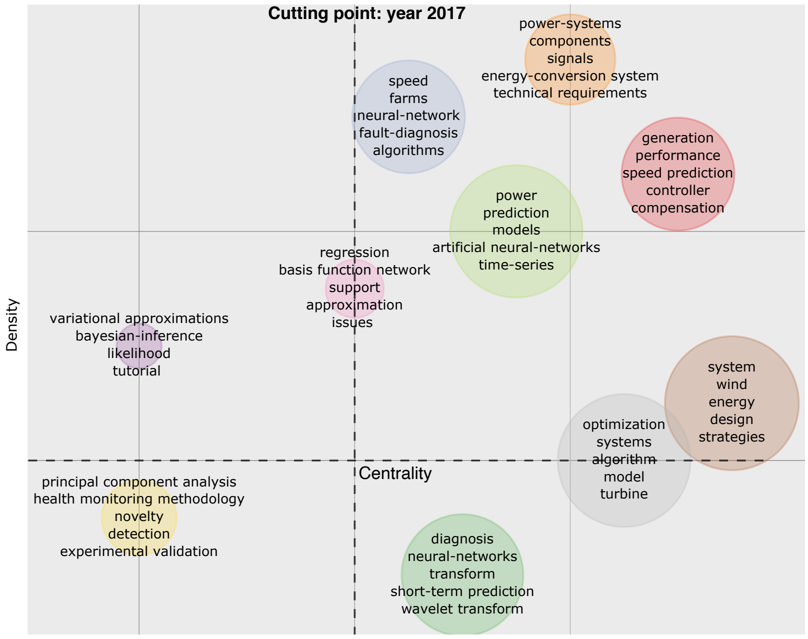


Fig. 2.8 Depiction of evolution of AI with corresponding themes over time. Here, the cutting point is the year 2017.

of different tasks, such as classification and regression being utilised for fault diagnosis, maintenance, forecasting turbine power output etc. are also clearly enunciated. It is interesting to note that still, there are mentions of terms which pertain to the signal processing domain, like wavelet transform and EMD – outlining that these techniques have not been foregone by the wind industry, but have continued to be used to complement many modern AI methods and algorithms towards data-driven decision support. The clusters and context for these AI techniques being utilised at present for O&M is shown in Figure 2.10, which clearly describes the dominance of classification methods aimed at diagnosis of faults and regression methods towards prediction of vital operational parameters during multi-step time-series forecasting, utilised together in conjunction with decomposition techniques conventionally prevalent in signal processing.

Regression techniques utilised for CBM and performance assessment/analysis Regression methods are the simplest form of ML techniques which have been utilised in the wind industry, wherein, the aim is to predict continuous values for vital operational parameters of the turbines over time. Given the time-varying nature of SCADA parameters, regression techniques are extremely suitable for application in a supervised learning envi-

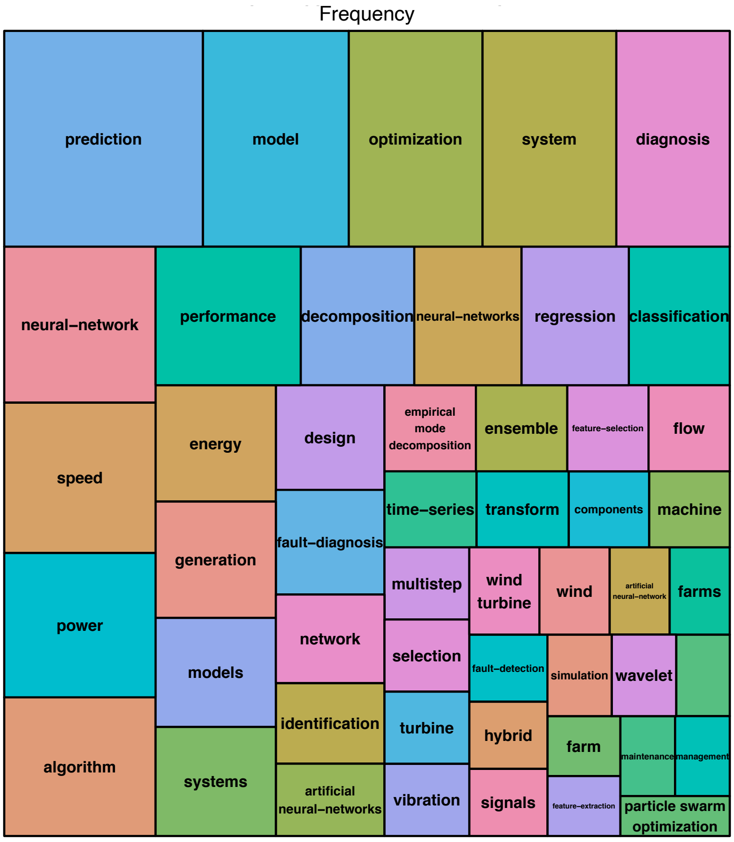


Fig. 2.9 Treemap describing hierarchical composition and structure of AI models utilised for data-driven decision making

ronment towards prediction of target values, like turbine power output, wind direction, wind speed etc. in new, unseen datasets.

Note that while there has been a dominant focus on signal processing techniques for CBM and performance assessment/analysis pre-2015, some studies have demonstrated promising early applications of utilising conventional ML methods which are worthy of mention. Clifton et al. (2013b) utilised simulated data obtained using aero-structural simulations for a 1.5 MW turbine towards applying regression trees for prediction of wind turbine power output, accounting for vital parameters like wind speed, turbulence intensity and expected shear. Their technique was observed to significantly outperform the conventional curve fitting technique (which utilises power curves) for prediction of turbine power output. In addition, their study demonstrated the ability of regression tree models to be further applied to data from new turbine test sites for prediction of the turbine's performance without the need for additional training data. A similar study by Clifton et al. (2014) focused on utilising decision trees for evaluation and prediction of turbine performance in mountain pass regions

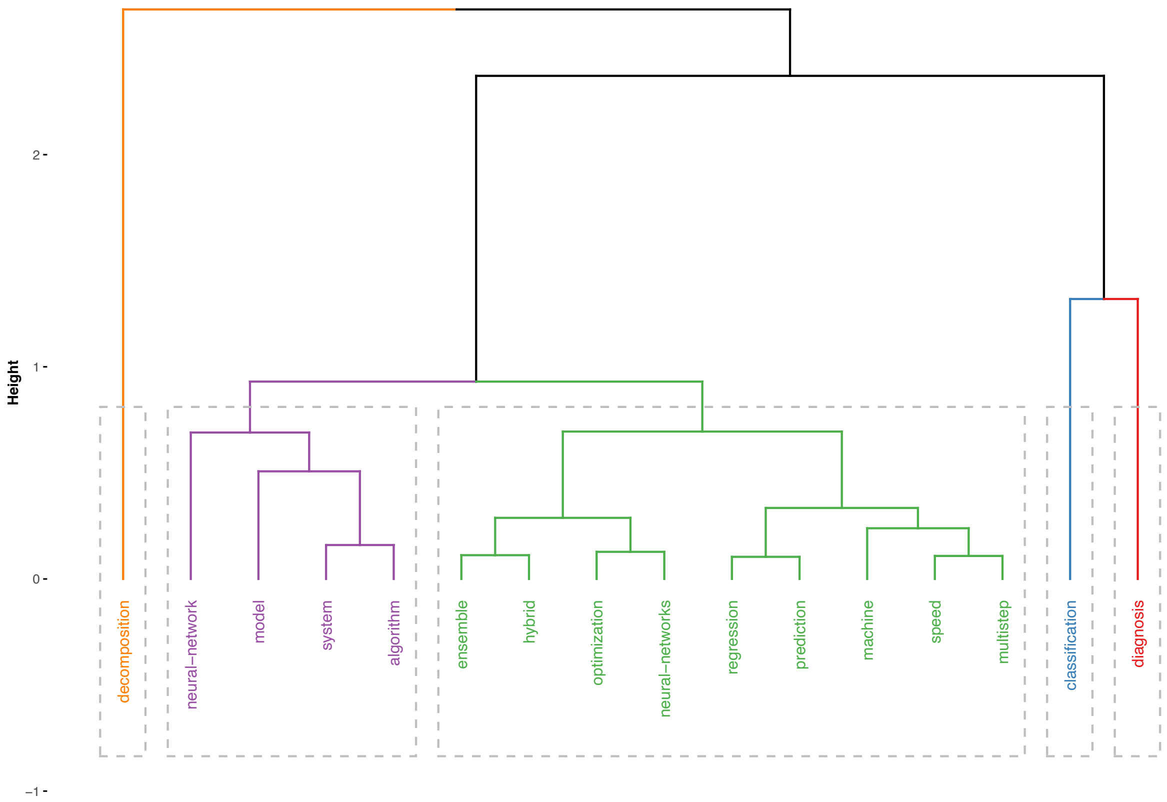


Fig. 2.10 Dendogram showing various clusters and their context towards application of AI methods for O&M at present

as a corresponding response to the prevalent pass wind time-series. There have been some studies which have also applied Support Vector Regression (SVR) (Zhao et al., 2010) for prediction of turbine power output. Yang et al. (2019) have previously utilised SVR for the development of a reconstruction-based ML model to detect faults in real-time across turbine sub-components, utilising the residual error between the SCADA feature signals to identify anomalies. Despite being simple to apply and interpret, such approaches do not utilise historical turbine data for decision making. Additionally, mathematical modelling of reconstruction signals is generally a computationally expensive process, which defeats the goal of achieving scalable real-time predictions that generally require minimal computational resources.

Besides performance assessment/analysis tasks, there have been some studies which have utilised regression techniques for CBM – Park et al. (2013), for instance, utilised Gaussian Mixture Models (GMM) alongside Gaussian Discriminative Analysis Models (GDA) for performing structural health monitoring in wind turbines, wherein, data characterising the wind field (like wind speed and direction, turbulence, profile etc.) were used. They found that such approaches can be highly promising for predicting the load response, and are capable of being extended to new turbine sites, given openly available data pertaining to wind resource assessment.

There has been a significantly rapid growth in applications of ML techniques for regression tasks post-2015. Particularly, DL techniques have witnessed an especially major focus during this period. In an early work, Du et al. (2016) proposed a specialised technique for anomaly detection by utilising the Pearson correlation coefficient for performing parameter selection in the process of modelling the turbine behaviour. They also used Self-Organising Maps for dimensionality reduction of the prevailing SCADA features. In the paper, the end predictions are utilised for mapping power outputs to corresponding ideal power curves for the turbines, which are thereby used to identify anomalies in operational behaviour based on points which fall outside the curves. While this technique is simple to apply and promising, it relies significantly on power curves for performing the final anomaly prediction in the process, which makes it less competent in comparison to other types of AI algorithms which are capable of directly generating predictions given vital SCADA parameters (Nghiem et al., 2017). There has been another study in this area by Morshedizadeh (2017), who focused on predicting turbine power production on the basis of historical data on turbine performance, demonstrating through their study that combining dynamic Multilayer Perceptron (MLP) model together with Adaptive Neuro Fuzzy Inference System (ANFIS) can play an integral role for optimal prediction of turbine power outputs. However, it should be noted that similar techniques have mostly been outperformed by other types of sophisticated AI algorithms in the last few years, especially those utilising DL.

With the move towards adopting more sophisticated AI models in CBM, DL techniques have seen wide applications in the prediction of vital operational parameters for turbines, which can be useful for the purpose of performance assessment. Qureshi et al. (2017) used data from an operational wind farm for developing an ensemble learning approach, combining deep autoencoder models (as the base-regressor) together with Deep Belief Networks (as the meta-regressor) for predicting turbine power utilising meteorological features. Their study showed that hybrid ensemble approaches like these can perform significantly better than conventional regression algorithms. In addition, the paper shows the feasibility of the proposed method to facilitate transfer learning, which can help provide power output predictions in circumstances with lack of additional training data. Recently, there has also been some interest in utilising Recurrent Neural Networks (RNNs), especially Long Short-Term Memory Networks (LSTMs) (Hochreiter, 1998; Hochreiter and Schmidhuber, 1997b) with SCADA features along with meteorological parameters. In comparison to conventional

Artificial Neural Networks (ANNs) (Zhang et al., 1998), RNNs are capable of accounting for past temporal information, which helps in making them a competent model for the processing of sequentially natured datasets, which are generally evident in the wind industry. There has been an early development in this area in a study by Kulkarni et al. (2019), wherein, the authors applied LSTMs to perform long-term forecasting of wind speed at a given wind farm site, and the predictions were ultimately utilised to perform fatigue analysis for blades from a 5MW wind turbine under consideration. Their approach outlines that RNNs hold immense promise to facilitate dynamic calculations of wind load. LSTMs have seen similar applications towards time-series forecasting of wind turbine power output. For instance, Zhu et al. (2017) and Liu et al. (2019) have showcased that LSTMs can achieve high forecasting accuracy for tasks such as short-term prediction of wind power, and generally significantly outperform conventional AI techniques such as ANNs and SVMs.

Classification techniques utilised for CBM and performance assessment Classification techniques play an integral role in ML towards segregation of two or more categorical variables, such as different types of faults in multiple turbine sub-components, operations segmented as per different power curve regions etc. Note that simpler techniques for classification use logistic regression (Peng et al., 2002), which can facilitate probabilistic predictions, thereby helping understand the most (or least) probable classes which fall into a particular group. However, these techniques often show poor performance for purposes of modelling and classification of non-linear data, particularly in cases wherein there are multiple possible hyperplanes, which is prominent with SCADA data being generally highly complex and non-linear in nature.

Classification techniques have seen wide utilisation in the last decade for analysis, diagnosis and prediction of turbine faults. An early application of such techniques was proposed by Leahy et al. (2016), who applied various types of classification algorithms to SCADA data from turbines with the aim of filtering and analysing faults and alarms in conjunction with the power curve. The paper showcased that SVM performed the best as a classifier model towards prediction of incipient faults in advance across multiple types of turbine sub-components. Despite demonstrating the promise of AI in tasks pertaining to CBM of sub-components in the turbine, the paper lacks in utilising more sophisticated models other than binary classifiers for the otherwise multi-class classification problem to identify specific faults. In addition, there is no application of feature selection and dimensionality reduction techniques on the SCADA data, thereby not accounting for the fact that all SCADA features utilised to train the SVM model may not be relevant.

Some studies have focused on utilising decision trees towards Fault Detection and Isolation (FDI). Si et al. (2017) utilised random forests together with Principal Component Analysis (PCA) for identifying faults prevalent in multiple turbine sub-components (like pitch system, blades, gearbox, yaw drive etc.) and determining the dominant SCADA signals contributing to the predictions. The paper shows that decision tree algorithms such as random forest are capable of measuring and parametrising the importance of the SCADA signals, which can play an immensely useful role in the analysis of predicted faults. Also, these algorithms can directly be fed with large datasets and have highly efficient performance in terms of training time compared to other popular approaches like SVM, which, despite their usual merits have limited capability of working with larger datasets without overfitting. Another notable study in this domain by Canizo et al. (2017) focused on the utilisation of big data frameworks like Apache Spark, Apache Mesos, Apache Kafka, HDFS etc. for the development of a real-time system for predictive maintenance. Their proposed system consists of a specialised online fault tolerant monitoring agent with a random forest learning model. The paper utilises SCADA data alongside historical failure records (which were previously stored in a cloud server) for predicting the turbine's operational status at an interval of 10-minutes based on SCADA data-stream processing. This methodology can particularly be helpful in facilitating real-time decision support for assisting engineers and technicians in the wind industry during O&M activities. Also, this is most likely the only study in the domain of CBM that proposes a complete solution for deploying AI models, right from development and training of the model to its end deployment on a cloud server with a front-end dashboard for the users. Despite the advantages of the methodology, the study mentions that the trained models for prediction-making were not updated for automatically adapting to actual operational statuses of the turbines, and incorporating online updates to the model is proposed as future work in the paper.

In a similar vein of research on leveraging classification techniques for CBM, Abdallah et al. (2018a) performed root cause analysis of turbine failures by utilising Classification and Regression Trees (CART). Particularly, the paper demonstrates the highly promising performance of ensemble bagged trees in identifying the sequence of events which contribute to faults across specific turbine sub-components. Moreover, the approach also provides the ability to identify the specific range of values for the SCADA features which lead to the faults (e.g. the gearbox oil temperature values going beyond a particular normal range). The decision trees can be visualised with ease by turbine engineers during O&M, who can thereby decide appropriate corrective actions. While this is promising, there are no details regarding performance metrics (like accuracy, prediction speed etc.) in this study.

Moreover, the paper does not explain how incipient failures can actually be averted through the utilisation of decision tree classifiers, given that SCADA features generally consist of series of continuous measurements over time, and for data with temporal nature, models such as RNN, Autoregressive Integrated Moving Average (ARIMA) etc. are generally better suited for providing reliable predictions. Another study in this area by Abdallah et al. (2018b) focused on a conceptual framework which describes a hardware-software solution, utilising decision trees to facilitate detection of faults in real-time. The study also proposes the interfacing of the predictive model with a distributed data storage cloud server to provide a system for real-time analytics. This framework can play a vital role in providing autonomous decision support, particularly by leveraging simple and easy to interpret AI models in the likes of decision trees. However, a major constraint would be the utilisation of more sophisticated AI techniques (especially those based on DL) for providing an interface with a real-time decision support system, given the added complexity and challenges such models pose for their deployment in the wind energy domain. This is a critical issue for the wind industry, and we believe that it is necessary to address this in the near future for effective O&M solutions.

More recently, the wind industry has witnessed rapidly growing interest in utilising DL models for O&M, especially in tasks pertaining to classifying faults in turbines. Figure 2.11 shows the evolving trends in data-driven decision making witnessed by the wind industry with time, clearly outlining the move from more conventional techniques utilising signal processing in the past towards neural networks for SCADA data at present.

The essence of DL lies in utilising neural networks consisting of multiple layers, which are capable of learning from datasets with complex, non-linear relationships (Suykens et al., 1996). Neural networks can automatically identify associations and patterns between the provided inputs and outputs, which makes them highly competent in learning and modelling complex intermediate representations prevalent in the data (Goodfellow et al., 2017), which is generally the prevailing nature of SCADA features. In recent efforts focusing on leveraging neural networks for the purpose of CBM pertaining to specific turbine sub-components, Lu et al. (2018) applied ANNs to SCADA data for the prediction of life percentage of the turbine sub-components. Their proposed approach can be intrinsic in identifying faults based on the conditional probability of failures, obtained using the ANN model's predictions and the historical account for the failure time distribution of the various components. Such approaches can help the O&M engineers and technicians to perform better planning and inventory management to ensure surplus storage for those sub-components which are most prone to failures and operational inconsistencies. However, there is no description of the training and test performance for the ANNs in the paper, as well as the basis on which the

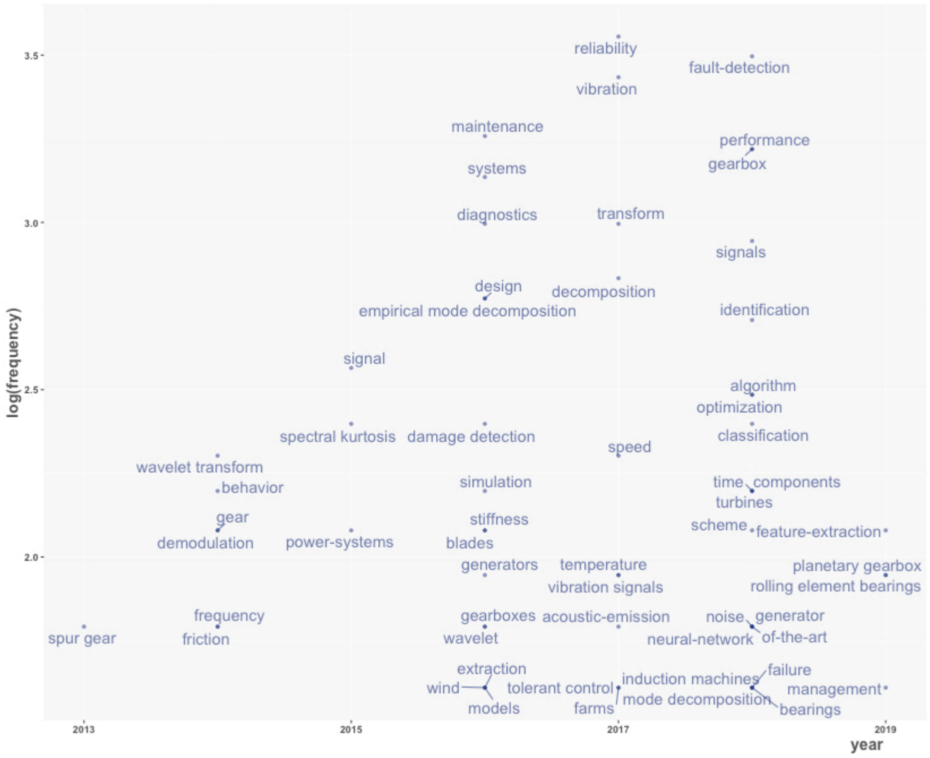


Fig. 2.11 Trends for data-driven decision making in the wind energy domain. The growth in interest towards utilising more sophisticated models like neural networks is clearly evident, while conventional methods (e.g. using signal processing) still continue to be in use.

specified network architecture was chosen. Moreover, the focus of the paper only specifically spans the prediction of lifetime percentage across four sub-components in the turbine, i.e. the gearbox, generator, rotor and pitch system, which, as per the authors suffer from being most prone to failures. However, turbines consist of many other integral sub-components, such as yaw system, hydraulic system etc., which are not addressed in the paper.

Qian et al. (2015) have previously utilised the Extreme Learning Machine (ELM) model, which can help in identification of faults on the basis of deviations of the prevailing SCADA features from ideal SCADA signals. The paper demonstrates that the ELM model outperforms conventional feedforward neural networks while taking a considerably less amount of time for training and making predictions, given that the ELM model can randomly perform updates on the weights and biases unlike ANNs which rely on gradient-based learning algorithms to perform optimisation. Moreover, the model is capable of predicting incipient faults in advance across specific sub-components of the turbine, which directly contributes to reduction of maintenance costs. However, the paper does not present the details of the ELM model utilised and lacks a performance comparison between the ANNs and ELM based on accuracy of fault prediction and training time. Also, as it is not always the case that deviations from ideal signals would be indicative of faults, the approach can often raise false alarms, especially with highly sensitive sensors and thereby lead to forced outages and increase in O&M costs.

Some studies have applied AI for diagnosing faults and identifying incipient failures in external sub-components of the turbines through computer vision techniques (Li et al., 2015, 2014; Moreno et al., 2018; Yu et al., 2017). In this area, Convolutional Neural Networks (CNNs) have seen utilisation by Li et al. (2014) for identification of faulty instances in images of turbine blades, with the model achieving a prediction accuracy close to 100% in some instances. Besides this, there has been successful demonstration of hybrid models like combinations of SVM together with CNNs which was proposed by Yu et al. (2017), wherein, the model showed promising results especially in learning from small datasets consisting of labelled images of turbine blades. Although such methods are generally well suited in CBM specific to aspects concerning external sub-components such as blades, they lack the ability to be utilised for the purpose of anomaly prediction across multiple other critical sub-components like pitch system, gearbox etc. In addition, as recording images requires the utilisation of drones or other similar image capturing devices, the methodology is not cost-effective and can be highly prone to failures and cases of false alarms, especially during situations of rain, snow, mist etc.

Installation of wind turbines at new sites is a complex process, generally requiring critical analysis of conditions including weather and location for ensuring that maximum power is produced at the lowest cost. The area of optimising turbine performance through appropriate planning for layouts has seen some work, e.g. by Dutta and Overbye (2011); Menghua et al. (2007); Wu et al. (2014) etc. A notable study in this domain by Wu et al. (2014) utilised AI techniques like genetic algorithms to perform layout planning of wind turbines along with optimisation techniques such as the ant colony algorithm to decide on the optimal line connections in the determined topology. This study takes into account the wake effect and utilises metrics including the wind speed time-series along with the cable parameters for interconnections between the turbines in the wind farm during optimisation of the layout. The paper shows that AI techniques hold immense promise in tasks pertaining to optimisation of turbine topologies as these algorithms can account for an exponential number of different cases and distribution types for determining optimal solutions, which is generally highly complex and practically infeasible when utilising manual techniques for optimisation.

As is clearly evident, while most of these existing studies have focused on performance assessment utilising metrics pertaining to power prediction (like power curve) and deviations identified from ideal SCADA signals, the domain of research with a specific focus on CBM pertaining to anomaly prediction and identifying incipient failures in turbines is presently still at an embryonic stage. In an early work in this domain (which is possibly the only paper in the pre-2015 period to particularly focus on DL techniques for CBM), Zaher et al. (2009) performed temperature-based anomaly detection in specific sub-components of the turbine such as generator and gearbox by utilising SCADA data. The paper utilises multilayer neural networks to identify abnormal situations for operational temperature prevailing in the turbine's sub-components, and the authors also extended the methodology to be applicable on an entire wind farm through a specialised Multi-Agent System (MAS) architecture. Note that this study did not use historical logs of failures, as these were not available to the authors and are additionally generally difficult to obtain for purposes of research, given their commercially sensitive nature to wind farm operators. Despite showing immense promise as an early study in DL for CBM and performing anomaly prediction across multiple sub-components in the turbine, the purpose of achieving complete automation in O&M is defeated, as professional engineers and technicians still need to identify and segregate the specific classes of faults (e.g. on the basis of their severity and according to specific alarm events).

In another vein of research following a different methodology, Bach-Andersen et al. (2015) used a CNN for the purpose of predicting faults by utilising vibrational signals obtained from turbines. The paper showcases highly promising results, and demonstrates that the CNN significantly outperforms other conventional ML techniques which the authors used as baselines. Similar promising results have been achieved by Ibrahim et al. (2016), who applied DL techniques to turbine data, specifically utilising ANNs together with Current Signature Analysis for predicting anomalies. In a more recent study, Pang et al. (2020b) utilised SCADA data and applied a hybrid model (the spatio-temporal fusion neural network) for multi-class fault prediction. Particularly, the paper proposes a novel application of multi-kernel fusion CNNs for learning the multiscale spatial features as well as the correlation prevailing between these variables in conjunction with an LSTM network for further learning the temporal dependencies in the data. The hybrid model was found to outperform several other competitive ML techniques, which outlines the promise of deep learners, especially the models combining multiple learning techniques towards predicting faults in turbines. In another aspect shown by a closely related study, Kong et al. (2020) developed a specialised hybrid model combining CNNs together with Gated Recurrent Units (GRUs) for performing fusion of the spatio-temporal SCADA features. Similar to LSTM networks, GRUs are capable of learning temporal dependencies present in complex and non-linear SCADA datasets, while utilising fewer parameters during training, which (in some instances) make it more effective in terms of accuracy and computationally efficient as shown in the paper. The CNN-GRU model proposed in the paper was trained by utilising historical data on normal behaviour for the turbines, and any prevailing deviations from normal operation circumstances based on residuals was utilised for identifying anomalies. The paper showcased that the method is highly effective for anomaly prediction, especially serving as a monitoring indicator for CBM. While these studies have demonstrated the capability of DL models in predicting failures with high accuracy, they fail to provide rationales and transparency behind their decisions, particularly regarding the SCADA features which exactly lead to the predicted fault (Chatterjee and Dethlefs, 2020e), which may make these approaches less suitable for practical adoption by turbine operators.

Owing to the challenging and time-consuming task of obtaining SCADA data consisting of historically labelled fault records, there have been some studies which have utilised unsupervised learning techniques for anomaly prediction. This includes the application of Denoising Autoencoders (DAE) by Jiang et al. (2018), who utilised SCADA time-series information obtained through multiple sensors and showcased that the DAEs are capable of learning non-linear representations from SCADA features under situations of noisy and fluctuating inputs. The DAE model was trained with normal data in this study, and the authors utilised a multivariate reconstruction model to detect faults by analysing the reconstruction error. The study also utilised a sliding-window methodology, which can be useful in capturing the prevailing non linear correlation in between the different SCADA features alongside the temporal dependencies, which makes it highly promising for effective detection of faults. Also, given that neural networks have mostly witnessed applications in supervised learning scenarios in the past, this study shows the promising avenues held by unsupervised learning, particularly with real-world turbine operational data. Note that despite the promise, most existing studies face a common challenge in lacking transparency, wherein, the black-box natured models are capable of making predictions with high accuracy but fail to provide rationales behind their predictions. Also, unsupervised learning techniques are generally utilised without any historical ground truth for failures and thus cannot be validated, making them less robust in comparison to their supervised learning counterparts.

For tackling the challenges posed by the issue of transparency, some studies have focused on employing Explainable AI models. Chatterjee and Dethlefs (2020e) utilised a novel hybrid model combining LSTMs with a gradient-boosted decision tree classifier (XGBoost) towards achieving explainable anomaly prediction using SCADA data. The study also showcased the feasibility of transfer learning in this aspect, by facilitating the prediction

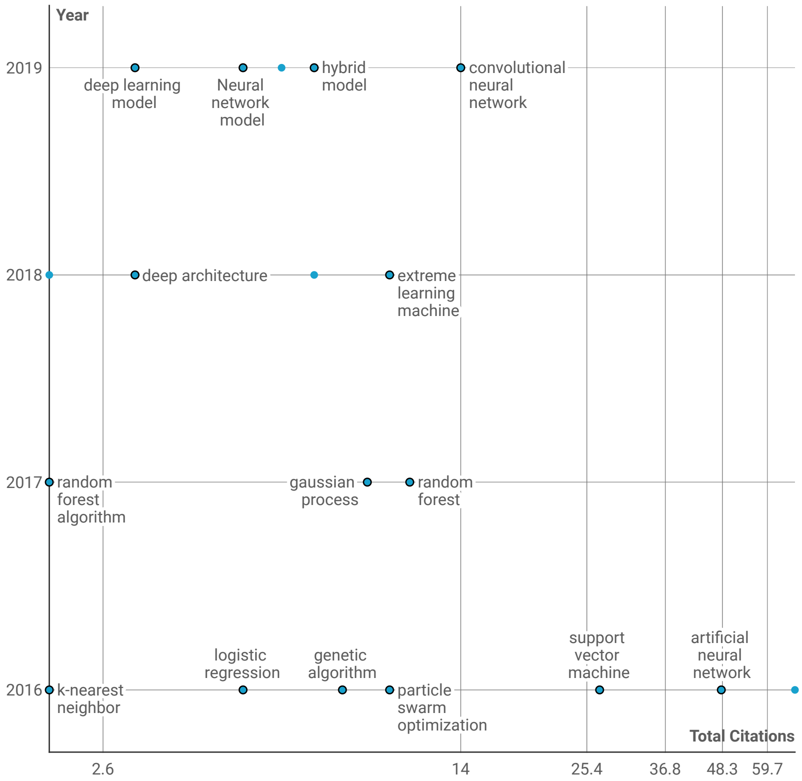


Fig. 2.12 Citation burst for terms relevant to AI prevailing in the wind industry from 2016-2019. The top 15 terms (on a logarithmic scale) across cited publications which utilise such models are outlined.

of faults in new domains (such as for wind farms which have not seen long periods of operation) without requiring access to data with historically labelled instances of failure. Wang et al. (2019) utilised a specialised LSTM architecture with attention for achieving transparent and interpretable prediction of wind power. The attention mechanism (Luong et al., 2015b) in neural architectures helps the learning models to focus dynamically on the relevant and vital SCADA features in the sequential dataset (time-series) which contribute to the prediction along with the list of features which influence the model's decisions, which generally leads to higher accuracy and transparency in the predictions. Similar to this effort, CNNs with attention mechanism have been applied to achieve predictions with higher accuracy and explainability, e.g. by Shivam et al. (2020) for the purpose of short-term wind speed prediction, by Chen et al. (2019a) for detecting imbalance faults in the blades and by Chatterjee and Dethlefs (2020c) for identifying causal relationships present in SCADA features during anomaly prediction. All these studies present novel insights on the immense promise which AI models tailored for the wind energy domain hold, especially RNNs and CNNs. However, it is clearly evident that the applications of Explainable AI models which the wind industry has witnessed are very limited in comparison to other domains like Natural Language Processing (NLP) and computer vision (Chatterjee and Dethlefs, 2020b), and this is discussed in more detail in Section 2.4.

Figure 2.12 outlines the top 15 keyword terms prevailing in the CBM publications utilising AI, which have received the strongest count of citation bursts (demonstrating their significant research impact and interest generated in the wind industry). It is interesting to note that this period was prevalent from the years 2016-2019, which makes it clear that during this time, AI for CBM of wind turbines witnessed a massive interest. We also observe that the publications which utilised ANNs in 2016 were the ones to garner maximum number of citations, with SVM being the second most popular ML technique. An importance inference which can be drawn is the dynamic shift which the wind industry witnessed from conventional ML models (like KNN, genetic algorithms, random forests, logistic regression and particle swarm optimisation) in the early periods of applications utilising AI in the wind energy domain (2016-17) to more sophisticated AI models, specifically those utilising DL techniques such as Deep Neural Networks (DNNs), ELM etc. From 2019 onwards, CNNs and other hybrid models which combine multiple neural architectures have seen significant research interest in the wind energy sector. Compared to this growth, there has been relatively little interest in the adoption of many other types of AI models (which includes LSTMs, fuzzy neural networks, autoencoders etc.). Figure 2.13 shows the composition evident for these less popular techniques. Notably, SVR and neural network models have dominated the consortium of these less popular techniques despite receiving limited attention, which outlines that they are likely to hold immense promise for CBM. We believe that that the adoption of such models more widely by the wind industry is integral to achieve optimal benefits in CBM during data-driven decision support.

Natural language generation techniques in the wind industry for human-intelligible decision support Despite applying AI models successfully for data-driven decision making with SCADA data, most existing studies have significantly neglected other sources of vital information available, especially historical records of alarms and failures. These records (generally termed as *event descriptions*) consist of comprehensive information detailing the historical faults which occurred in turbines in the form of human-intelligible natural language phrases describing alarms prevailing in various turbine sub-components (such as gearbox, yaw, pitch system etc.) alongside the time-stamps for these events in relation to the recorded SCADA features. Generation of *informative* messages from SCADA data is a data-to-text generation problem. In this area some studies have utilised Natural Language Generation

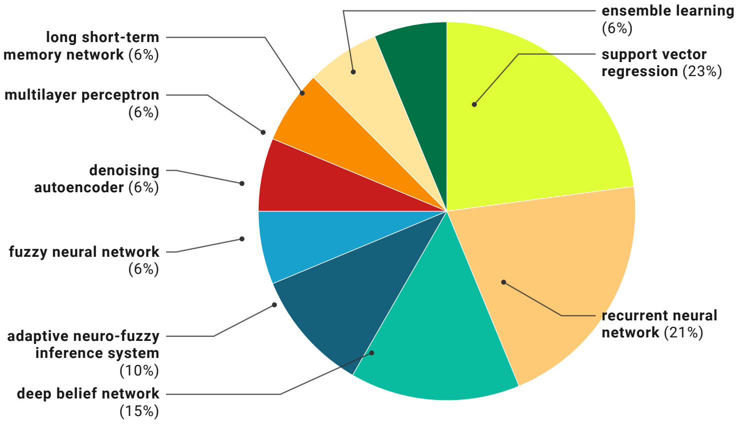


Fig. 2.13 Pie chart depicting the composition of prevailing less popular techniques used for CBM based on the citation burst for keywords from 2016-2019. The least frequently occurring keywords in cited papers across a total of 291 CBM publications which utilise such models are used.

(NLG) techniques, building upon the immense promise demonstrated by such methodologies in various domains such as automated planning, weather forecasting and spatial navigation etc. (Gardent et al., 2017; Garoufi and Koller, 2010; Gong et al., 2019; Juraska et al., 2018). NLG can play a critical role in supporting O&M decisions by shortening the time frames for analysis alongside providing engineers and technicians with human-intelligible decisions, which can help them to better understand the prevailing context of incipient failures. In addition, the overall purpose of data-driven decision making and automated planning is practically more or less defeated if AI models cannot provide suggestions for maintenance actions besides accurate predictions of faults. NLG techniques can act as a boon in the path to achieving transparent decisions, especially considering the sequential nature of datasets prevailing in the wind energy domain (including alarm messages, SCADA features and maintenance manuals/reports etc.). Specialised techniques for NLG such as few-shot learning (Chen et al., 2020c) also have the ability to generate detailed and informative messages even under circumstances with limited data for training, which makes NLG a highly promising area for the wind industry to widely adopt.

In one of the earliest works evident in this domain, Sowdaboina et al. (2014) applied NLG techniques for summarising time-series information which is relevant to the wind energy domain, primarily parameters such as wind speed and direction etc. Dubey et al. (2018) utilised Case-Based Reasoning (CBR) techniques for developing an end-to-end system capable of generating textual summaries for such types of meteorological information, and demonstrated the highly promising results obtained when CBR techniques were combined with conventional rule-based NLG techniques. While these studies have shown success with the presentation of such information, a key drawback which they entail is that they are only capable of presenting very limited information (i.e. generally 1-2 parameters), while there are generally multiple (often hundreds) of different SCADA features along with various types of failure records which could be potentially utilised for achieving transparency during decision support. The development of NLG systems tailored for such tasks is a challenging and time-consuming area, and additionally also creates many other constraints in situations wherein there is a lack of sufficiently labelled information (such as the ground truth labels of SCADA features which actually contribute to faults).

For tackling challenges like these, AI models have witnessed limited, but extremely promising results for CBM in the wind industry. In possibly the only study focusing on this area, Chatterjee and Dethlefs (2020d) demonstrated the applicability of utilising AI-based NLG models like Transformers for explainable decision support. The Transformer model (Devlin et al., 2019; Vaswani et al., 2017b; Vig, 2019) is a specialised type of neural architecture, which consists of the multi-head attention mechanism, that makes it capable of focusing on the relevant features in sequential datasets and elimination of recurrence which vanilla RNNs conventionally use completely. This also helps the model to perform better in learning associated relationships prevailing between features, while additionally reducing the computational complexity of the tasks significantly, making Transformer models highly promising for training using modern ML hardware. Owing to the sequential nature of SCADA datasets and the relevant desired outputs for the wind industry (such as alarm messages and maintenance actions), such techniques have been shown to achieve success in providing engineers with detailed diagnoses for failures in a human-intelligible format as well as suggesting appropriate maintenance strategies to avert any catastrophic failures during O&M. In addition, such models have the advantage of being explainable and transparent, and can provide a summary consisting of the exact lists of SCADA features leading to the model's predictions for alarms and maintenance strategies by utilising the mechanisms pertinent in multi-head attention (Vaswani et al., 2017b). More details on such types of NLG models is provided in Section 2.4. Clearly, while NLG techniques have witnessed promise in their early developments and applications pertaining to CBM, they have not seen wide adaptation and application for data-driven decision making. We believe that utilising NLG models is integral for the wind industry, in particular leveraging deep learners for generating human-intelligible O&M reports.

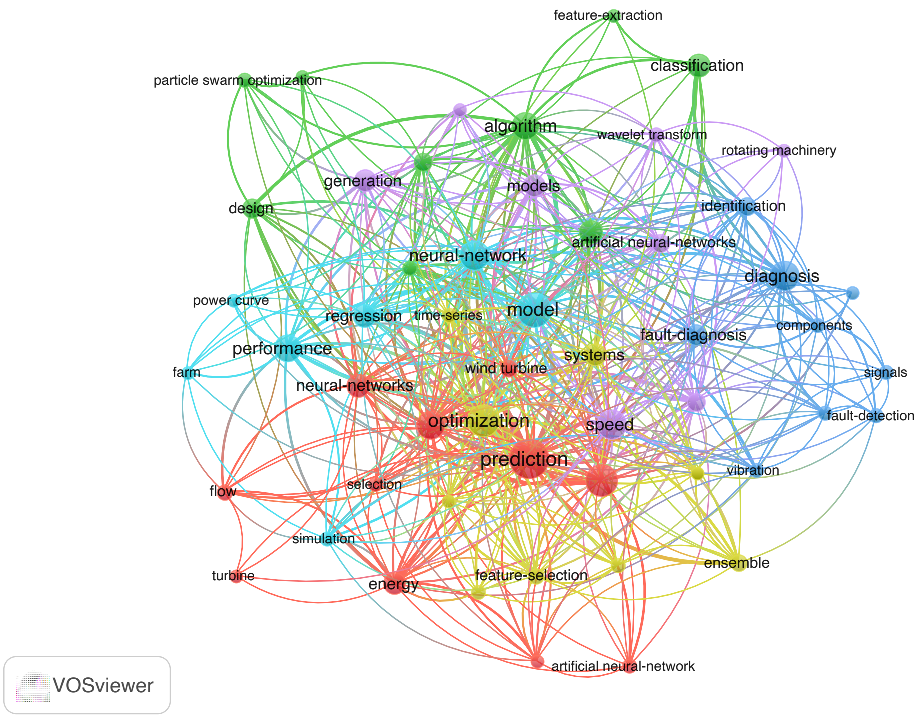


Fig. 2.14 Network visualisation performed in VOSviewer for AI utilised for CBM in the wind industry, based on publications from 2010-2020. The graph edges show the conceptual association prevailing between terms from different types of clusters, where similar colours represent strong association.

Reinforcement learning techniques in the wind industry for planning and optimisation

Given the generally complex and highly uncertain environments for deployment of wind turbines, it is often critical to perform optimisation and control of the turbines as a system. For achieving this, Reinforcement Learning (RL) (Kasim, 2016; Mousavi et al., 2018; Robbins and Monro, 1951) techniques have received some attention – which is a specialised branch of AI that has witnessed few applications in the wind energy domain pertaining to autonomous decision support and planning. In some early instances of research in this area, Fernandez-Gauna et al. (2016); Tomin et al. (2019) utilised RL algorithms for performing intelligent control of a wind turbine system's Multi-Input-Multi-Output (MIMO)-based controller. These studies have demonstrated promising results in comparison to most other traditional control methods, which are often subject to significant challenges when tackling multi-objective problems that are evident in O&M of present-day wind turbines. In another vein of research, Saenz-Aguirre et al. (2020) utilised Deep Reinforcement Learning (DRL) techniques for performing yaw control in turbines, and their study showed that such techniques when incorporated with the additional learning capabilities of ANNs can significantly outperform the traditional algorithms for RL. Similar promise of DRL techniques has been demonstrated by Chatterjee and Dethlefs (2020b), who performed maintenance planning for offshore vessel transfers by utilising operational SCADA data alongside multiple other parameters like predicted fault types and their severity, weather conditions etc. This demonstrates that RL holds immense promise and is feasible for adoption in the wind industry, and we believe that it is integral to pursue more research in this domain for performing optimal planning and optimisation during O&M.

Figure 2.14 outlines the network visualisation based on all AI publications which were prevalent in the last decade, which clearly showcases the stagnant rise in utilisation of AI techniques pertaining to CBM in the wind energy domain. On the basis of the graph edges along with the multiple different clusters (which are represented with different colours), it is clearly evident that there is a continued prevalence of predictive AI models for classification, regression and optimisation tasks utilising neural networks and conventional techniques for signal processing (such as wavelet transform), alongside power curves being utilised for similar purposes. It is also interesting to note that in these models, feature selection and feature extraction methods continue to play a significant role, which enunciates the prevailing focus on feature engineering when utilising SCADA data, as is generally the case with conventional ML techniques. Clearly, the focus on NLG and RL methodologies is very limited.

2.4 Future perspectives for the wind industry

Based on the scientometric analysis of the past and present in the wind industry, there is clearly an interesting and challenging problem faced in autonomous prediction and scheduling of O&M utilising data-driven decision making techniques. Although existing studies have made promising advances in various specific tasks like wind power forecasting (Messner et al., 2020) and predicting anomalies (Helbing and Ritter, 2018b), there has been very limited research into incorporation of explainability and transparency in the utilised AI models for data-driven decision support. The evident lack of research, in particular for the area of fault prediction during CBM, can most likely be attributed to the challenges in obtaining SCADA datasets and labelled historical alarms/failure records from wind turbines, as these are generally of commercially sensitive nature to wind farm operators.

Below, we focus on discussing some major challenges presently faced by the wind industry (and which are most likely to continue in the time to come) towards application of

AI techniques for data-driven decision making, and also provide a perspective on particular strategies which could possibly be used to tackle these challenges.

Table 2.2 Summary of different types of openly available datasets in the wind industry at present, which can be utilised in the domain of CBM

Dataset	**Type**	**Year Released**
2011 PHM Society Conference- Anemometer Fault Detection Data Challenge (PHM Society, 2011)	Paired anemoemeter data based at same height and shear data for anemometers at different heights, comprising parameters like wind speed, wind direction and temperature aimed at identifying excessive error owing to damage or wear conditions	2011
Wind Turbine High-Speed Bearing Prognosis Dataset (Bechhoefer et al., 2013; E. Bechhoefer, 2013, 2018)	Bearing health prognosis dataset consisting of vibration and tachometer signals from a real-world turbine high-speed shaft bearing (Bechhoefer et al., 2013), which also faced actual inner race fault conditions	2013
ENGIE La Haute Borne (Engie Renewables)	SCADA data from an operational onshore wind farm	2013
NREL Wind Turbine Gearbox CBM Vibration Analysis Benchmarking Dataset (National Renewable Energy Laboratory- National Wind Technology Center, 2014)	Vibration data obtained through accelerometers and high-speed shaft RPM signals collected during dynamometer testing, alongside information on real damage conditions in turbine gearbox for performing benchmarking of vibration based CBM techniques	2014
Platform for Operational Data: Levenmouth Demonstration Turbine (ORE Catapult, b)	Data from an operational offshore wind turbine, including SCADA, historical logs of alarms, substation data, and Met mast data	2017
Ørsted Offshore Operational Data (Ørsted)	SCADA data from 2 operational wind farms, with on-site 10-minutes statistics from wave-buoy and ground based LiDAR	2018
EDPR Wind Farm Data (EDP Renewables, 2018a,b)	Historical dataset from an operational offshore wind farm comprising of SCADA signals, Met mast data, turbine failure logs and relative positions of turbines and Met mast	2018

Table 2.3 Summary of different types of openly available datasets in the wind industry at present, which can be utilised for performance assessment/analysis

Dataset	**Type**	**Year Released**
NREL Western Wind Dataset (National Renewable Energy Laboratory, b)	Data with historical weather information (wind speed, air temperature, pressure etc.) and power output from multiple operational wind turbines	2004
NREL Eastern Wind Dataset (National Renewable Energy Laboratory, a)	Simulated data of wind speed and turbine power output, with short-term forecasts	2004
Platform for Operational Data: Floating Turbine Design Cases (ORE Catapult, a)	Measurements from an operational floating turbine, with operational cases for multiple wind speeds and wave heights	2019

2.4.1 Challenges in availability of data and ensuring quality

AI techniques are generally dependent on the availability of large amounts of data to facilitate optimal decision making in tasks pertaining to real-world application domains (Foresti et al., 2020). However, as the prevailing nature of data in the wind industry is generally highly commercially sensitive (Chatterjee and Dethlefs, 2020e), most wind farm operators are not willing to share this information in the public domain, which is an integral resource for researchers. In addition, annotation of events and alarms which vary rapidly in case of complex engineering systems such as wind turbines is an immensely challenging task for engineers as well as wind farm operators, and this may not always be on top of the agenda. Also, it is possible for new turbines to not have been in operation for long periods of time (Chatterjee and Dethlefs, 2020e), which makes it challenging for acquisition of even small datasets. For analysing the prevailing situation at present in the wind industry, we provide a comprehensive summary enlisting some datasets presently openly available which can be utilised for CBM in Table 2.2 – these are the only prevailing information sources available in the public domain at present to the best of our knowledge. It is interesting to note that of all these data sources, only two consist of historical alarm logs for failures. Also, it is worth mentioning that besides these datasets for CBM, some other datasets exist which primarily fall beyond the scope of CBM, and these are enlisted in Table 2.3. While these cannot be used to make informative decisions by training AI models for the purpose of fault diagnostics or prognostics in O&M activities, they can be vital to assess the performance of turbine operations (such as the efficiency and power production). We refer the interested reader to Menezes et al. (2020) for comprehensive details on several other types of datasets and their applications for O&M in the wind industry.

While datasets consisting of historical weather information alongside turbine power outputs can play an integral role in forecasting future trends in the wind industry and deriving novel insights pertaining to operational feasibility of turbines (such as by utilising power curves) (Goudarzi et al., 2014), they suffer from the inability to be utilised for fault prediction across specific turbine sub-components and cannot provide effective descriptions of the possible causes leading to the faults. The datasets which contain vibration data, especially with vibration signals recorded during historical situations of faults can play a potentially more useful role, as they can help characterise the operational status of various turbine sub-components (e.g. the gearbox, pitch system etc.), and can thereby support vibration analysis based CBM for the wind industry with an immensely powerful role in O&M.

The SCADA datasets which are openly available at present help in identifying operational parameters in turbines and their different sub-components (like gearbox oil temperature, active and reactive power, pitch angle etc.). In addition, these datasets also usually consist of meteorological information (such as wind speed, temperature, air pressure etc.) which is measured at the Met Mast (Mittelmeier et al., 2016), and these can be highly beneficial during wind resource assessment. In situations with lack of historical alarm logs, unsupervised outlier detection based AI techniques (Goldstein and Uchida, 2016) can be useful for discovery of hidden patterns across the SCADA features and identification of potential faults based on the discriminatory features which affect data points present in certain clusters. However, such types of techniques are not capable of being validated as they lack the ground truth pertaining to normal operation/anomaly, and more importantly, fail to provide more fine-grained and comprehensive descriptions of prevailing faults alongside their causes, such as through alarm messages.

SCADA datasets which consist of labelled historical alarm records generally prove to be a more useful resource for application of AI techniques, as supervised learning techniques (Helbing and Ritter, 2018a) can be utilised for the development of predictive AI models for fault prediction through training of the model on a subset of the historical datasets, and thereby facilitate prediction making on new, unseen instances of test data. This often serves to be more reliable, as it facilitates the retrieval of performance metrics (like accuracy), owing to the fact that the original labels for normal/abnormal operation are available. Also, as alarm logs can provide detailed descriptions for failures in terms of messages which can describe the exact sub-components in the turbines which have the faults alongside their characteristics (Chatterjee and Dethlefs, 2020d), they are capable of providing significantly more detailed insights during O&M. However, as outlined before, these types of datasets are very difficult to obtain. This also creates additional challenges in the production of meaningful novel results and comparing them with baselines, given that researchers generally apply the AI models and train them with very specific corpuses of SCADA data (with possibility of wide variations across datasets based on the features and the turbine specifications), and such data used in these publications are mostly not shared publicly along with the published research. This pertaining trend thereby severely limits the comparability as well as replication of the published research in the wind industry.

Besides the major challenge posed by lack of access to *useful* data, another significant difficulty is the quality of datasets in the wind industry. As major developments continue at a rapid pace in the wind energy domain, there are several types of big data, especially with growing resolution and complexity which are becoming available from different sources such as LiDARs and buoys, operational data obtained from hundreds of sensors in the turbine and its sub-components, wind and wave metrics etc., which make it integral to perform appropriate filtering and quality control to clearly interpret vital information pertaining to turbine operational status (van Kuik et al., 2016). It is important to note that the issues with data quality severely affect fault diagnostics and prognostics in wind turbine CBM, but besides this, they also have a strong influence on additional O&M tasks which despite being beyond the scope of CBM could play an integral role for turbine operators in performing performance assessment/analysis.

There has been very limited focus prevalent in the wind industry on the identification of key challenges and issues which persist in the utilisation of turbine data, being particularly vital for the training of AI models that depend on accurate and scalable data (Roh et al., 2019) in providing informative decisions. In likely the only study to specifically focus on the issues of data quality in the wind industry, Leahy et al. (2019) outlined the key issues which pertain to the lack of unified standards across multiple types of datasets (like SCADA, alarm codes, maintenance actions and work orders etc.), limited availability of alarm and failure information with useful context and the major challenges in manually processing such information into usable formats to train data-driven CBM models. More recently, as the research focus on utilisation of AI for O&M in the wind industry has continued to develop, particularly in the area of leveraging DL techniques, there are new issues which are emerging pertaining to data quality challenges which affect the development and deployment of AI models, especially highly sophisticated learners as enunciated below:-

1. **Prevalence of imbalanced datasets:** SCADA datasets which consist of historical records of alarms generally face a significant imbalance amongst the data samples pertaining to normal operation and anomalous behaviour (Chatterjee and Dethlefs, 2020e), wherein, there are a significantly higher number of samples from the data being classed as normal operation, given the limited availability of failure records or in situations of some fault types (such as in gearbox, generator, blades etc.) having a much higher frequency in comparison to others (Zhu and Li, 2018). By training AI models with imbalanced datasets, the models can generally become biased in predictions towards the majority class samples (labelled as normal operation), and thereby potentially cause missed detections, wherein, the model would classify anomalous instances as normal. Such situations are most likely to be overlooked by turbine operators during O&M decision making and planning (Leahy et al., 2017), and these can thereby lead to sudden and unexpected failures posing significant costs.

2. **Insufficient quality of available contextual information pertaining to faults:** The alarms logs and SCADA data are capable of indicating the occurrence of faults in different sub-components of turbines, alongside the relationships of such failures in components amongst other sub-component types and adverse environmental conditions (Gonzalez et al., 2016). More recently, these historical alarm records are starting to become available as brief natural language phrases (e.g. *"Wind direction transducer error 1 & 3"*), that can help provide detailed contextual information on the failures and can thereby be leveraged in data-to-text generation systems to produce effective event descriptions from SCADA features for fixing/averting potential failures in line with expert judgements (Chatterjee and Dethlefs, 2020d). The basic principle of data-to-text generation systems focuses on utilising NLG techniques for generating human-intelligible unstructured descriptions for failures in the form of natural language phrases based on structured SCADA data. NLG models, especially those based on neural machine translation typically have a heavy dependence on appropriate quality of data samples utilised to train the models, and such models are quick to memorise the low-quality samples (Rikters, 2018). The alarm messages available in the wind industry are often of insufficient quality standards to merit appropriate training of NLG models in achieving human-level intelligence, and suffer from a major challenge in lack of diversity across the available corpus as some particular types of alarm messages (such as *"Pitch System Fatal Error"* owing to the frequent occurrence of disorientation in pitch angle of turbines (Chatterjee and Dethlefs, 2020d; Myrent et al.) are generally

prevalent more frequently during routine O&M tasks, and are thus given much more attention in the wind industry.

Engineers and technicians may not always prioritise the manual annotation (or development of suitable automation techniques) for recording alarm messages which summarise the contextual information pertaining to low-priority failures (such as *"HPU 2 Pump Active For Too Long"*) (Chatterjee and Dethlefs, 2019a), leading to significant variance in the quality of such messages across different types of turbine sub-components. It is essential to have an appropriate diversity in the data samples, to ensure generation of coherent text and deriving useful insights (Ji et al., 2020) for domain-specific tasks, which makes the utilisation of NLG for decision support a challenging ambition for the wind industry. Also, unlike plain text (such as those utilised in translation from one language to another), alarm messages prevalent in the wind industry often contain important symbols & numbers (such as in the message *"(DEMOTED) Gearbox oil tank 2 level shutdown"* (Chatterjee and Dethlefs, 2019a) – which indicates the exact gearbox tank that was shut down owing to the fault), and NLG models generally lack in ability to learn these nuances, wherein, such symbols potentially contribute as noise to the natural language alarm message phrases.

Besides the issue of lack in contextual information on failures being vital to train NLG models, some other areas can also potentially benefit from adequate availability of such types of data. This can, for instance play a pivotal role in influencing the O&M planning strategies for offshore vessel transfers, wherein, an improved availability of fault contexts can facilitate maintenance personnel to better anticipate the required spared parts they may need to carry. This can thereby pave way to improved inventory management and planning for the wind energy sector. Other potential uses of such failure information can include, for instance, the inclusion of thorough details in the form of detailed service logs for turbine maintenance, which can play an instrumental role in contextualisation of rarer fault types and new errors and/or operational inconsistencies witnessed by maintenance personnel during O&M. All such aspects thus contribute directly to human-intelligible and informative decision support, which can be integral to O&M during data-driven CBM and performance assessment/analysis in the wind energy domain.

Below, we provide an outline of some key areas on which the wind industry may focus for tackling challenges pertaining to availability of data and quality issues:-

- **Widely encouraging wind farm operators to make datasets openly available:** The simplest essence of utilising AI models is to leverage more *useful* data, which helps to train models across more diverse scenarios of turbine operation. While some organisations have already started taking the positive steps in making SCADA datasets (and in some situations, also historical alarm records) openly available (as previously described in Table 2.2), this clearly is not sufficient to train AI models at present for making them more robust (and truly autonomous), particularly DL techniques which generally rely on higher amounts of data (Najafabadi et al., 2015) for tuning the models and their parameters.

  If few operators in the wind industry have made their data publicly available, what stops most other wind turbine operators to share their SCADA datasets (alongside the failure and alarm records) to be utilised for Research & Development purposes? Kusiak (2016) mentions that *competition* is the primary reason behind this reluctance, as the sensitive information obtained from turbines can often reveal their technical specifications and performance metrics, as well as expose prevailing poor design practices. While this indeed leads to a major challenge, we believe that multiple options could otherwise be explored by wind farm operators to tackle this, which may include development of non-disclosure agreements and anonymising particularly sensitive information (such as extremely detailed technical metrics and specifications). While this may not address the challenges in shared benchmarks and replicability of research mentioned earlier, such data can be useful to promote greater awareness and application of data-driven decision making models in the wind industry. Also, as turbines generally suffer degradation and after their end of useful life, they are decommissioned (Topham and McMillan, 2017), we believe that such historical data from decommissioned (which would thereby be non-operational) turbines can play a vital role in facilitating development and helping train AI models for the purposes of experimentation and tests. These models can then be utilised in the future by adapting to new sources of information when they become available with time, while at the same time not falling into the challenging constraints of being commercially sensitive.

- **Encouraging wider adoption of transfer learning techniques to leverage insights from any sources of available data:** It is often a highly challenging task in ML to obtain training data which can be utilised for the development of high-performance learning models that can match the prevailing feature space distributions of test datasets (Weiss et al., 2016). This makes it integral to develop learning models tailored to tasks

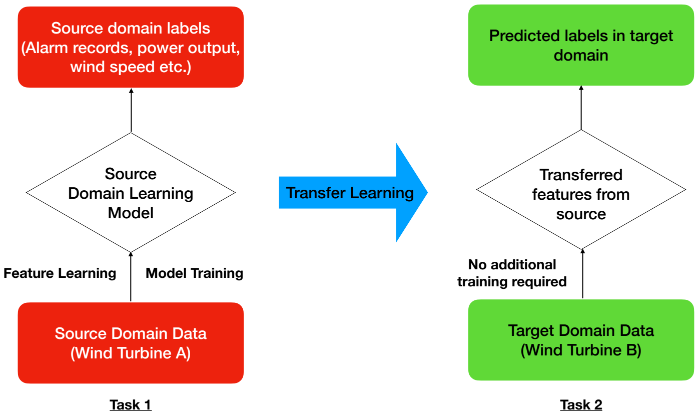

Source domain labels (Alarm records, power output, wind speed etc.)
Source Domain Learning Model
Feature Learning
Model Training
Source Domain Data (Wind Turbine A)
Task 1
Transfer Learning
Predicted labels in target domain
Transferred features from source
No additional training required
Target Domain Data (Wind Turbine B)
Task 2

Fig. 2.15 Description of the typical knowledge transfer process – data from the source domain can include SCADA features, alongside meteorological parameters and unstructured data pertaining to maintenance manuals etc.

in the target domain by training AI models on closely related tasks in the source domain, as shown in Figure 2.15. The wind industry has witnessed very limited applications of transfer learning techniques in comparison to the significantly notable advances made in other areas such as computer vision and NLP (Hussain et al., 2019; Ruder et al., 2019). There are only few studies in the wind industry which focus on leveraging transfer learning with SCADA datasets from turbines, and their primary focus is in the area of predicting wind power (Qureshi et al., 2017) for performance assessment/analysis tasks. Other prevailing studies utilise transfer learning for predicting short-term wind speed (Hu et al., 2016) and assessing the presence of ice on turbine blades (Zhang et al., 2018a). However, the area of predicting faults in wind turbines (being an integral aspect in O&M) by leveraging transfer learning has witnessed scarce application to the best of our knowledge. The only works at present in this domain either specifically focus on predicting faults across multiple turbine sub-components (Chatterjee and Dethlefs, 2020e) or towards monitoring vital metrics and parameters (such as from the gearbox) to identify anomalies based on deviation from normal behaviour (Pan et al., 2021).

It is extremely important for the wind industry to focus on applying AI techniques to achieve more fine-grained analysis as well as prediction of turbine failures through transfer learning, by leveraging historical records of alarms, work orders, operator manuals etc. Moreover, as these types of records are generally the most difficult to acquire for researchers (as described earlier) and also equally challenging for engineers to regularly annotate, transfer learning can prove to be beneficial in the development of high-performance learning models even under situations wherein sufficient amounts of training data are not available. We envisage that by widely adopting such techniques in the wind energy domain, the uptake of AI in tasks pertaining to CBM can see significant enhancement and also help in making the generally complex O&M tasks more dependable under circumstances with paucity of data.

- **Ensuring quality control of datasets in the wind industry:** Some studies have outlined the necessity to ensure quality control of datasets in the wind industry, especially through standardisation of information and development of unified standards and taxonomies by turbine operators and manufacturers (Leahy et al., 2019; van Kuik et al., 2016). These are vital aspects for facilitating development and deployment of highly sophisticated AI models for O&M in the long-term. Based on our reviews in this chapter, we believe that there are multiple possible options which could be explored for ensuring better quality of data. The most basic option is in providing necessary training to engineers and technicians in the wind industry, wherein wind farm operators can help them understand and establish a standardised pathway for annotation, analysis and interpretation of domain-specific information pertaining to operational conditions of turbines by following a common framework or certain industry standards that could be explored for O&M with a consensus established from multiple turbine operators across the globe. As a second potential option, it would be very useful to encourage the adoption of data science and analytics widely in the wind energy domain, by supporting engineers and technicians in such tasks via specialised resources (such as software applications consisting of interactive Graphical User Interfaces (GUIs) that can enable simplified storage, annotation and analysis of datasets including SCADA features, failure logs and alarm records in the wind industry), alongside providing them with the required guidance and insights from data scientists on tackling common challenges in applying data science methodologies. While significant investments have been made within the wind industry in some critical areas (such as design and manufacturing of turbines etc.), there has been limited focus on investing in monitoring, development and analysis of datasets in our view. We envisage that adoption of unified standards and more investments in this area can help the wind industry to greatly benefit from the Return on Investment (ROI), that can help in training AI models for the purpose of decision support by utilising high quality datasets, thereby making such models potentially a more informative source of decision making, and also ensuring their accuracy and scalability.

- **Utilisation of specialised statistical and AI techniques to overcome the challenges posed by data quality:** Datasets obtained from wind turbines generally contain noise and outliers (such as power being produced at zero wind speed) which result from communication failures, abnormal equipment operations etc. (Wu et al., 2020a), posing significant challenges in utilisation of such information for training AI models. Some studies have demonstrated that the application of specialised techniques for the removal of noise and outliers can help in making the datasets more efficient, versatile and enhance their suitability for data analytics and information mining. In likely the earliest demonstration of such statistical methods applied for robust filtering of datasets, Llombart et al. (2006) utilised the Least Median of Squares (LMedS) methodology for detecting noise and outliers present in turbine power curves. The paper showcases that this approach outperforms other traditional statistical techniques used for filtering (such as those based on mean and standard deviation of binned segments of data), and can thereby help in eliminating the requirements for manual filtering towards removal of outliers. A similar study by Sainz et al. (2009) focused on combining the LMedS method with a specialised random search technique, which can provide the ability to filter the modelled data based on additional parameters besides wind speed – considering metrics such as wind direction etc. In a more recent study, Shen et al. (2019) outlined that specialised algorithms for data filtering, such as change point grouping and quartile algorithm can help to improve the data quality pertaining to power curves by leveraging the outlier distribution characteristics for this purpose. Some other studies in this domain have utilised statistical filtering techniques, particularly those which have been classically popular in statistics and control areas, such as Kalman filters for localisation of noise and outliers during wind energy assessment (Melero et al., 2012), but such techniques are generally complex and rely on extensive mathematical modelling for any potential success. While the prevalent approaches for data filtering in the existing studies are clearly well-suited to traditional AI models, especially those focused on regression tasks, they cannot handle other complex challenges posed by imbalanced datasets and issues pertaining to lack of sufficient contextual information on failures, which we have discussed before in this chapter.

  For handling imbalanced datasets in the wind industry, there have been successful applications of oversampling methods like Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002) in some studies (Chatterjee and Dethlefs, 2020e; Ge et al., 2017; Yi et al., 2020). SMOTE is a popular statistical algorithm that can help to generate synthetic data points for samples in datasets which belong to the minority class (such as labelled historical records of anomalies), thereby facilitating a better overall balance between the majority class (e.g. normal operation instances) and minority classes prevalent in the dataset. There are some other types of oversampling techniques which have been popular in the AI community, particularly Adaptive Synthetic Sampling (ADASYN) and Ranked Minority Oversampling in Boosting (RAMOBoost) (Sáez et al., 2016). In some situations, these can be more efficient, but they are yet to be applied in the wind energy domain for CBM to the best of our knowledge. We believe that it is essential for the wind industry to particularly focus on adoption of oversampling techniques at a larger scale, to ensure informative decision making even under circumstances wherein the data is limited and imbalanced.

For facilitating greater uptake of NLG models, it is integral to focus on enhancing the diversity of contextual failure information in the wind industry, which would signify that the focus needs to not only be given to critical fault types which have been discussed in our reviews, but also on other low-priority faults. Some other options are also potentially viable for facilitation of human-intelligible O&M policies to be generated under situations with inadequate quality of data. Firstly, it is integral for the turbine alarm logs and maintenance records to be better organised, and ensure regular updation of such information. As a second potential option, specialised NLG models and techniques like those based on few-shot learning (Chen et al., 2020c) can be utilised for facilitating NLG in situations wherein there is lack of sufficiently high quality data for training AI models. Thirdly, there is likely immense potential for the wind industry to utilise generalised language models which have already been pre-trained with large corpuses consisting of billions of parameters, such as Bidirectional Encoder Representations from Transformers (BERT), OpenAI's Generative Pre-trained Transformer (GPT-2/GPT-3) etc. – which have achieved state of the art performance in multiple downstream tasks for NLP and NLG (Devlin et al., 2019; Shoeybi et al., 2020). Such models are capable of being fine-tuned with customised domain-specific information obtained from small corpuses and can help to overcome the major drawbacks faced due to inadequately available high-quality data in the wind industry, as well as ensure more fine-grained, contextualised and comprehensive descriptions for failures and O&M strategies for fixing/averting faults when compared to only brief alarm messages that are presently prevalent in this domain.

Table 2.4 Emerging challenges for data quality in the wind energy domain. These affect the development of highly sophisticated AI models and can likely be mitigated by adopting specialised strategies during data curation & pre-processing.

Challenge in data quality	**AI techniques that are affected**	**Possible solutions to overcome the challenge**
Imbalanced datasets	Supervised learning techniques for classification of faults; Reinforcement learning techniques for O&M planning; NLP techniques for classifying alarm messages	Utilising oversampling techniques for balancing imbalanced class distributions (Sáez et al., 2016) e.g. Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002), Adaptive Synthetic Sampling (ADASYN) (Haibo He et al., 2008), Ranked Minority Oversampling in Boosting (RAMOBoost) (Chen et al., 2010), Deep Convolutional Generative Adversarial Networks (DC GAN) (Xie and Zhang, 2018) etc.
Lack in diversity of alarm messages	NLG techniques for generating contextual information on faults	Few-shot learning techniques (Chen et al., 2020c) to learn from low-diversity and limited training data; Generalized language models pre-trained on large corpuses of information such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), Generative Pre-trained Transformer (GPT-2/GPT-3) (Brown et al., 2020; Radford et al., 2019) etc.
Low-quality datasets with noise, corrupted values and outliers	Supervised and unsupervised classification (for fault prediction) and regression (for forecasting vital operational parameters) techniques; Reinforcement learning techniques for O&M planning;	Change-point grouping and quartile algorithms (Shen et al., 2019), Least Median of Squares (LMedS) method (Llombart et al., 2006), LMedS with random search (Sainz et al., 2009), statistical and control filtering techniques like Kalman filters (Melero et al., 2012), specialised loss functions in DL, data re-weighting and training procedures (Karimi et al., 2020), class noise and attribute noise identification techniques (especially ensemble-based noise elimination) (Gupta and Gupta, 2019), specialised ML-based noise reduction techniques such as Multi-step finite differences, Splines, Mixture of sub-optimal curves etc. (Minutti et al., 2018)

Missing values in datasets	Supervised and unsupervised classification (for fault prediction) and regression (for forecasting vital operational parameters) techniques; Reinforcement learning techniques for O&M planning; NLP techniques for classifying alarm messages; NLG techniques for generating contextual information on faults	Statistical imputation techniques (Jerez et al., 2010) to replace missing values with substituted values based on other available data e.g. using measures of central tendency like mean/median, K-Nearest Neighbours (KNN) imputation (Pan et al., 2015), Self-Organising Maps imputation for incomplete data matrices (Folguera et al., 2015) etc.

While some safety-critical disciplines like clinical decision support (Demner-Fushman et al., 2009) have reaped significant benefits from the better explainability and context that natural language messages (and reports) can provide, there is a pressing need for the wind industry to focus on optimal utilisation of all types of relevant datasets which can be useful to facilitate decision making, especially under constraints posed by complex, unorganised and low-quality datasets for O&M records in the short-term. This is vital until there are better quality datasets widely available in the wind energy domain, which, to our strong belief, can only be achieved in the long-term owing to the challenges pertaining to transition from traditional methods for data acquisition to high-quality storage and information retrieval systems, such as cloud data centres. Alongside these problems, the wind industry also needs to deal with the constraints of high-resolution SCADA data (such as those at 1-second intervals), missing values and outliers, rapidly changing events and alarm information etc., which is particularly essential for facilitating real-time decision support. Table 2.4 provides a summary of the emerging issues pertaining to data quality that are starting to pose significant challenges in the wind industry in recent times, alongside the possible strategies which can be adopted to facilitate informative decision making during O&M under such constraints.

2.4.2 Challenges in deployment of AI models for real-time decision support in O&M

While the reviews in this chapter have outlined some positive steps taken in real-time deployment of decision making models in the wind industry (Abdallah et al., 2018b; Canizo et al., 2017; Yang et al., 2019), these are clearly limited compared to the significant research that has been pursued in developing AI models in offline environments. More importantly, the existing studies in the area of real-time deployment of decision making models mostly focus on more traditional and simpler ML models like decision trees, random forest, SVR etc., which despite being promising are clearly not sufficient to address the present demands of the wind industry experiencing rapid growth in the complexity of big data.

As the domain of Internet of Things (IoT) continues to become popular in its global advent, some studies have outlined techniques which could be helpful in interfacing turbine sensors and actuators with cloud frameworks and the internet. An early study in this domain by Kalyanraj et al. (2016) proposed the utilisation of IoT for turbine control, alongside logging vital parameters like power generation and vibration levels etc. from data. Another notable study in this area by Alhmoud and Al-Zoubi (2019) proposed a framework towards utilising IoT platforms for individual turbines in wind farms, wherein, they can be connected to a cellular network by utilising microcontrollers. Any new data which thereby becomes available can be saved and processing can be performed directly on cloud servers, facilitating access to such information from anywhere across the globe via devices like computers and mobiles, wherein, suitable commands can also be specified to the turbines in wind farms. Despite being a promising framework for gathering real-time information from turbine sensors towards performance optimisation, alongside its valuable potential in determining optimal O&M activities, the paper outlines the key factors which limit practical realisation along with wider deployment of IoT in the wind energy domain – lack of budget and necessary skills and training. Other barriers outlined in the paper include security concerns, challenges pertaining to communication protocols etc. While few such studies clearly showcase the immense potential of IoT in facilitating real-time decision making, they do not have a specific focus on AI and the challenges faced in utilising and deploying the models for O&M tasks.

We believe that at present, the availability of continuous flow of information (such as in the form of SCADA features from sensors) is not a key challenge in achieving real-time decision support, as most wind farm operators have recently placed a significant emphasis on the development of efficient data logging techniques and information processing systems on cloud servers, as outlined by the reviews conducted in this chapter. However, the major challenge which the wind industry continues to face is the exponential rise in complexity of such datasets, wherein, lack of unified standards and simplified formats make these sources of information difficult to be utilised for inference with trained AI models. Also, as higher resolution data continues to become increasingly available instead of the traditionally popular SCADA information at 10-minute intervals, additional constraints arise in feeding SCADA features to the AI learning models, particularly for facilitation of continual learning (online updates) and re-training the model with new sources of data as they become available.

We believe that the pressing issues which presently hold the wind industry back in utilisation of AI models (in particular deep learners) for the purpose of real-time decision making are primarily attributed to inadequate computing power and resources, rising memory costs and major security/privacy concerns. While the scale and complexity of these challenges can be reduced by utilisation of smaller models (such as deep learners with fewer hidden layers), this would also generally limit the model's ability to learn in complex tasks, as the capability of such networks to go deeper or wider in nature is essential to ensure high performance, particularly with large-scale DNN models (Jin et al., 2018). Thereby, only utilising simpler/shallower models to overcome this challenge is clearly not a good solution, as it would generally lead to a significant trade-off in terms of computational costs and the models' performance, which is often a critical factor in O&M tasks for the wind energy domain.

Table 2.5 Emerging challenges for real-time deployment of AI models in the wind energy domain. Potential strategies to overcome such challenges are outlined.

Real-time deployment challenge	**Description**	**Possible solutions**
Memory cost constraints	AI models, especially deep learners face high bandwidth memory requirements; Memory usage during training is especially dominated by the need for intermediate activation tensors for storing temporary information during backpropagation (Jain et al., 2020)	In-memory computing to perform forward and backward pass in neural networks in place without the requirement to move around weights (Eleftheriou et al., 2019) etc. , memory efficient DL frameworks for large-scale data mining (Nguyen et al., 2019) e.g. MXNet (Chen et al., 2015), memory efficient adaptive optimisation method (Anil et al., 2019), specialised frameworks for developing memory efficient invertible neural networks e.g. MemCNN in PyTorch (Leemput et al., 2019), automatic efficient management of GPU memory by using specialised techniques e.g. computational graphs for models with swap-out and swap-in operations for holding temporary results in CPU memory (Le et al., 2019) etc.

Inadequate computing power	Recent advances in AI, especially DL have led to models which often utilise tens of millions of parameters for tasks pertaining to real-time processing of data streams (Vanhoucke et al., 2011); DL models require substantial computational resources during training and inference phases to run in a quick manner (Chen and Ran, 2019)	Using dedicated hardware for ML with High Performance Computing (HPC) platforms such as Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) (Chen et al., 2020b; Steinkraus et al., 2005; Wang et al., 2019), cost-efficient training mechanisms for fast model training such as PruneTrain (Lym et al., 2019), resource constrained structure learning for deep networks (Gordon et al., 2018), utilising edge computing techniques during DL to accomplish low-latency and high computational efficiency (Chen and Ran, 2019); using cloud computing platforms (He et al., 2018) such as Google Cloud AI, Amazon Web Services, Azure Machine Learning, IBM Watson Machine Learning etc.
Concerns on communication security and privacy	Real-time decision support systems face risks of security concerns during data streaming; ML model policies can be interfered with malicious attacks when performing real-time control in dynamic environments (Clark et al., 2018); AI models can be subject to adversarial attacks during training/testing phases (Qiu et al., 2019); Wind farm SCADA systems can be subject to cyber attacks and intrusion (Yan et al., 2011; Zabetian-Hosseini et al., 2018)	Secure learning approaches for defense against training and inference time attacks (Papernot et al., 2018), encryption and secure coding of data streams e.g. through Low-density Parity Check (LDPC) (Jang et al., 2017) etc.

Table 2.5 summarises some possible strategies which can prospectively be utilised in the wind industry for overcoming the rising challenges in deployment of highly sophisticated AI models, especially deep learners for the purpose of real-time decision making.

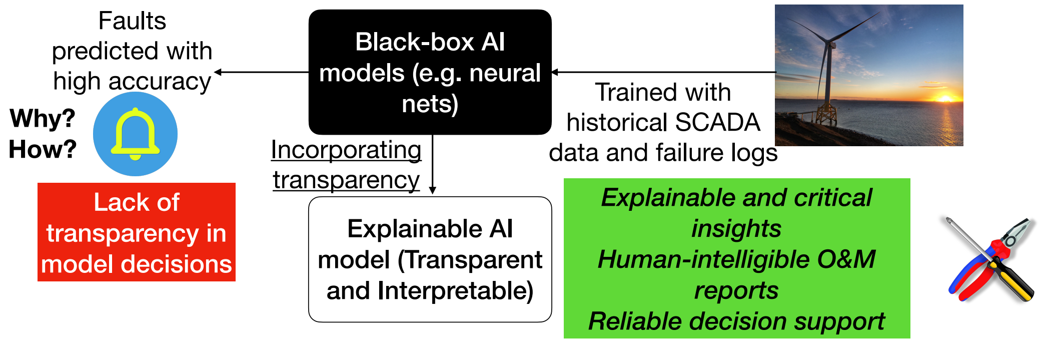

Faults predicted with high accuracy
Why?
How?
Lack of transparency in model decisions
Black-box AI models (e.g. neural nets)
Trained with historical SCADA data and failure logs
Incorporating transparency
Explainable AI model (Transparent and Interpretable)
Explainable and critical insights
Human-intelligible O&M reports
Reliable decision support

Fig. 2.16 An outline of the explainability challenge for AI in the wind energy domain – while black-box AI models are able to make predictions with high accuracy, they cannot provide effective reasoning and rationales behind their predictions. Explainable AI models can help to overcome such challenges.

2.4.3 Lack of transparency in decisions made by black-box AI models

As is evident from the previous discussion, although most AI models are capable of generating predictions with high accuracy (such as predicting different types of faults and forecasting turbine power output), they continue to witness a significant challenge in providing transparency in their decisions owing to their inherently black-box nature. This prevailing challenge is outlined in Figure 2.16. Note that while conventional ML models (like decision trees, for instance) are capable of providing transparent predictions which can be more easily interpreted (Sezer et al., 2019), they are generally outperformed by deep learners significantly in most real-world tasks. We observe that wind farm operators are reluctant to widely adopt such data-driven decision making models and thereby focus more on traditional methods which utilise signal processing and numerical physics-based models. We believe that incorporation of trust into decisions made by the black-box learning models is essential for the wind industry, which can be achieved by switching from black-box AI techniques to transparent Explainable AI models as discussed further:-

Table 2.6 A brief description of Explainable AI models which are relevant to CBM of wind turbines. There are some models which have not been leveraged to date, for which we outline their prospective applications.

Explainable AI Model	**Description**	**Applicability to wind turbine CBM and performance assessment/analysis tasks**
Explainable Deep Neural Networks (xDNN) (Angelov and Soares, 2020)	A non-iterative and non-parametric DL architecture, combining reasoning and learning in a synergy. Provides explanations based on probability density function automatically learnt from training data distribution	Can prospectively be applied towards fault prediction in turbine sub-components, anomaly detection in blade images, predicting vital SCADA parameters (e.g. wind speed and power output)

Long Short-Term Memory Networks (LSTMs) with attention (Luong et al., 2015b)	The attention mechanism allows LSTMs to focus on vital parts of input sequences, providing easier and higher quality learning; Attention weights can provide transparency in key features which cause LSTM to generate its predictions	Wind power prediction (Wang et al., 2019)
Convolutional Neural Networks (CNNs) with attention (Zheng et al., 2017)	The attention mechanism provides ability to focus on vital segments of input data in the CNN layers and convolutional filters, with attention weights providing explainablity in predictions.	Short term wind speed prediction (Shivam et al., 2020), imbalance fault detection in turbine blades (Chen et al., 2019a), causal inference for discovering novel insights and hidden confounders (Chatterjee and Dethlefs, 2020c)
SHapley Additive exPlanation (SHAP) (Lundberg and Lee, 2017a) + Any Black-box AI model	Provides explainations for outputs generated by any ML model based on local explainations through game theory approach ; provides force plots and interpretable explanations of decision trees/ensembles of trees	Fault prediction in multiple turbine sub-components (Chatterjee and Dethlefs, 2020b)
Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) + Any Black-box AI model	Provides local linear explainations for the ML model's behavior; can be utilised for explainable classification tasks with two or more classes	Prospective applications include explainable binary/multi-class anomaly prediction in turbine sub-components, classification of blade images, segmentation of alarm messages
Sequence-to-Sequence (Seq2Seq) model with attention (Luong et al., 2015a)	Specialised recurrent neural network architecture for sequential data; incorporates attention mechanism to focus on vital parts of input sequential data; can provide transparency in identifying relevant features used during the prediction process	Wind power forecasting (Fu et al., 2019), prediction of alarm messages (Chatterjee and Dethlefs, 2020d)
Transformers (Vaswani et al., 2017b)	Utilises multi-head attention mechanism and removes recurrence in the conventional encoder-decoder Seq2Seq architecture; can provide transparent decisions in terms of key features which lead to generated predictions through self-attention scores ;more computationally efficient than Seq2Seq models	Short-term load forecasting (Meng and Xu, 2019), prediction of alarm messages and maintenance actions (Chatterjee and Dethlefs, 2020d)

eXtreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016a)	A novel, sparsity-aware tree boosting algorithm which can provide feature importances in datasets utilised for predictions; highly computationally efficient and scalable	Fault detection in multiple subcomponents (Chatterjee and Dethlefs, 2020b,e; Wu et al., 2020b; Zhang et al., 2018b), wind power forecasting (Browell et al., 2017), gearbox fault prediction (Yuan et al., 2019)

Adopting Explainable AI models in the wind industry: To tackle the challenges posed by lack of transparency in black-box natured AI models, Explainable AI (XAI) (Barredo Arrieta et al., 2020) can provide highly promising avenues to achieve trustworthy decision making. XAI can bring improvements in the performance of AI models as explanations can help trace issues, challenges and pitfalls induced in datasets and the prevailing behaviour of the features, while also assisting the engineers and technicians to instil better trust and confidence in the AI model's predictions. The wind industry has witnessed very limited applications of such XAI techniques, wherein, only few studies apply XAI models for the purpose of CBM and performance assessment/analysis. At present, most applications typically focus on prediction of turbine power output, while there has been rather limited attention given to the area pertaining to prediction of faults and maintenance action strategies during O&M. Table 2.6 provides a comprehensive summary of some prominent XAI models which have seen utilisation in the wind energy domain, alongside other models which can potentially be utilised in future applications.

Figure 2.17 outlines the trends in utilisation of Explainable AI models for data-driven decision making in the wind industry, wherein, the slow growth in this area is clearly evident. As can be seen, while explainable decision tree models (such as XGBoost), specialised packages and libraries (e.g. SHAP in Python) to incorporate transparency and CNNs incorporated with attention mechanism have witnessed comparatively greater focus in the wind industry, there are several other types of techniques, particularly those which utilise LSTMs with attention mechanism, Transformers and Sequence-to-Sequence (Seq2Seq) models etc. which are facing paucity in terms of being applied for data-driven decision support. This can likely be attributed to the lack of sufficient insights on XAI, as well as a prevailing reluctance in applying AI within the wind industry, which we hope this thesis can help tackle. We are optimistic of the immense potential which such models hold, in particular by leveraging NLG – owing to the fact that besides providing accurate predictions (such as for alarm messages in turbines), these models can also identify feature importances for the contributing

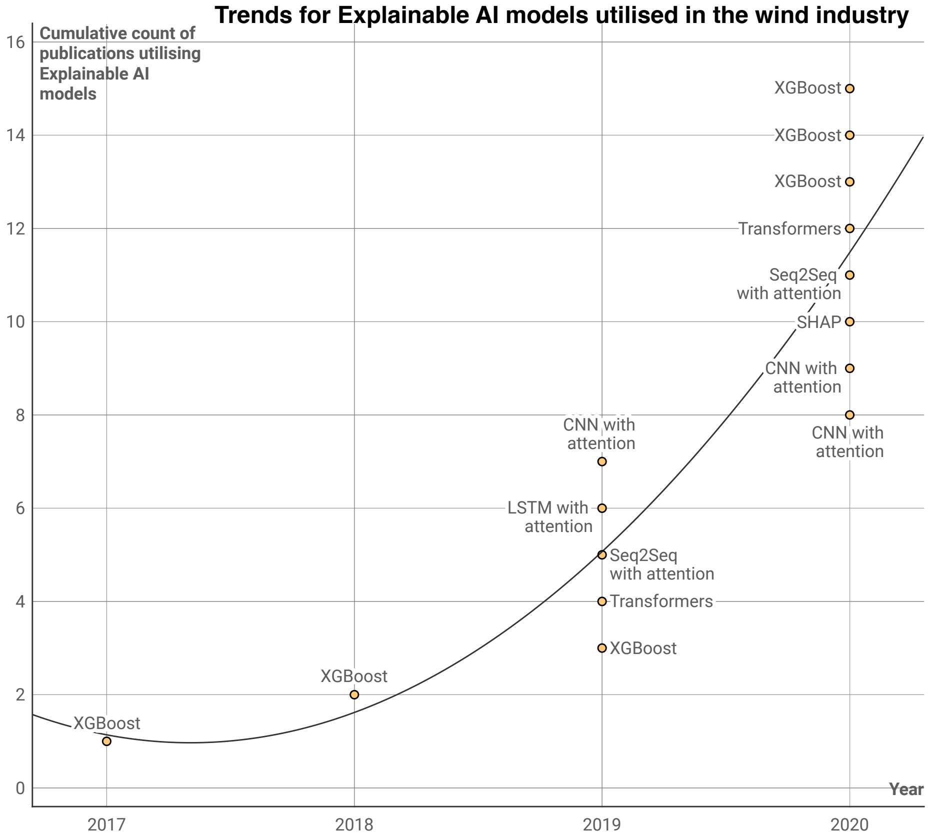


Fig. 2.17 Trends in utilisation of Explainable AI models in the wind industry for decision support. There is a clear evidence of slow and static growth in this domain.

causes leading to the failures, alongside suggesting comprehensive maintenance actions in a human-intelligible format towards fixing/averting failures in such circumstances (Chatterjee and Dethlefs, 2020d). Besides the areas outlined in this chapter, we believe that there are possibly a plethora of other applications in utilising XAI for data-driven decision making during CBM and performance assessment/analysis, which need to be explored in the wind industry in the near future to facilitate transition to human-intelligible diagnosis of failures and prognosis for operational inconsistencies occurring in turbines.

2.5 Discussion

Figure 2.18 presents a graphical roadmap which summarises the likely viable future in utilisation of AI models for decision support in O&M in the wind energy domain over the next five years. This roadmap is based on our own perspective and interpretation of the scientometric reviews conducted in this chapter – and is subjective in nature. Note that while

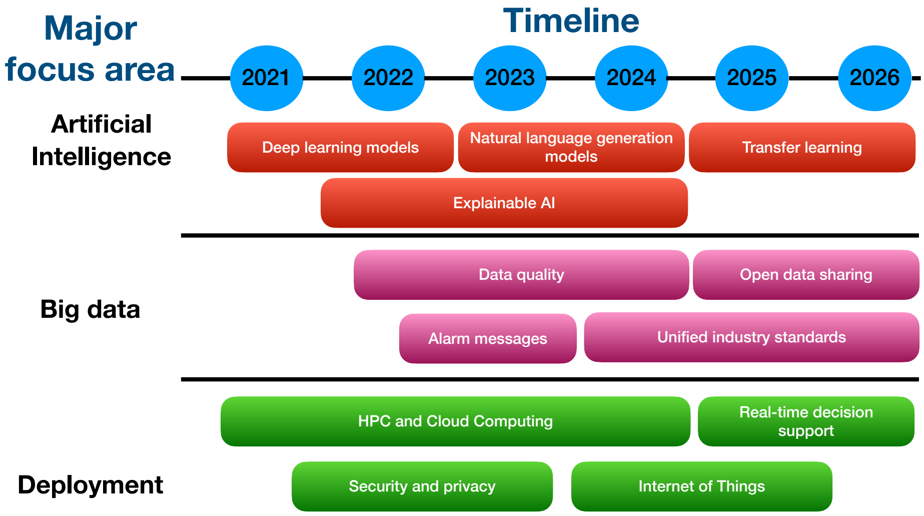


Fig. 2.18 Graphical roadmap based on our own perspective – depicting the likely *major* focus areas for the wind industry in the future on the basis of current trends, successes and challenges. When a specific topic first appears, it would likely receive majority focus during that time period.

it is impossible to be certain of the definite future ahead, the current successes and challenges which have been described in this chapter along with the rise in utilisation of AI in the wind industry shows that there is likely a highly promising future along these directions. More details on the enunciated focus areas are shown in Table 2.7.

The roadmap presented above has been segmented into three major subgroups – Artificial Intelligence (pertaining to the general practice of adopting AI in the wind industry), Big Data (pertaining to tackling issues in data quality and availability) and Deployment (pertaining to end deployment of decision making models in real-world practical scenarios in the wind industry), wherein, each of these specific subgroups consist of multiple relevant topics (like DL within the AI subgroup). It is integral to note that the roadmap enunciates the major areas of focus for the wind industry in the next few years, and the first appearance of new topics in this roadmap (such as data quality in the year 2022) indicates the likely major priority which this area would receive at that time. Once the highlighted topics start to disappear from the roadmap (for instance, data quality is assigned a timeframe only till the year 2024), these topics would continue to be important for the wind industry, however, with a likely lesser priority. These insights are based on the general delay which prevails in between the time of first publication of associated methods in the AI community to the time when such AI models are utilised in the wind industry for the purpose of data-driven decision making. For instance, the Transformer model utilised in NLG tasks (Vaswani et al., 2017b) was first published in

Table 2.7 Summary of likely viable future roadmap and the major focus areas in the wind industry for adoption of AI for data-driven decision making

Major focus area	**Description**
Deep Learning models	Wider adoption of DL models for CBM (especially RNNs and CNNs)
Explainable AI	Growing focus on XAI, with transition from XGBoost to other transparent learners (e.g. transformers with attention mechanism)
Alarm messages	Better availability of alarm messages with contextual information of faults
Natural language generation models	Uptake of NLG models in "few" wind farms for real-time human intelligible decisions
Data quality	Better quality data available with improved sensors and automated processing of alarms on cloud
HPC and Cloud Computing	Wider adoption of HPC in the wind industry, especially GPUs and TPUs through cloud computing
Transfer learning	Transfer learning sees immense utilisation with rise in development of new wind farms which lack sufficient operational data to train AI models from scratch
Open data sharing	Increased open-sharing of datasets for CBM, especially with historical alarm and failure information
Unified industry standards	Unified industry standards developed in the wind industry pertaining to standard taxonomies and data formats
Internet of Things	IoT becomes widely popular in the wind industry, with uptake for real-time control and O&M of wind farms from anywhere in the world
Security and privacy	Security and privacy challenges are significantly overcome through adoption of AI models after critical testing for adversarial attacks; SCADA systems become more secure to intrusions
Real-time decision support	Highly sophisticated AI models are deployed widely by wind farm operators across the globe for real-time decision making in O&M activities

the AI community in the year 2017, but it only saw applications for O&M tasks in the wind industry for short-term load forecasting (Meng and Xu, 2019) in 2019 and in generating human-intelligible alarm messages (Chatterjee and Dethlefs, 2020d) in 2020. Clearly, the adoption of such methods generally takes 2-3 years after the time when they first appear in the AI community, wherein, this time delay is an important factor to consider.

It can also be enunciated from the roadmap that while it is envisaged that a transition to DL and NLG models would take place in the short-term, alongside rising focus in adopting XAI models for transparent decision making, the wind industry would likely take more time to progress in the aspects of data availability and quality and adopting HPC & Cloud Computing techniques more widely. It is also highly likely for transfer learning techniques to witness an immense growth in utilisation in the near future, as the competitive nature of the wind industry can lead to a high possibility of wider open data sharing in the next few years, given that few wind farm operators have already taken the positive steps required for sharing such datasets openly. However, we believe that adopting IoT for real-time decision support would not likely be possible in the short-term, as the ongoing challenges and concerns relating to data security and privacy require careful consensus and through analysis in the wind energy sector.

If the current trends in growth towards utilisation of AI models in data-driven decision making continue at the same pace in the wind energy domain, we believe that achieving real-time decision support across most wind farms globally would not be an impossible feat, particularly by the next five years (year 2026). This, surely accounts for cautious optimism and the possibility of an AI winter to again prevail in the wind energy domain. However, we enunciate that by following the pathways in the roadmap to the best possible, wind farm operators can likely save immensely on O&M costs, and thereby facilitate wind energy to be adopted globally for tackling climate change.

2.6 Conclusion

In this chapter, we have provided a systematic literature review of the past, present and future of applying data-driven decision making techniques in the wind industry by leveraging scientometric analysis through statistical computing techniques. By deriving evidence-based insights to trace the thematic and conceptual evolution of CBM and performance assessment/analysis, we demonstrate the significant rise in interest towards leveraging AI techniques for decision support, particularly deep learners. An important observation from our literature review is that despite the rapid rise in AI in the wind energy domain, more traditional methods like those utilising signal processing will likely continue to complement the AI models during rapid phases of transition. This chapter also shows that AI applied in the wind industry is still in its embryonic stages compared to some other disciplines like NLP and computer vision, which have made significant advances in this area.

We have demonstrated the key challenges which the wind industry is facing in utilising data-driven decision support techniques more widely, which, in particular, are attributed to challenges in availability of quality data, concerns in deploying AI models for real-time decision support and limited transparency in the decisions made by the black-box natured AI models. For combating these challenges, we have outlined the vital need to shift focus towards more sophisticated and tailored AI models, in particular those utilising DL and NLG for supporting Explainable AI towards achieving human-intelligible and trustworthy decisions. We envisage that the literature review conducted in this chapter can play an integral role in encouraging researchers in the wind industry to specifically focus on the outlined critical areas which have mostly been neglected in the past studies, which can help facilitate smoother transition to AI from academic labs at present to the wind industry in the near future. This thesis would thereby focus on some of these critical areas, and we would leverage Explainable AI models to achieve more fine-grained and transparent decisions in CBM. In the next Chapter (Chapter 3), we will discuss exploratory data analysis and pre-processing of real-world SCADA datasets in the wind industry, which will eventually be utilised for experiments with Explainable AI models in the later chapters.

Chapter 3

Exploratory Data Analysis and Pre-processing

> Hiding within those mounds of data is knowledge that could change the life of a patient, or change the world.
>
> Atul Butte, Stanford

3.1 Introduction

In ML, the key step to understanding and applying various algorithms and models for experimentation requires *useful* data, which is easy to comprehend, analyse and process computationally. A first look at the data at hand can help answer many questions, and visualising and statistically analysing the data can pave the way to make any further experiments potentially more successful. Such complex information can then be utilised for teaching machines to perform tasks and make decisions automatically, with minimal or no human intervention (Malgaonkar et al., 2016; Oussous et al., 2018; Sivarajah et al., 2017; Tannahill and Jamshidi, 2014). An appropriate quantity and quality of data is vital to ensure optimal data-driven decision making for real-world applications (Foresti et al., 2020). Figure 3.1 describes the process of transitioning from raw data to useful insights, by processing and extracting relevant patterns based on domain knowledge, wherein ML models play an integral role in discovering patterns in the datasets (Wang and Blei, 2019).

AI techniques can help process massive amounts of data for discovering novel insights and actionable knowledge (Ali et al., 2016; Bhatnagar, 2018; Qiu et al., 2016). At present,

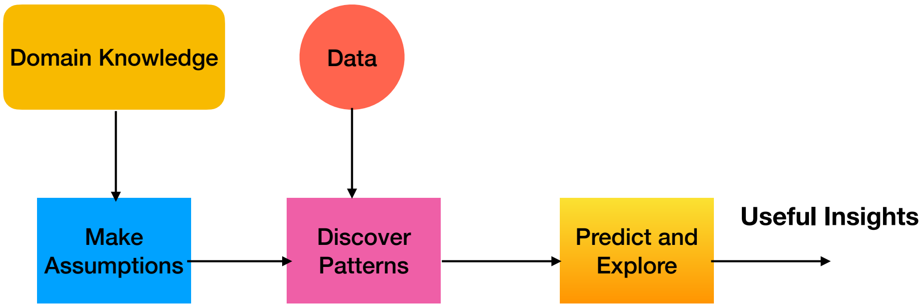


Fig. 3.1 Transitioning from raw data to useful knowledge - ML models help to discover *useful* patterns in big data

most wind turbines and wind farms consist of a multitude of sensors within their various sub-components and sub-systems (Shearer, 2019), which record the operational status of the turbines and the different alarms raised during faults. Interestingly, other than the direct internal status of the turbine which is recorded by these sensors, an enormous amount of information is available pertaining to the immediate outside environment in which the turbine operates, such as the wind speed, air pressure, temperature etc. All this information is logged through a data-acquisition system called SCADA (Supervisory Control & Acquisition Data), which contains extremely valuable metrics for planning and maintenance activities (Peharda et al., 2017).

A major challenge with acquiring SCADA data, as also outlined in Chapter 2, is its extremely confidential nature, due to which the turbine operators are generally reluctant to share such information. Also, as different turbine manufacturers and wind farm operators use different taxonomies and structures for recording SCADA variables (Tao et al., 2019), every dataset can vary in terms of the recorded variables and their specifications. In this thesis, we aim to explore and apply AI techniques to SCADA data and demonstrate through case studies the application of such algorithms and the results on such data for data-driven decision making. We approach the experimental process from different perspectives towards O&M of turbines and use two different SCADA datasets in this thesis, each with its own merits as outlined below:-

1. **Levenmouth Demonstration Turbine Operational Data – Primary dataset for our study**:- The SCADA data from the Levenmouth Demonstration Turbine (LDT) [1] is arguably the most important dataset used in this thesis. The importance of the data itself stems from the fact that this dataset contains the most significant information

[1] Platform for Operational Data (POD) Disseminated by ORE Catapult: https://pod.ore.catapult.org.uk

on the turbine operational status and historical faults over time. This is the only dataset utilised in the thesis which contains the alarm logs of all historical faults which have been recorded in the turbine, along with the description of *why* the turbine was stopped, thus making it extremely suitable for supervised learning. By learning from historical faults, ML models can be trained to identify the relevant conditions which most-likely contribute to a fault, and this can be effective in different aspects of preventive, predictive and corrective maintenance planning.

2. **La Haute Borne Wind Farm Operational Data**: We use data from the ENGIE's La Haute Borne Wind Farm [2] as our secondary dataset. This publicly available dataset contains information from the world's first open data onshore wind farm. Although, the information available only includes the SCADA variables recording the operational status of turbines with no availability of historical fault logs, the key advantage of this dataset is its huge size, with abundant information recorded over a three year period from 2013-16. We use this information to support a variety of our experiments and validate our ML models for fault prediction in the lack of labelled fault information through transfer learning.

3.2 Levenmouth Demonstration Turbine (LDT)

The LDT is an offshore wind turbine rated at 7 MW located in Levenmouth, Fife, Scotland. The data from the LDT consists of the following different parameters in different files. A brief description of each of these files is given below for coherence:-

- **Met Mast data**: The Met Mast records information for the wind speed and direction at 10-minute intervals.

- **Substation data**: The turbine substation information is recorded at 10-minute intervals, and consists of variety of information such as reactive power, current and voltage measurements.

- **Alarm data**: This file consists of a log of all historical alarms which have occurred in the turbine and recorded by the maintenance personnel. There are start and end times recorded for each alarm, and a brief event description of the alarm category.

- **Turbine data**: This includes SCADA data from the turbine with a variety of measurements for operational parameters like temperature, pressure and electrical variables.

[2] ENGIE Open Data Wind Farm: https://opendata-renewables.engie.com/pages/home/

- **Processed events**: This consists of processed events with information on faults that occurred in the turbine as well as a categorised functional group for the fault, such as pitch system, gearbox, hydraulic system etc. Notably, the event descriptions with the corresponding alarm messages recorded by maintenance personnel whenever a fault occurred are available in this file.

- **Unknown data and useless data (As labelled by maintenance personnel)**: Alongside all these files, there was a collection of unknown data and useless data, which the maintenance personnel have not been able to identify as being of any application to the operational procedures, and the variables are not recognised at the time of recording. The useless data consists of measurements which are not considered to be of much significance to the wind turbine operators, while the unknown data consists of measurements which are unconventional and difficult to label. It is imperative to mention here that the useless data and the unknown data files were not used for the purpose of our study, as these seem to be redundant with no definite pattern or structure of measurements, and may only contribute marginally to an already high dimensional SCADA data. For this study, we discard all unknown and useless data.

3.2.1 Description of power curve

The power curve (Aly et al., 2016; Lee et al., 2020; Trivellato et al., 2012b) is an extremely important criterion for determination of any wind turbine's performance. The power curve shows the relation between the wind speed and a particular turbine's power output. A wind turbine is generally said to be operating normally when the wind speed is above the *cut-in speed* – which is the minimum wind speed at which the turbine generates power, and below the *cut-out speed* – which is the maximum wind speed, beyond which the turbine should stop operating to prevent breakdown and component failures. The ideal wind turbine operates at the *rated wind speed* – which is the speed at which the turbine generates its maximum (rated) power. Whenever a wind turbine enters regions 1 and 3 of the power curve (i.e. below cut-in wind speed or above cut out speed), it can generally be interpreted as a situation of an operational inconsistency in the turbine. While power curves can help to identify underperformance or faults based on any departure/deviation from expected/ideal values of the fitted curve (Chatterjee and Dethlefs, 2019c; Sohoni et al., 2016), it is vital to consider the fact that not all types of turbine failures and their specific root causes can be indicated by the power curve (as alarms can be raised for incipient faults during anomalous behaviour of specific sub-components e.g. gearbox during high temperatures, even when the turbine operates in normal regions of the power curve) (Chatterjee and Dethlefs, 2020d), which makes consideration of the historical alarm logs for turbines important.

It is imperative to mention here that there can be two key reasons for no power generation at wind-speed above cut-in speed and below cut-out speed (Chatterjee and Dethlefs, 2020e):-

1. **Requested shutdowns:-** When the turbine is shut down due to requests by the local community due to noise curtailment, curtailment from the grid, for maintenance operations etc.

2. **Forced outages:-** These are the events which actually underline a fault within the turbine, due to which the turbine is shut down to prevent any further damage to the sub-components.

It is the forced outages which are important for consideration as events which actually underline a fault or an impending fault in the wind turbine. For this purpose, the events outlining the requested shutdowns are eliminated from the SCADA data. Further, the various data consisting of Met Mast data, LDT Substation Data, Turbine Data and SCADA data with electrical, mechanical and other parameters obtained from the sensors are merged together at time-stamps which are exactly the same for each of the data measurements to facilitate fair comparison and identification of faults. This is thereby used for modelling of the power curve of the wind turbine.

Figure 3.2 shows the power curve based on the 10-minute averages of wind speed (in m/s) and the power output of the turbine (in kW), modelled as a scatter plot before any pre-processing.

3.2.2 Pre-processing of data and labelling of faults

Originally, the data consisted of features which were not in a sorted (ascending/descending) order as per time-stamp. We sorted the data based on the time-stamp in an ascending order to make the analysis more coherent. Also, to facilitate the analysis and to understand the correlation amongst the data, it was merged together corresponding to the instances where the time-stamps were the same for all the files.

There are some interesting observations from the power curve:-

1. There seems to be two power curves which appear on the scatter plot. The LDT is rated to operate at 7 MW (7000 kW). Although, the rated power is 7 MW, there is an issue of noise curtailment. According to Keller et al. (2014), "Noise from wind turbines comes from two sources: the mechanical and the aerodynamical noise.". To prevent

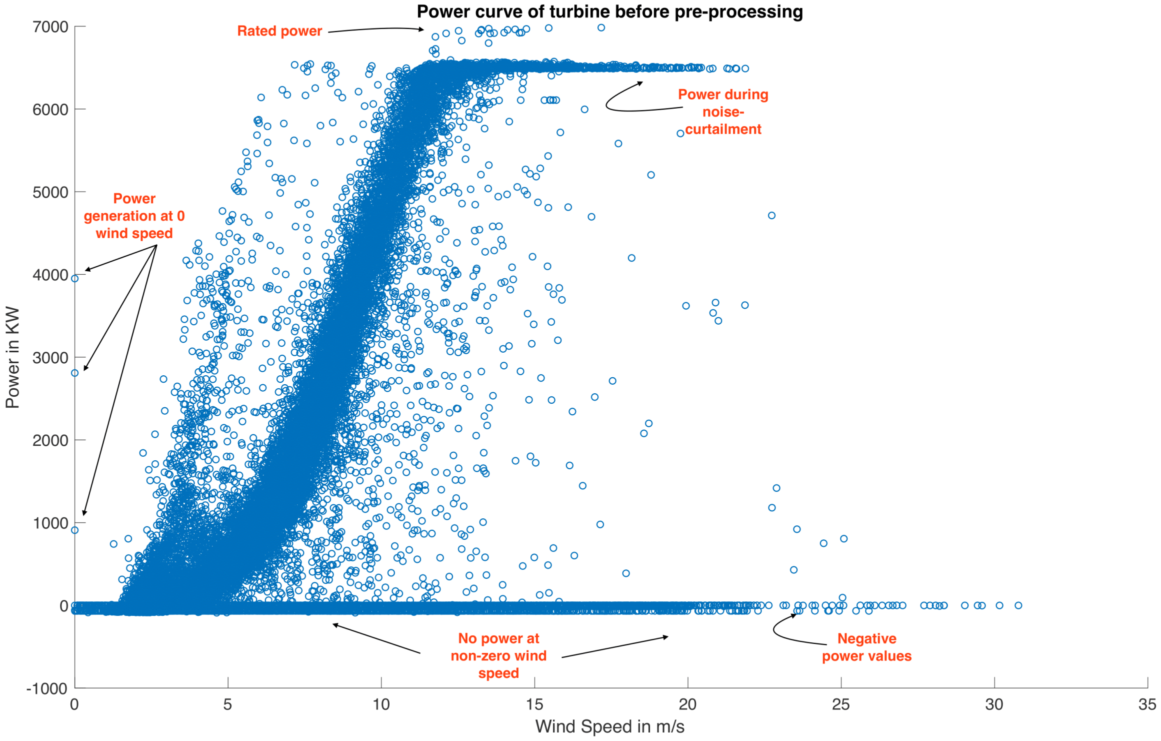


Fig. 3.2 Power curve of the turbine developed before pre-processing. There is a prevalence of noise and outliers in the original data.

the turbine from causing noise pollution and affect nearby homes and residential areas. Additionally, multiple factors besides noise issues may affect the power generation capacity of the wind turbine, including (but not limited to) prototype turbine design constraints, transmission limitations etc. Thereby, the turbine is restricted to operate at a maximum of 6.5 MW (6500 KW) by the turbine operator.

2. Furthermore, there are several points in the scatter plot of the power curve which are non-conventional:-

   - Power generated when wind speed is 0 m/s. (Points on the Y axis).
   - No power (0 power generated) at non-zero wind speed. (Points parallel to X axis).
   - Negative power values.

   The points pertaining to power generated at 0 wind speed and negative power values are incorrect measurements/wrong data due to malfunctioning of the sensors. These require pre-processing and data cleansing. Further, there are three key reasons for no power at non-zero wind speed:-

- Requested shutdown of the turbine (Eg. due to regular maintenance, service and repairs, noise curtailment etc.).
- Forced outages (due to faults in the turbine sub-components).
- Wind speed below cut-in speed.

It is the forced outages which are of utmost importance. They are the actual faults which occur in the various sub-components of the turbine like pitch system, generator, gearbox etc. To develop the final power curve and perform pre-processing, we remove the following events from the data:-

1. Requested shutdowns.
2. Noise curtailment.
3. Missing values (NaNs).
4. Technical standby (Eg. regular tests, maintenance, repairs and service etc.).
5. Power generated below cut-in wind speed.
6. Negative power values.
7. Power generated at 0 wind speed.

For the purpose of our study, an anomaly is considered to have occurred whenever there is a fault raised in the processed events data for alarms in between a specific time duration (*TimeOn*-when the alarm was started and *TimeOff*-when the alarm was cleared). Based on the alarm logs, all faults which had occurred in the turbine's operation were available, and the faults were labelled in the dataset within the duration of these alarms. All other circumstances, wherein there was no alarm raised was considered to be normal operation for this study. Further, to specifically ensure only anomalies which have occurred as a consequence of an actual fault in the turbine are targeted and not due to requested shutdowns, the forced outages in the turbine are considered based on the unique alarms list to be an indicator of an anomaly. The flowchart shown in Figure 3.3 briefly outlines the process used for data pre-processing and labelling of faults. It should be emphasised here that if the turbine is not operating at the rated power (7MW), it should not necessarily be interpreted as a fault. This is because the turbine can very well operate at the rated (7 MW) power, but due to requested shutdowns and noise curtailment problems as outlined before, it might have been restricted to operate at 6.5

Algorithm 1: Pseudo-code for data pre-processing

```
Input: Met Mast Data, Substation Data, Turbine Data, Alarm Records
Result: Merged and pre-processed data for experiments
for file in files do
                            // Sort data by time-stamp in ascending order
    $file \leftarrow sortrows(file,'StartTime')$
    $merged\_data \leftarrow innerjoin(file,'Keys','StartTime')$
                               // Merge all data at similar time-stamps
end
for samples in merged_data do
    /* Remove data points with zero wind speed but prevailing power      */
    /* If active power of turbine is non-zero and wind speed is zero     */
    if $Power\_kW\_Mean\ != 0\ \&\ Anemo\_1\_Mean == 0$ then
        /* Eliminate the matched SCADA data samples by assigning an empty
           list                                                          */
        merged_data ←[] end
        /* Eliminate data samples with negative power values             */
        /* If active power is negative (outlier/anomaly)                 */
        else if $Power\_kW\_Mean < 0$ then
            /* Eliminate the matched SCADA data samples by assigning an empty
               list                                                      */
            merged_data ←[]
        end
        /* Identify the cut-in wind speed                                */
        if $Power\_kW\_Mean > 0$ then
            $wind\_speed \leftarrow Anemo\_1\_Mean$
            $cutin\_speed \leftarrow min(wind\_speed)$
        end
    end
```

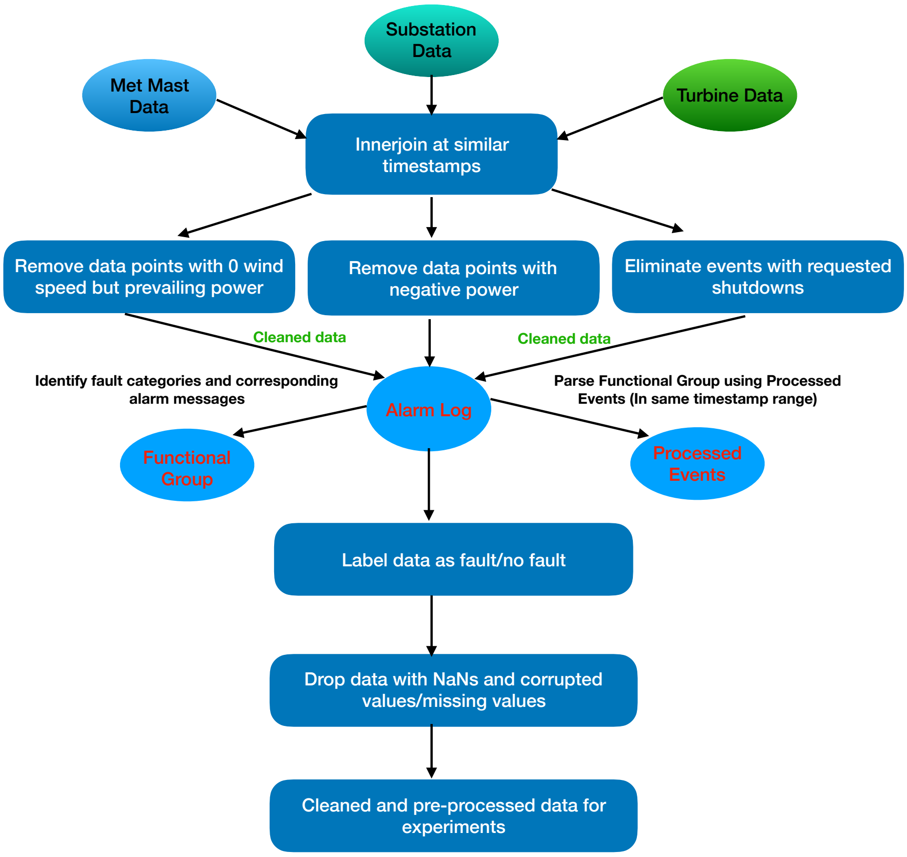

Substation Data
Met Mast Data
Turbine Data
Innerjoin at similar timestamps
Remove data points with 0 wind speed but prevailing power
Remove data points with negative power
Eliminate events with requested shutdowns
Cleaned data
Cleaned data
Identify fault categories and corresponding alarm messages
Parse Functional Group using Processed Events (In same timestamp range)
Alarm Log
Functional Group
Processed Events
Label data as fault/no fault
Drop data with NaNs and corrupted values/missing values
Cleaned and pre-processed data for experiments

Fig. 3.3 Flowchart outlining the data pre-processing and cleansing process

MW. Algorithm 1 describes the brief process and pseudo-code utilised for merging the data files and pre-processing.

After pre-processing, we had access to faults arising in various different functional groups. However, the primary constraint was that very few of these faults were actually available as labelled classes when correlated with the SCADA data. For our study, we chose to eliminate all events wherein there were no instances of labelled faults available corresponding to the SCADA data as our primary purpose was to identify faults based on SCADA data, and labelled faults are necessary for this purpose. Finally, we obtained a total of 14 functional groups for our study as elaborated in the next paragraph.

The power curve for the LDT under the normal and anomaly conditions is shown in Figure 3.4. To create a smoother curve for visualisation, the normal operations were grouped together and a binned scatter plot was developed. It was found that the majority of anomalies are labelled as *Partial performance degraded* - a situation in which the prevailing wind speed is sufficient for producing power, but the turbine is not operating at its optimal capacity due

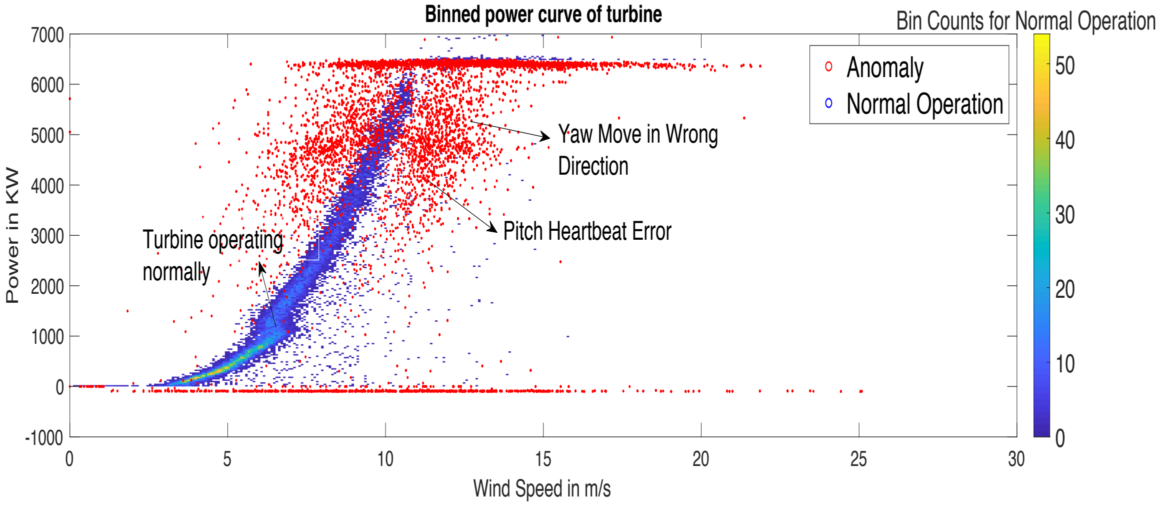


Fig. 3.4 Binned power curve for the LDT under normal and anomaly conditions based on ground truth

Table 3.1 Example of LDT's SCADA data structure – As the actual values of the features are confidential, the numbers shown here are for illustration purpose only

Time-stamp	**Feature 1** (X_1)	**Feature 2** (X_2)**......**	**Feature N** (X_N)	**Functional Group**
dd/mm/yyyy hh:mm:ss	2.104	0.890	8.124	Pitch System Interface Alarms
dd/mm/yyyy hh:mm:ss	1.245	3.753	9.509	Hydraulic System
dd/mm/yyyy hh:mm:ss	0.156	1.234	7.120	No fault

to some internal problems within its sub-components. The power curve shown under normal conditions is based on ground truth of labelled anomaly data from the alarm log, and should not be confused with an idealised power curve (Chatterjee and Dethlefs, 2019c, 2020e; Sohoni et al., 2016; Trivellato et al., 2012a), which is a reflection of the ideal operation of the turbine under ideal conditions (uniform and steady wind, zero yaw error etc.). Specifically, ground truth refers to the actual observations (normal operation/anomaly) which had historically occurred in the turbine's operation in real-world conditions, based on the LDT alarm log, and an anomaly for this study denotes a situation wherein there was a fault in one of the 13 different functional groups (pitch system, gearbox, yaw brake etc.) associated with the turbine. We consider the normal operation of the turbine as the 14th functional group for this study. See Table 3.1 for an example of the LDT's SCADA data structure.

After all pre-processing steps, a total of 21,392 SCADA samples were obtained from the LDT, each containing 102 features. These features consist of 10-minute averages for the mean, maximum, minimum and standard deviation values for vital parameters in the turbine like pitch angle, rotor speed, gearbox oil temperature etc. as well as environmental metrics like wind speed, wind direction etc. Note that from this point in the thesis, we will refer to

Table 3.2 Example format of input data structure with corresponding alarm messages – As the actual values of the features are confidential, the numbers shown here are for illustration purpose only

Time-stamp	**Feature 1 (X_0)**	**Feature 2 (X_1)......**	**Feature n (X_n)**	**Event Description (Alarm Message)**
dd/mm/yyyy hh:mm:ss	2.104	0.890	8.124	Turbine Operating Normally
dd/mm/yyyy hh:mm:ss	1.245	3.753	9.509	Pitch System Fatal Error

all 102 features as SCADA data. While few parameters like wind speed were not originally a part of the SCADA data in the original files, after merging and pre-processing all files, we consider these together alongside other SCADA features to obtain a single dataset and ensure coherence. We found that our dataset consisted of 26 different categories of alarm messages, which were historically recorded in the turbine's alarm logs. Figure 3.5 depicts the word cloud showing the prevalence of alarms in the turbine's logs – an interesting observation outlining that natural language is present in data from wind turbines. The sample data format for the alarm messages corresponding to SCADA features is outlined in Table 3.2.

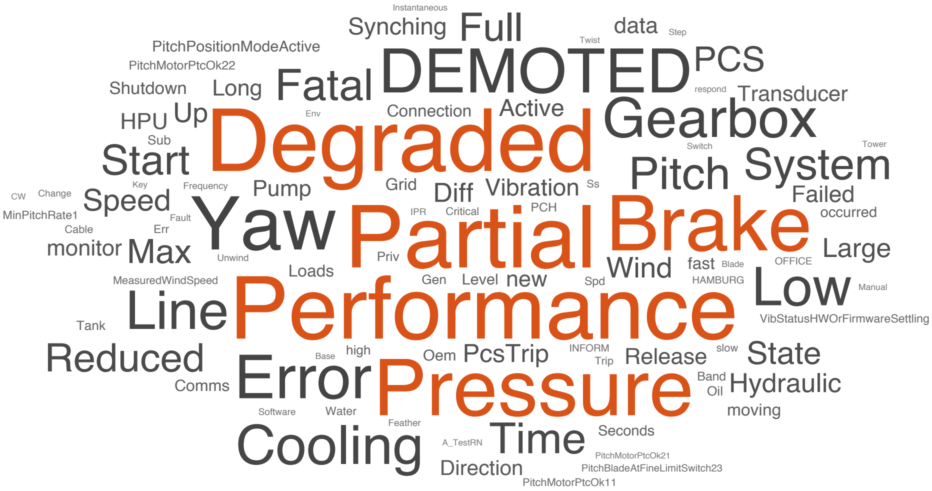


Fig. 3.5 Wordcloud outlining the prevalence of natural language in wind turbine data (alarm messages)

3.2.3 Synthetic Minority Over-Sampling Technique (SMOTE) for the LDT SCADA data

A significant challenge with our LDT SCADA dataset (and a common issue in general with SCADA data in the wind industry) is the prevalence of a major imbalance between data samples labelled for normal operation and anomaly. In our case, we had 16,948 instances of

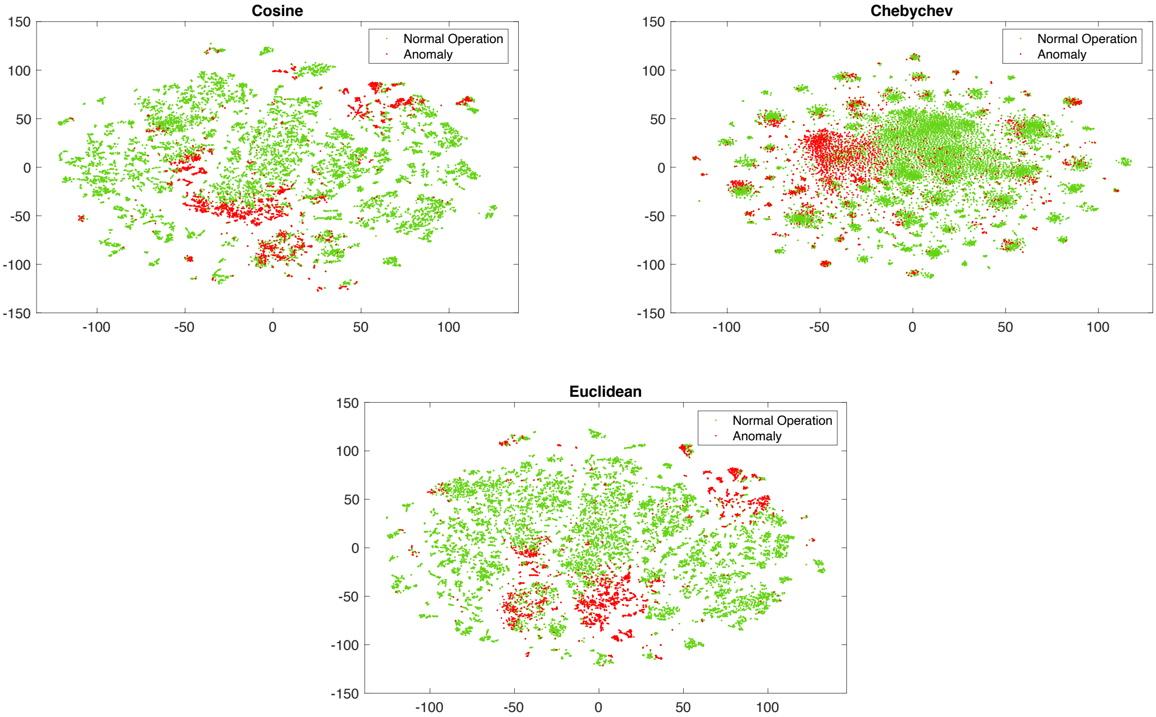


Fig. 3.6 t-SNE cluster plots of SCADA data reduced to two dimensions, with different similarity metrics. The significant imbalance between the classes for normal operation and anomaly is clearly visible, with a sparse distribution of the data points.

normal operation, while the remaining 4,444 instances represent an anomaly. This challenge of imbalance in our dataset can be better represented through the application of t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton, 2008), which is an unsupervised, non-linear statistical technique popularly utilised for visualisation and exploration of multi-dimensional data originally in high-dimensional space by mapping and reducing them into two or more dimensions suitable for human inference. We utilised t-SNE with two-dimensional data embeddings and various distance measures (Cosine, Chebychev and Euclidean) as similarity metrics to visualise the high-dimensional SCADA dataset with varying perspectives, as shown in the scatter plot in Figure 3.6. As can clearly be seen, besides the significant difference in the number of samples for normal operation of the turbine and anomalous behaviour, the t-SNE clusters are clearly not distinctly separable, with overlaps prevailing between the clusters. Also, it is interesting to note that the data samples are widely scattered in the two-dimensional space with a sparse distribution. Owing to the huge imbalance between these classes as well as in the types of anomalies which have historically occurred in the turbine's sub-components, we utilise the Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002) for balancing the samples by oversampling the minority class of data, and thereby balancing the overall class distribution for our dataset. This is integral to prevent any bias which AI models may make towards the majority (normal operation) samples during training in the later chapters.

SMOTE is a popular type of statistical algorithm commonly used during data analytics and pre-processing, and the key principle behind its working is the continuous lookup of successive samples in the original (unbalanced) training dataset and estimation of the distance between the coordinate points occupied by the K-Nearest Neighbours across the entire feature space (Chawla et al., 2002). Synthetic data points for the minority class are generated in SMOTE through multiplication of the estimated vectorial distance between the coordinate points of nearest neighbours in the imbalanced dataset with a random-valued real number in the range of 0 to 1, and then, finally, performing a summation with the sample's present value in the SCADA feature space. More details on SMOTE and its working can be found in Chawla et al. (2002).

We utilise the *imbalanced-learn* (Lemaître et al., 2017) library in Python towards implementation of SMOTE on an 80% portion of the original LDT SCADA data. This is the same portion of data which will later be utilised in training AI models for decision making, while the remaining 20% of the original (untouched) data is left for testing. A total of 17,112 data samples are utilised for the SMOTE, of which 13,594 represent samples of normal operation, while the remaining 3,518 are samples of anomalies. The synthetic data finally generated by the algorithm consists of 27,188 samples, of which 13,594 instances fall into normal operation and the remaining 13,594 belong to the anomaly class i.e. a perfect balance is obtained between the samples of the two classes.

To evaluate the resampling process performed (and its validity), we applied the *Kolmogorov-Smirnov test (KS test)* (Hassani and Silva, 2015), which is a statistical test and non-parametric technique for comparing the equality between different probability distributions i.e. in our case the original and synthetic data distributions. The results for SMOTE are illustrated in the heatmap shown in Figure 3.7, wherein a p-value of 0 would denote completely different (distinct) probability distributions, and 1 would represent that the distributions are perfectly equal (same). As can be visualised, the majority of features in the original and synthetic data have a similarity of at least 75% (with any of the 102 SCADA features having at least 50% similarity), and despite sudden changes in the form of valleys for some features, we can safely account for the fact that the synthetic data obtained is a good probabilistic approximation of the initial LDT SCADA data before oversampling. Note that the average p-value obtained was 0.898 between the original and synthetic data distributions.

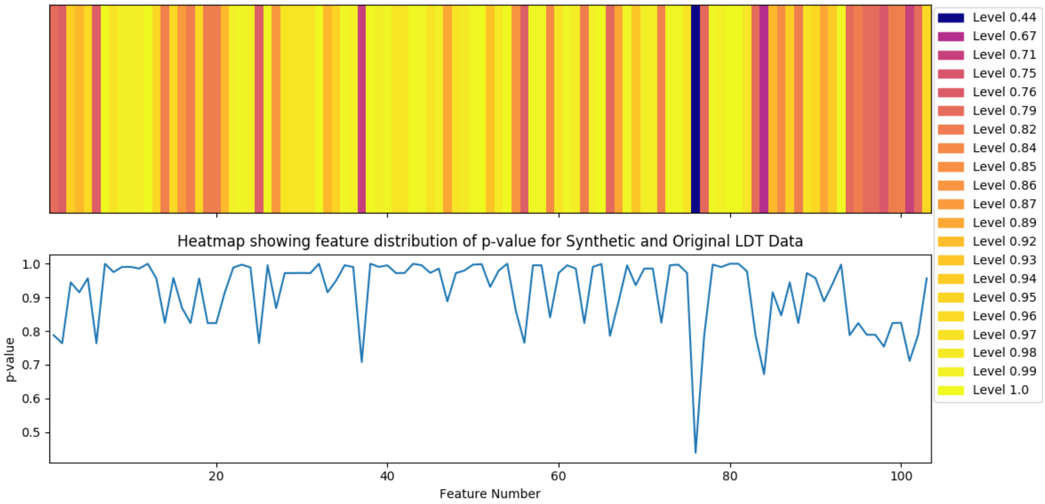


Fig. 3.7 Visualisation of the p-values between the original and synthetic data distributions for the LDT. Clearly, the majority of the SCADA features in the original and synthetic data distributions have at least 75% similarity (with at least 50% similarity for any feature), representing that the synthetic SCADA features are statistically similar to the original features before oversampling.

Oversampling for alarm messages

A significant challenge with the pre-processed LDT dataset containing historical alarm records was the presence of major class imbalance between alarm types, as some alarm messages are generally more frequent than others. This can lead to an imbalanced training dataset for generating alarm messages using AI models. For instance, there were 5,050 recorded instances of the *"Pitch System Fatal Error"* alarm which can be attributed to the frequent pitch angle disorientation in turbines that is fairly common. However, there were much rarer messages such as *"HPU 2 Pump Active For Too Long"* accounting for 2,525 cases. Even rarer cases prevalent in our dataset include *"PcsFaulted"* with only 101 instances. As already outlined, training over an imbalanced dataset would lead the AI model to learn a biased prediction policy that prefers majority samples of alarm messages in most instances pertaining to frequent alarms, while failing to learn a valid representation for the alarms which are less prevalent and rarer.

To resolve this problem, one possible solution is to generate more instances of labelled data through simulation techniques and based on heuristics. However, this would require engineers for annotating additional alarm messages for multiple conditions and was deemed too costly and challenging, given the complex nature of SCADA data and presence of multiple features. There is an alternative technique which can reproduce additional data points for alarms, such as by introducing random noise or performing uniform random sampling on SCADA features. While techniques like this are simple to apply and effective, they contain a major risk of changing the overall distribution of the SCADA data during the process.

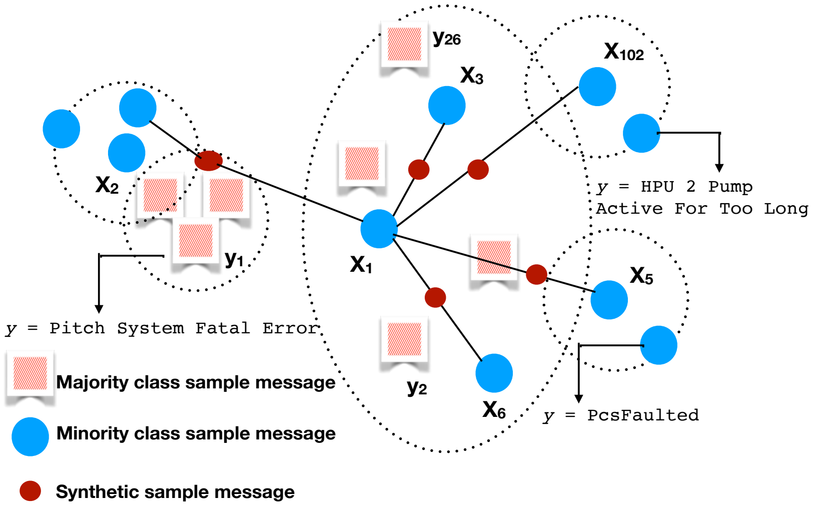

y26
X3
X102
X2
y1
X1
X5
y2
X6
y = HPU 2 Pump Active For Too Long
y = Pitch System Fatal Error
y = PcsFaulted
Majority class sample message
Minority class sample message
Synthetic sample message

Fig. 3.8 Depiction of synthetic minority oversampling process for alarm messages corresponding to the LDT SCADA data

We again decided to utilise SMOTE to tackle this challenge, and this approach is depicted in Figure 3.8. Given the 102 SCADA features in the LDT dataset represented as $\mathbf{X}_t = (X_1,\ldots,X_{102})$, SMOTE creates a feature space mapping from the numeric input features to specific classes of alarm messages represented as $\mathbf{y} = (y_1,\ldots,y_{26})$. This is performed by learning to create synthetic samples from majority samples of alarm messages to minority samples in accordance with the vectorial distance in between the data points.

3.3 ENGIE La Haute Borne Onshore Wind Farm Data

The SCADA data from the ENGIE La Haute Borne open wind farm pertains to real-world operational data from an onshore wind farm located in Meuse, France. This dataset consists of four wind turbines, wherein, each turbine is rated to operate at 2 MW, thus giving the wind farm ability to produce a total power of up to $2MWx4$ = 8 MW. The four individual turbines in our dataset are represented by unique legends as R80711, R80790, R80721 and R80736 respectively. We utilise exactly the same types of features which are common to this dataset with the LDT SCADA data, consisting of total of 102 features. There were a total of 840,380 samples in the dataset, of which 210,095 samples are specific to each of the four turbines in the wind farm. Figure 3.9 depicts the power curves (actual based on operational data, not ideal) for all turbines in the ENGIE dataset.

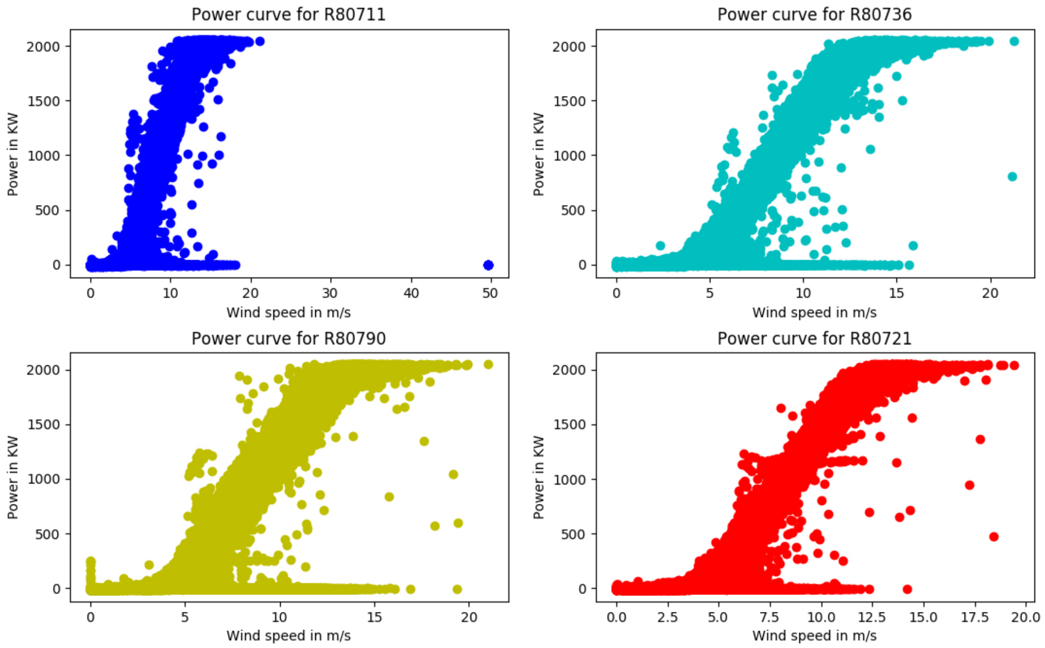


Fig. 3.9 Power curve visualisation for all turbines in the ENGIE La Haute Borne wind farm as per operational SCADA data

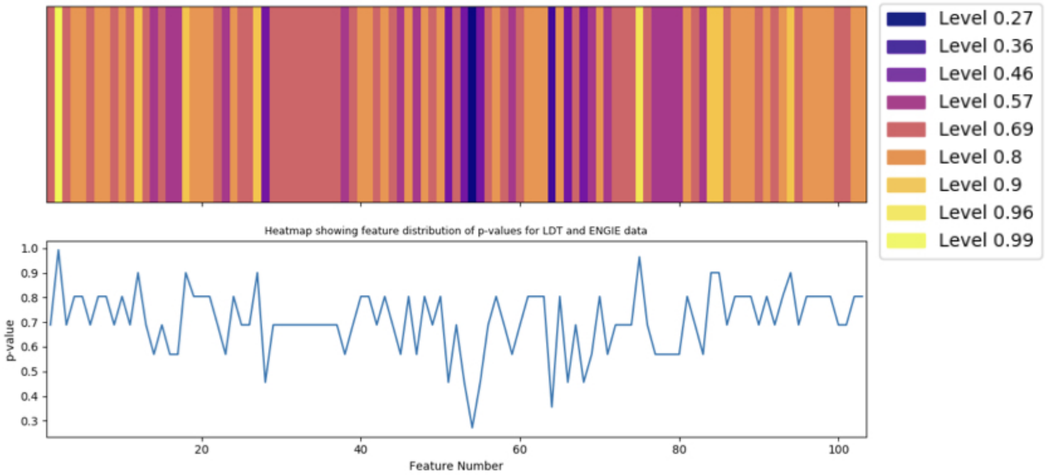


Fig. 3.10 Visualisation of the p-values between the LDT and ENGIE SCADA datasets. Clearly, there is a notable difference between the statistical distributions of SCADA features for these datasets.

Note that as the LDT and ENGIE SCADA datasets pertain to different types of turbines with different specifications, there is a significant difference between the feature distributions for these datasets. This phenomenon (similar to the previous depiction in Figure 3.7) is visualised in the heatmap shown in Figure 3.10, which clearly outlines that the majority of the 102 SCADA features vary substantially in terms of their distributions for the LDT and ENGIE datasets. For the ENGIE data's distribution in terms of its similarity to the LDT data, we obtained an average p-value of 0.705 through the KS test. This clearly serves as a notable evidence on the differences between our datasets, which is later used in Chapter 4 for the purposes of demonstrating transfer learning.

Fault labelling through the *K*-Means++ clustering algorithm Owing to the lack of explicit labels on the turbine's operational status in the ENGIE dataset, we utilise the *K*-Means++ clustering algorithm (Wu, 2012) for grouping the data into two clusters i.e. normal operation and anomaly. This algorithm assigns individual points in the dataset based on the prevailing SCADA features into one of *K* possible classes which is dependent on the distance of the points from the cluster centroid, refer Krishnan and Krishnan (2018) for more details. For our fault labelling task, we utilise a value of *K=2* and train the algorithm over a maximum of 500 iterations, with the Euclidean distance metric towards estimating the clusters.

Due to the lack of ground truth for historical faults in the ENGIE data, while it was not possible to verify the clustered labels against the true labels, we performed an evaluation based on the LDT data to offer a comparative estimate of the clustering algorithm's performance. For this purpose, we utilised the same algorithm towards validation on the LDT data for which the historical ground truth labels for faults were available. On 100 iterations of the algorithm with a varying number of centroid seeds, we obtained an accuracy of 87.8% for predictions of normal operation vs anomaly clusters, which validates the suitability of the technique for the ENGIE dataset.

Finally, we obtained 840,380 samples in the ENGIE data, containing 633,365 instances of normal operation and 207,015 remaining instances of anomaly. The clusters for these two classes are depicted in Figure 3.11. As can be seen, while there are significant overlaps between the two classes for some instances, they are still clearly separable. This is a characteristic pattern which is not uncommon to real-life datasets and their distributions (Whang et al., 2015).

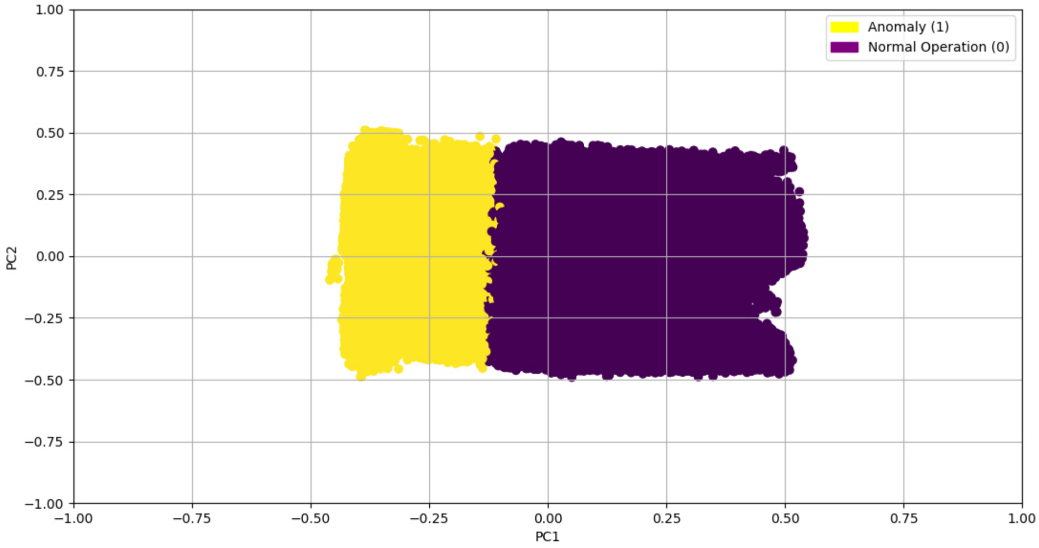


Fig. 3.11 Scatter plot outlining the clustered classes in ENGIE data based on top-two principal components. Despite some instances of overlaps between the clusters for normal operation and anomaly, the distributions are distinctly separable.

3.4 Discussion

The complete details of the 102 SCADA features utilised in this thesis from the LDT and ENGIE datasets is provided in Appendix A. In the forthcoming chapters, we utilise the LDT SCADA data as the primary source of knowledge for training Explainable AI (XAI) models in Chapters 4, 5, 6 and integrating the XAI model with a domain-specific multimodal knowledge graph in Chapter 7. The ENGIE SCADA data is utilised in Chapter 4 to demonstrate the application of transfer learning in DL models in case of lack of labelled training data. Note that while we only utilise the ENGIE data in Chapter 4, we believe that it can be adapted to other applications proposed in the forthcoming chapters with appropriate modifications to the data structure depending on the type of AI models utilised. Besides the comprehensive and detailed explanations of these datasets in this chapter, we briefly mention the type of datasets used and their size/dimensions in-text within the relevant sections for data description in the forthcoming chapters. With potentially useful, coherent and informative pre-processed data at hand now, we will utilise the datasets for experiments in training DL models for explainable anomaly prediction and offshore vessel transfer planning in the next chapter (Chapter 4).

Chapter 4

Deep Learning for Explainable Anomaly Prediction and Offshore Vessel Transfer Planning

> The only thing worse than starting something and failing is not starting something.
>
> Seth Godin

This chapter builds upon the work which has previously been published by the author as an article in the journal of *Wind Energy* (Chatterjee and Dethlefs, 2020e), a book chapter in *Developments in Renewable Energies Offshore* (Chatterjee and Dethlefs, 2020b) and an article at the *WindEurope Offshore* conference (Chatterjee and Dethlefs, 2019b).

4.1 Introduction

In the pursuit towards data-driven decision making, ML techniques have seen a growing uptake in the wind industry over the last decade, as evident from the literature review in Chapter 2. DL techniques, in particular, have recently started to dominate the family of AI algorithms applied in CBM for O&M of wind turbines. However, such decision making models suffer from an important shortcoming – *lack of transparency*. While such algorithms can predict faults in wind turbines with high accuracy, they fail to provide rationales behind their decisions. This is primarily due to the black-box nature of DL algorithms, which generally leads to a lack of trust in utilising these models for practical applications. Additionally, utilising a model with high accuracy is of little use, until a more transparent diagnosis can be performed into the possible causes which contribute to an anomaly in specific turbine sub-components (e.g. gearbox, pitch system etc.).

In this chapter, we propose the application of a hybrid DL approach, combining a special type of Recurrent Neural Network called Long Short-Term Memory (LSTM) with a gradient boosted decision tree classifier (XGBoost). The LSTM helps in providing accurate predictions of impending faults in the turbine, while the XGBoost provides transparency and outlines the possible features in the input SCADA data giving rise to the anomaly. We also demonstrate some other wider applications of the proposed technique beyond accurate and transparent prediction of faults, including its portability across different domains through transfer learning, and the ability to perform maintenance action planning for offshore vessel transfers through Deep Reinforcement Learning techniques.

The chapter is organised as follows: Section 4.2 provides an introduction to Artificial Neural Networks and describes their limitations, and discusses the motivation in utilising Recurrent Neural Networks for our anomaly prediction task. The proposed DL model for explainable anomaly prediction is discussed in Section 4.3. Section 4.4 briefly describes the datasets utilised in this chapter. The experiments and results are discussed in Section 4.5. The applications of Deep Reinforcement Learning for vessel transfer planning are proposed and discussed in Section 4.6. Finally, Section 4.7 concludes the chapter and describes the scope of the forthcoming chapters.

4.2 Artificial Neural Networks (ANNs)

Human brains are complex biological computing machines which learn from sensory information and adapt to changing environments in order to make decisions, often from disparate sources of information (Cox and Dean, 2014). In a similar manner, neural networks are inspired by the functioning of the human brain, and can automatically modify their internal structure to achieve a specific function objective (Grossi and Buscema, 2008). The basic concept behind a neural network is that it takes as input a collection of features (also called predictors), and uses them to automatically learn the best set of rules to reach a specific goal, without being explicitly programmed. The key features in the case of wind turbines would include historical SCADA data measured from the various sensors in the turbine (e.g. wind speed, pitch angle, gearbox oil temperature etc.). The goal is generally either a classification task, wherein, given a specific set of features, the category which these belong to is to be predicted (e.g. the current operational status of a wind turbine), or a regression task, wherein, continuous values are predicted for a target variable (e.g. future forecasts for turbine power output).

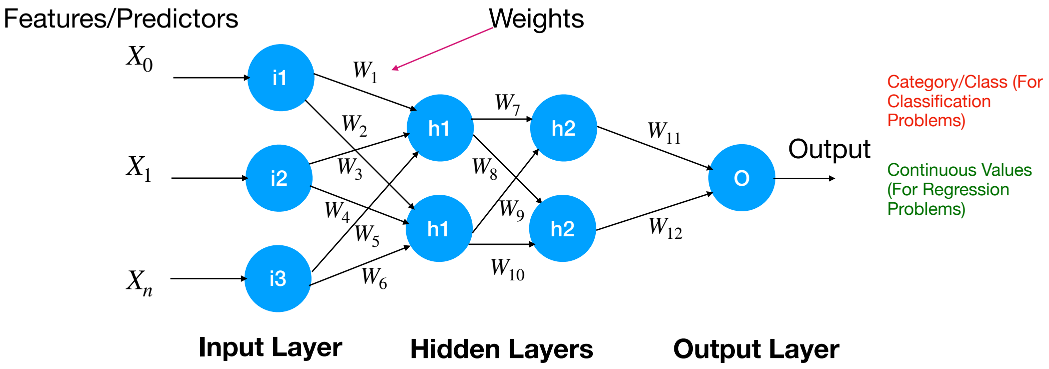


Fig. 4.1 The basic structure of an Artificial Neural Network (ANN). Every network consists of an input layer, one or more hidden hidden layers and an output layer. Unique weights w_k in the model interlink transitions from the nodes in one layer to another.

4.2.1 Feedforward neural networks

The simplest form of Artificial Neural Networks (ANNs) are referred to as the feedforward neural networks (Bebis and Georgiopoulos, 1994; LeCun et al., 2015; Thomas and Wilscy, 2011), wherein, the name itself signifies the very idea – the input parameters (features) in such a network are continuously transmitted (fed forward) to the further layers until the model arrives at the output, followed by backwards pass of these values to the previous layers in order to optimise the objective of minimising the loss (error) which the network makes during its predictions. Every neural network is structurally composed of three key components:-

- **Input Layer:** This is the very first layer in every network, which takes in multiple input features (predictors) to accomplish the learning objective in mapping them to a specific target. Note that the features here can be numeric values as well as categorical values. Regardless of the type of data being utilised for the learning task, the essence of the input layer is that the input features always need to be either originally numeric, or transformed into a numeric format to facilitate mathematical computations. For example, an input here can even be images and videos, which can be numerically represented via the intensity of pixels, or text (words and sentences), for which word embeddings can be calculated to encode them as numeric vectors. Similarly, categorical inputs can be one-hot encoded to convert them to numeric.

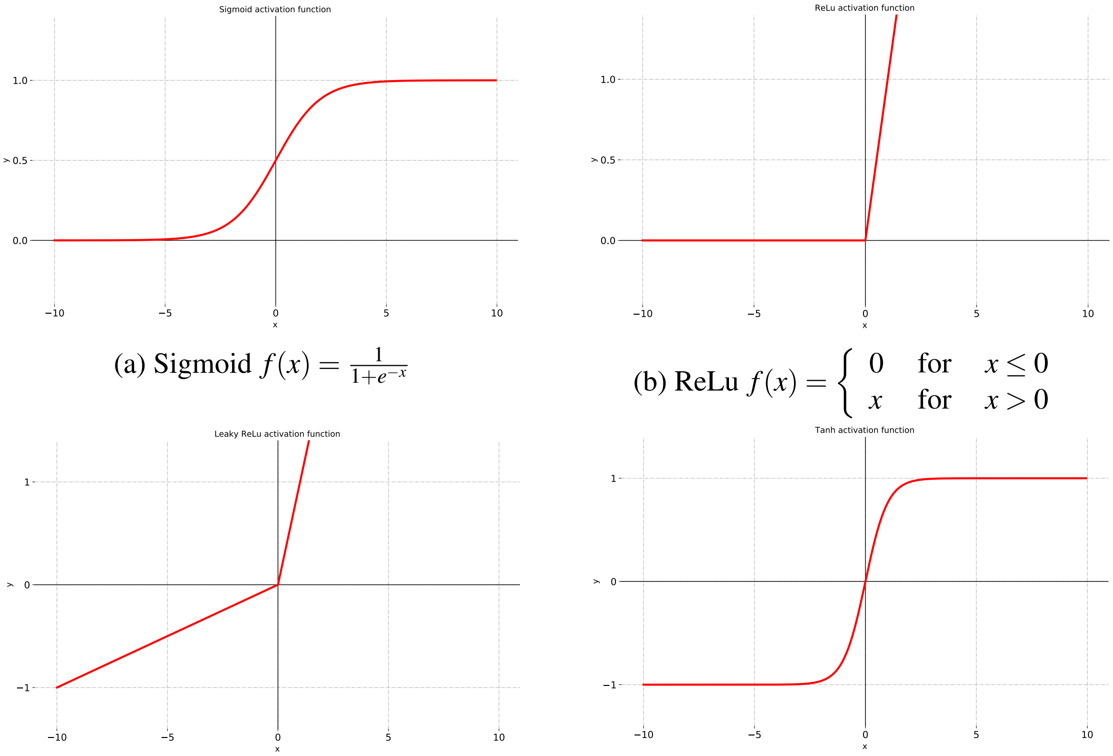
- **Hidden Layer(s):** These are intermediate layers of a neural network, wherein, all mathematical computations take place in order to achieve a specific objective in the most efficient manner. The hidden layers basically model the parameters received from the input layer and apply mathematical functions on them, in order to identify complex relationships and patterns between the features. The mathematical functions which the network automatically learns help in generating different forms of representation of the features, which can finally be utilised by the output layer to generate the final predictions. Any neural network can have either a single or multiple hidden layers, and the very presence of more than one layer is vital for the purposes of DL. While earlier neural networks were simpler and only consisted of an input and an output without any hidden layers (called Single-Layer Perceptron) (Marcialis and Roli, 2005; Šarūnas Raudys, 1998), such models are rarely used at present, due to their limited capabilities in modelling non-linearly separable problems.

- **Output Layer:** The final layer of every network is the output layer, wherein, the model outputs its final predictions. The final predictions can be a particular class of the output in case of classification problems (e.g. 0 or 1 for binary classification, such as predicting whether a turbine is operating normally or has an anomaly), or continuous values for regression problems (e.g. predicted power output for the wind turbine).

Figure 4.1 represents the basic structure of an ANN. The complex mathematical computations which take place within the hidden layers of the network have a very specific objective – the network needs to optimise its internal parameters (weights) in order to achieve the *global minima* (minimum loss or error in making the predictions). The unique weights, denoted as w_k represent the strength (importance) which the network assigns to each feature during computations, and form part of the hidden layers. It is integral to note that the weights of the network are updated during a special process called *training*, wherein, the network is fed with a subset of the initial dataset to facilitate learning and discovery of relevant patterns in the data. The loss is generally estimated in terms of an error metric such as mean squared error, mean absolute error etc., in terms of the differences between the predicted values and the ground truth (expected values) from the network. Equation 4.1 describes a typically used loss function for the network, wherein, n represents the total number of training samples in the dataset. Ideally, we would prefer a zero loss wherein, the true and predicted values are completely equal, and the goal here is to bring the overall loss as close to zero as possible during training.

$$\text{Loss}(y_{true}, y_{predicted}) = \sum_{i=1}^{n} (y_{true} - y_{predicted})^2 \tag{4.1}$$

The network continuously updates its weights in order to achieve the global minima, and during this process, an activation function is applied to restrict the values at different nodes in the network within a specific range. This is vital as initially, the numeric features for instance, can have different scales and ranges, which would make the learning process inherently complex. Some commonly utilised activation functions for the ANN include Sigmoid, Rectified Linear Unit (ReLu), Leaky ReLu, Hyperbolic Tangent etc, with non-linear functions generally preferred for modelling complex, real-world datasets (Dureja and Pahwa, 2019). Figure 4.2 depicts some of these activation functions.



(a) Sigmoid $f(x) = \frac{1}{1+e^{-x}}$

(b) ReLu $f(x) = \begin{cases} 0 & \text{for} \quad x \leq 0 \\ x & \text{for} \quad x > 0 \end{cases}$

(c) Leaky ReLu $f(x) = \begin{cases} ax & \text{for} \quad x \leq 0 \\ x & \text{for} \quad x > 0 \end{cases}$

(d) tanh $f(x) = \frac{2}{1+e^{-2x}} - 1$

Fig. 4.2 Some commonly utilised activation functions in DL. The functions map the original values in the datasets to a restricted range of values, to assist the learning process.

Consider the input features fed to the input layer of the ANN as X.These features are fed-forward to the consecutive hidden layers, and the network optimises its weights through a process called *back-propagation*, wherein, an optimisation algorithm such as gradient descent (Ruder, 2016) applies differential calculus to minimise the objective function for the network's loss. Equation 4.2 demonstrates the gist of back-propagation.

$$\frac{\partial \operatorname{Loss}(y_{true}, y_{predicted})}{\partial W} = \frac{\partial \operatorname{Loss}(y_{true}, y_{predicted})}{\partial y_{predicted}} * \frac{\partial y_{predicted}}{\partial z} * \frac{\partial z}{\partial W} \tag{4.2}$$

Here, z describes the output of the network at each hidden node, represented as a summation of the dot product of the prevailing weights w_k in the hidden neurons and input features X with an added constant term b called bias, as per equation 4.3.

$$z = b + \sum_{i=1}^{n} X_i w_k \tag{4.3}$$

The key goal for the ANN is to map the inputs X within the data to corresponding outputs y through estimation of abstract hidden states h. In this situation, the hidden representations of the inputs are computed according to equation 4.4, where f represents a specific type of non-linear activation (e.g. Sigmoid, Hyperbolic Tangent, ReLu etc.).

$$h = f(X) \tag{4.4}$$

The final performance of the network is always evaluated on the *test set*, which is a separate sub-portion of the original dataset held out before training the network. The accuracy of the network on the unseen test set provides the basis to evaluate how well a network generalises to the task at hand, which can either be a classification problem or regression problem.

Drawbacks of feedforward ANNs Note that although the conventional feedforward ANN can effectively model complex-non linear data, they lack the notion of time (Lipton, 2015) and can only focus on formative aspects of the data during training. In many real-life applications for computational decision support, the data often has a temporal nature (e.g. SCADA data from turbines is measured periodically at generally 10-minute intervals). The ANN thereby suffers from the problem of neglecting past information in time, affecting its performance on such temporal tasks. This motivates us towards the realisation of temporally-aware networks, as discussed next.

4.2.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) (Hopfield, 1982), are an extremely powerful type of ANN, which can recognise patterns and effectively model complex data possessing sequential

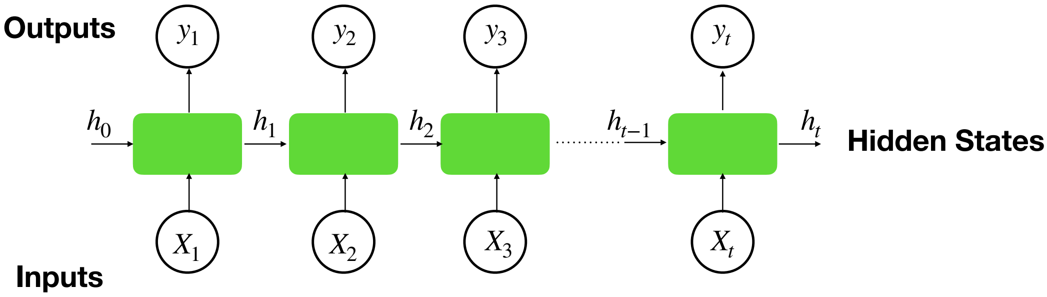


Fig. 4.3 Structure of a Recurrent Neural Network (RNN) – feedback connections from present outputs to future inputs give the network the notion of memory

nature and temporal dimensions. RNNs are not limited to only numeric data with temporal aspects (as even text, genome data, audio signals etc. have a sequential nature), and have been game-changers in the areas of machine translation, time-series forecasting, text classification, image recognition, anomaly detection, text generation (De Mulder et al., 2015; Sherstinsky, 2018) etc. Unlike conventional feedforward networks which only consider the present inputs, the RNNs can consider both, past information as well as present inputs. This crucial temporal characteristic of RNNs is owed to a special type of *memory* which they possess, and can arguably be considered as an artificial replica of the human brain (Ghosh, 2019).

Figure 4.3 outlines the basic structure of a RNN. The basic notion here is that a decision which the RNN makes at any past time-step directly affects the new decisions which the network would make at the later instances of time. Thereby, the RNN's decisions at time-step $t-1$ contribute to shaping the network's behaviour at time-step t. This idea forms the very basis of a RNN, and the learning process helps the network to derive insights and predictions for new, unseen data, which is very similar to the way humans learn to make decisions in everyday life (Dezfouli et al., 2019). A RNN consists of feedback connections from the past outputs to the present inputs, and as new predictions continue to be made, the present outputs become past inputs and this process repeatedly continues, giving the RNN the notion of *memory*.

The information from the past prevalent in the sequential data is stored within the RNN's hidden states, which can help the network *remember* information from multiple time-steps in the past. This ability of the RNN is owed to the processing of long-term dependencies within the sequence, constituting its new predictions as a function of past information (from the hidden states) and the present inputs. Consider equation 4.5, describing the estimation of hidden states within the RNN. Here, h_t describes the present hidden states, and h_{t-1}

represents the past hidden states at the previous time-steps. x_t denotes the input at the present time-step.

$$h_t = f_W\left(h_{t-1}, x_t\right) \tag{4.5}$$

The Hyperbolic Tangent activation function (tanh) is mostly utilised in the RNN architecture to restrict the values between -1 and 1. While other activations popular for conventional ANNs e.g. ReLu as mentioned in Section 4.2.1 can also be utilised, such activation functions would generally cause the computed values in the RNN to explode during the constant feedback process, as they are not restricted between a specific range. The *tanh* activation makes the feedback process in the network more computationally effective in comparison to other activation functions (Le et al., 2016), and the function's second derivative (obtained during gradient descent) can also sustain for longer without reaching zero values. This gives the network an ability to increase (strengthen connection or increase importance given to past information) or decrease (weaken connection or decrease importance given to past information) its states, as both negative as well as positive values can be preserved through this process. The goal in the learning process (back-propagation and weight updates) of the RNN is very similar to conventional feedforward networks, with the loss computed between the predicted ($\hat{y}$) and expected outputs (y) to be minimised. A commonly used metric for estimating this loss in RNNs e.g. is cross-entropy, as shown in equation 4.6,

$$(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N}\sum_i y_i \log \hat{y}_i. \tag{4.6}$$

The network's hidden states after application of the activation function are denoted through equation 4.7. w_{hh} and w_{xh} represent the weights computed at the hidden layers of the network during the feedback process, for the past and present information respectively.

$$h_t = \tanh\left(W_{hh}h_{t-1} + W_{xh}x_t\right) \tag{4.7}$$

The RNN model continuously uses past and present temporal information for its prediction-making process, during which, the hidden states h are computed in a recursive manner in accordance with equation 4.8. In this equation, t represents the present input (at the current time-step).

$$h_t = f\left(x_t, h_{t-1}\right) \tag{4.8}$$

Finally, the network outputs new predictions y_t in accordance with equation 4.9.

$$y_t = W_{hy}h_t \tag{4.9}$$

Drawbacks of Recurrent Neural Networks While the RNNs pave way towards a specialised technique in understanding the context of temporal variations in sequential data which conventional feedforward networks were incapable of, they suffer from a major problem: *vanishing and exploding gradients* (Bengio et al., 1993b; Hochreiter and Schmidhuber, 1997a; Tan and Lim, 2019). In Figure 4.4, the Sigmoid activation function is depicted, and in this case, whenever the values of the function are either too high or too low, this problem occurs and gradients either explode or vanish (on either extremes of the function's values), making the learning process infeasible.

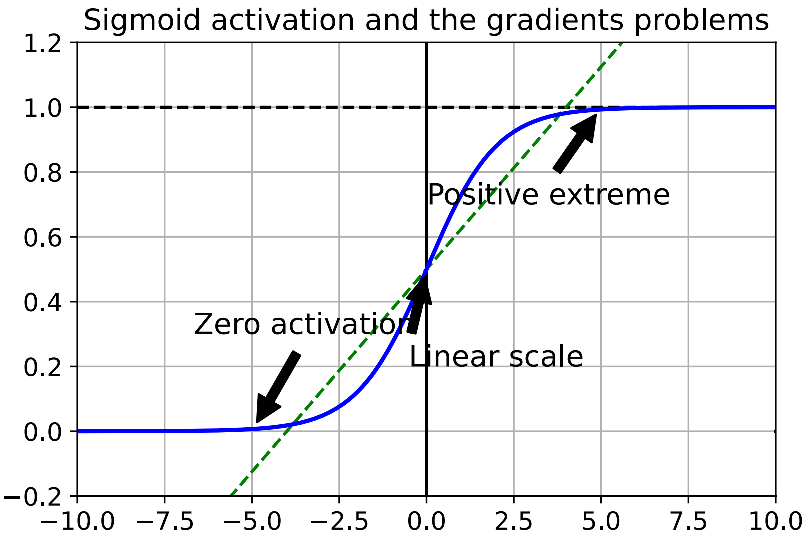


Fig. 4.4 The vanishing and exploding gradient problem – too high or too low values of the activation function affect the ability of the RNN to update weights effectively

Vanishing Gradients: As the feedback from past information is fed continuously to the RNN's present inputs, the number of matrix multiplications and gradient calculations in the learning process also continue to grow. In long sequences of data, the repeated application of the activation functions during the estimation of hidden states for the past and present information causes the values of these states to shrink in an exponential manner. These values continue to shrink and at some point, they vanish (approach 0), making it impossible for the model to learn by updating its weights.

Exploding Gradients: In another aspect, wherein the network is learning from long temporal sequences, the error gradients often continue to accumulate. This makes the network perform significantly large updates of weights, and can make the RNN unstable. In such a situation, it may be impossible for the network to approach the global minima, as it would only be stuck in local minimas during the backpropagation process. Thereby, the RNN fails to work effectively in many real-life situations wherein the data utilised spans long sequences of inputs (e.g. wind turbine SCADA data often spans hundreds of features across multiple time-steps). This motivates us towards applying a more refined and improved subset of the RNNs as discussed next.

4.3 Proposed Learning Models for Anomaly Prediction with SCADA data

4.3.1 Long Short-Term Memory (LSTM) model

Long Short-Term Memory Networks (LSTMs) were first proposed by Hochreiter and Schmidhuber (1997b) and have shown immense success over the years in performing decision making for multiple real-world sequence classification tasks, especially in the domains of machine translation and speech recognition (Gers, 1999; Graves et al., 2013; Graves and Schmidhuber, 2005; Mikolov et al., 2010; Sundermeyer et al., 2013; Sutskever et al., 2011). Just as humans utilise a sequence of different words to interpret the meaning of a complete sentence, LSTMs can generate predictive decisions, given sequential data consisting of temporally ordered observations, such as the sequence of continuous measurements from a wind turbine's SCADA logs measured over consecutive intervals of time. Unlike conventional (vanilla) RNNs, LSTMs can effectively make predictions from long sequences of input data, owing to their primary capability in learning long-term temporal dependencies from sequential data and effectively tackling the challenge of *vanishing and exploding gradients* faced by vanilla RNNs (Bengio et al., 1993a; Hochreiter, 1998) which was outlined in Section 4.2.2. The proposed learning model for explainable anomaly prediction is introduced in this section.

Consider a continuous sequence of SCADA features X which encompasses the input to the learning model, and consists of multiple vectors characterising turbine operational parameters (e.g. pitch angle, wind speed, gearbox oil temperature etc.). We define an output here denoted by y, which would outline the present operational status of a wind turbine (or multiple wind turbines in case of a wind farm), representing either *normal operation* or an *anomaly* in a turbine sub-component.

Figure 4.5 outlines the basic internal structure of a LSTM network. Note that the hidden states h_t are computed based on the inputs X_t, past hidden states h_{t-1} and the previous

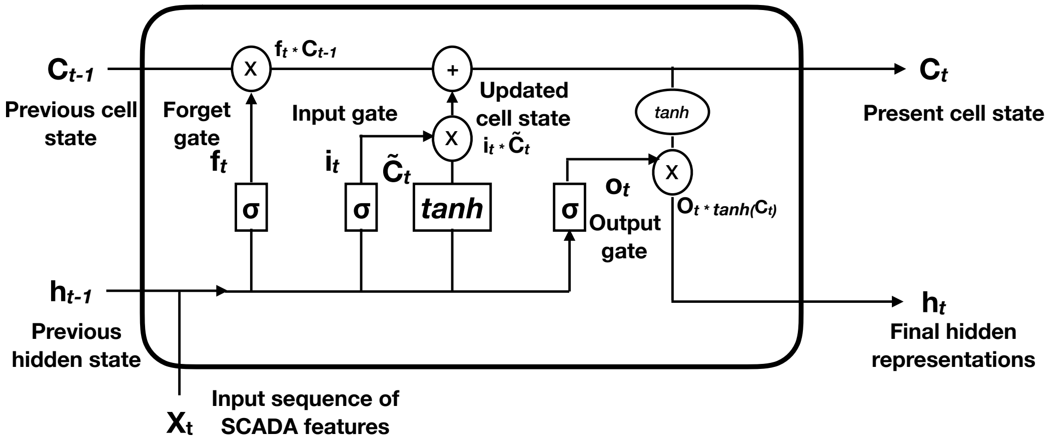


Fig. 4.5 Depiction of the LSTM model architecture, facilitating learning from SCADA sequential features. The various gates and cell states in the network help to capture relevant long-term temporal dependencies in the input data.

cell states C_{t-1}, which are dynamically controlled through multiple *gates* in the model. The present states of the cell C_t is a vital parameter, as it effectively captures information pertaining to the sequential data passed across multiple time-steps during the recursive updates of weights, and focuses on both the present input X, as well as the updates to the gates (f_t, i_t or o_t). Below, we provide a brief description of each of the gates integral to the LSTM's learning process. More detailed and thorough characteristics of these gates can be found in existing literature e.g. Graves (2013).

1. In the network architecture, the forget gate represented as f_t is modelled in the form of a Sigmoid function to decide the amount of information in the present time-step of the input data sequence to be retained or forgotten. The forget gate provides an output as a real valued number in the range of 0 (denoting forgetting of all information in the data) and 1 (denoting retention of all information in the data). This is enunciated below in equation (4.10).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4.10}$$

2. Within the LSTM model, the input gate denoted by i_t (modelled as a Sigmoid layer) provides the basis on which a decision is made regarding new information to be added to the cell states at the forthcoming time-steps. An activation function is utilised at this stage (generally *tanh*) to dynamically distribute the present cell state's values

in the range of -1 to 1, which is further used to infer updated cell states $\tilde{C}_t$. This phenomenon is described mathematically in equations (4.11) and (4.12).

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{4.11}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{4.12}$$

3. The output of the cell state denoted by o_t is determined at the final stage, wherein, an activation function (generally Sigmoid – σ) restricts the values of the output to the essential features (contributing to the output) of the cell state, and redundant components of the feature space (non-relevant features) are ignored. At this stage, the *tanh* squashing function is again utilised to limit the output within the range of -1 and 1. Additionally, the hidden representations of the input h_t at time t are evaluated based on o_t and C_t. This is outlined below in equations (4.13) and (4.14) respectively.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{4.13}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{4.14}$$

Given the generally complex and non-linear structure of SCADA data (consisting of multivariate time-series), we expect the LSTM to capture important (and relevant) long-term temporal dependencies in the data and automatically determine the key features which can optimally contribute to the prediction making process (which in our problem would represent the wind turbine's operational status). This can help eliminate requirements for manual feature engineering to identify redundant features, and thus contribute to accurate and robust decision making for O&M.

4.3.2 XGBoost

The XGBoost model is a highly scalable and efficient implementation of gradient boosted decision trees, and was first proposed by Chen and Guestrin (2016b). XGBoost is basically a supervised learning model, which can build an ensemble of multiple decision trees for classification, and thereby produce promising results, sometimes even with sparse sources of data samples (Leventis and Leventis, 2018). An example of the structure of a decision tree for

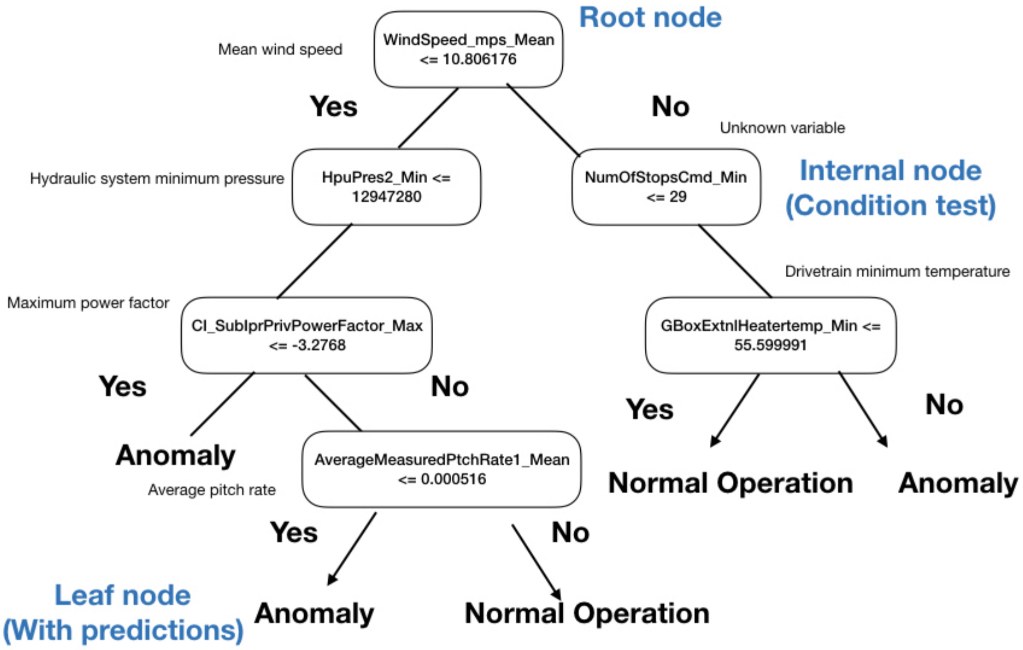

Root node
Mean wind speed
WindSpeed_mps_Mean <= 10.806176
Yes
No
Unknown variable
Hydraulic system minimum pressure
HpuPres2_Min <= 12947280
NumOfStopsCmd_Min <= 29
Internal node (Condition test)
Drivetrain minimum temperature
Maximum power factor
CI_SublprPrivPowerFactor_Max <= -3.2768
GBoxExtnlHeatertemp_Min <= 55.599991
Yes
No
Yes
No
Anomaly
Average pitch rate
AverageMeasuredPtchRate1_Mean <= 0.000516
Normal Operation
Anomaly
Yes
No
Leaf node (With predictions)
Anomaly
Normal Operation

Fig. 4.6 An example visualisation of an ensemble of decision trees for prediction making using SCADA data. The model makes predictions based on multiple conditions wherein, the threshold of SCADA feature values are evaluated.

identifying turbine operational status based on SCADA data is provided in Figure 4.6. The key essence of the model's working lies in identifying multiple conditions and determining the best possible decisions on the basis of the SCADA features' threshold values. With the presence of an ensemble of multiple decision trees (wherein, each strand learns to provide individual predictions of output), these are finally combined to generate a joint output. For our learning task, the individual decision trees are trained on separate parts of the SCADA training data.

The working process of the XGBoost model is described below in brief:-

1. Initially, the value of the objective loss function (which outlines the model's prediction error) is minimised. Here, the loss function $\mathscr{L}^{(t)}$ depends upon actual labelled values of targets in the training dataset (ground truth) y_i, and is modelled as a function of multiple distinct Classification and Regression Tree (CART) learners computed at each iteration t, as described in equation (4.15). Note that i represents individual samples for the learner in the process. Further, to minimise $\mathscr{L}^{(t)}$, second order Taylor approximation is applied, and an updated version of the objective function is obtained as per equation (4.16), with g_i and h_i representing the statistical approximations of the loss function's gradient at first and second orders respectively in this scenario.

$$\mathscr{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\right) + \Omega(f_t) \tag{4.15}$$

$$\tilde{\mathscr{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)\right] + \Omega(f_t) \tag{4.16}$$

2. The successive learners for the upcoming iterations are determined by the XGBoost model on the basis of identification of the decision trees which lead to the minimum loss in the prediction process. To accomplish this, a scoring function represented as q is utilised, which performs an evaluation of the individual learners' loss over multiple iterations in the training data, and computes the reduction in loss (with increasing number of iterations) at successive nodes. For this purpose, the *exact greedy algorithm* is used in the computation process, which can greedily minimise the value of the objective function's loss at the present time-steps t. Refer equation (4.17) for the minimisation process.

$$\tilde{\mathscr{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{4.17}$$

The global minima point for the objective function is determined by isolation of the classifier's weights in accordance with equation (4.18).

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{4.18}$$

The overall final gain obtained by the XGBoost learning model is evaluated at this point. Assuming that the learning algorithm performs a split on the root node into two individual leaf nodes represented as L_L and L_R, the decision making process at a particular instance of time is performed based on the split objective function as per equation (4.19).

$$\mathscr{L}_{split} = \frac{1}{2} \left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{4.19}$$

3. At the final stage of the prediction making process, a probability score p is computed for the different successive predictions (belonging to the range of $\{0, 1\}$), utilising

a specific cross-entropy based loss function to achieve binary classification. This is represented in equation (4.20).

$$y \ln(p) + (1-y)\ln(1-p) \text{ where } p = \frac{1}{(1+e^{-x})} \tag{4.20}$$

The final classification probability is computed with application of the Sigmoid activation function (to eventually provide importances of features leading to the predicted target). This gives the XGBoost model the ability to classify new, unseen data in binary terms, and additionally, also provides a feature importance ranking for the relevant (important) predictors utilised by the learning model. This thereby will help to provide transparency to our DL model, which we aim to achieve, when utilised in conjunction with the LSTM model, as will be discussed later.

4.3.3 Accuracy Meets Transparency: Augmenting the LSTM with XGBoost

In this section, we will discuss the novel idea of bringing together the LSTM model and XGBoost to provide both, accurate predictions of anomalies in the wind turbine as well as transparent decision making. Below, we outline the process through which the augmentation of these two learning models is achieved. As can be seen from Figure 4.7, in our approach, the LSTM takes in sequences of SCADA features as input to make its predictions, and the XGBoost learner will provide the importances for contributing features on the basis of the hidden representation of the input SCADA data induced by the LSTM. The latter part of this process is accomplished through a combination of multiple decision trees in the ensemble, which are combined utilising a meta learner to provide a final vote on the predictions to be made. Note that a similar model has previously been proposed in the medical domain for hypoxemia prediction in real-world hospitals' operating rooms (Chen et al., 2018), but we modify the architecture and extend it to the wind energy domain, giving the model the ability to continuously read SCADA logs in the input, and also facilitate transfer learning.

Below, the brief process involved in the working of the proposed model is described:-

1. SCADA data with multiple features (multivariate time-series) represented as x_t is provided as the input to the LSTM model, along with the corresponding historical labels y of normal operation/anomaly (ground truth). The LSTM network utilises these to generate the values at the output gate (o_t) in line with equation (4.13) outlined earlier.

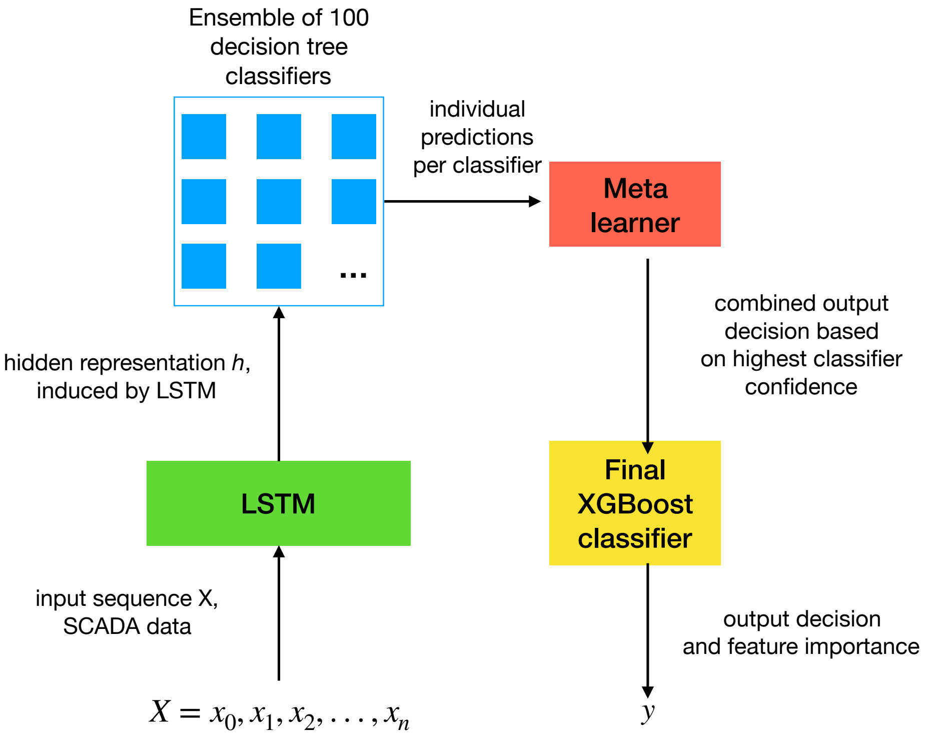

Ensemble of 100 decision tree classifiers
individual predictions per classifier
Meta learner
combined output decision based on highest classifier confidence
hidden representation h, induced by LSTM
LSTM
Final XGBoost classifier
input sequence X, SCADA data
output decision and feature importance
$X = x_0, x_1, x_2, \ldots, x_n$
y

Fig. 4.7 Augmentation of the LSTM with XGBoost, wherein, the representations from the LSTM's final hidden layer are fed to the XGBoost classification model's ensemble learner

2. The hidden representations of the SCADA data h_t (which are eventually captured through the output gate o_t's values) are forwarded and fed into a dense layer in accordance with equation (4.21). The dense layer performs a matrix multiplication of the prevailing hidden states and the weight connections matrix at the LSTM layer. Note that here, y_t represents the LSTM model's intermediate outputs (a binary value for normal operation/anomaly), and is utilised to map the LSTM's hidden states to individual sub-trees in the XGBoost model's classification ensemble.

$$y_t = W * o_t \tag{4.21}$$

3. The final hidden representations of the input SCADA data are now obtained, represented as $h_t = \{h_0, h_1, h_2 ...., h_{t-1}\}$. For more details, refer equation 4.14 above.

4. Finally, the hidden representations obtained are utilised to be fed as training samples into the XGBoost model. As already outlined before, the XGBoost model considers an ensemble of multiple decision trees in the predictive process rather than a single decision tree. To this end, consider that the XGBoost contains N weak classifiers,

wherein, each of these is trained on different portions (subsets) of the training data h_t. Here, the weights from the multiple weak classifiers are combined to obtain a final classifier, which would account for the maximum likelihood in obtaining correct predictions for the labels, given the historical ground truth for the training samples $y \in h_t$. The best overall classifier obtained after the joint vote in the ensemble is finally utilised in obtaining the final predictions $\hat{y}$, in accordance with equation (4.22). Note that $\hat{y}$ here represents the turbine's predicted operational status, with 0 representing normal operation and 1 representing an anomaly. For our study, we utilised a total of 100 different weak classifiers in the ensemble, which would function as estimators in the XGBoost model.

$$\hat{y} = \underset{y \in h_t}{\arg\max} l\left(y; y_1, \ldots, y_N\right) \tag{4.22}$$

The learning process of our augmented LSTM-XGBoost model is briefly enunciated in terms of pseudo-code in Algorithm 2.

4.4 Description of datasets and pre-processing

4.4.1 Source domain: SCADA data from the Levenmouth Demonstration Turbine (LDT)

We utilised the SMOTE oversampled LDT SCADA data described in Chapter 3 as the source domain, with 27,188 measurements at intervals of generally 10 minutes, each consisting of 102 features. There were 13,594 occurrences of anomaly and 13,594 samples for normal operation i.e. a perfect balance in these 2 classes. All features in the dataset were scaled in the range of 0-1 during the normalisation procedure, and we adopt the same step later in the target domain to facilitate transfer learning across different datasets.

4.4.2 Target domain: SCADA data from ENGIE La Haute Borne wind farm

We utilised the SCADA data from the La Haute Borne wind farm described in Chapter 3 as the target domain. The data used contains a total of 840,380 samples (with 210,095 for each of four different turbines in the wind farm), generally at 10-minute intervals. After the application of K-Means++ clustering, there were 633,365 samples for normal operation,

Algorithm 2: Pseudo-code for augmentation of the LSTM learner with the XG-Boost model

```
Input: Mulivariate time-series of 102 SCADA features $X = \{x_1, \cdots, x_n\}, x_i \in \mathscr{X}$
Result: Turbine operational status $y$ (normal/anomaly) – 2 binary output classes
/* Encode fault categories (functional groups) as binary values */
for $y_i$ in $y$ do
    if $FunctionalGroup == 'NoFault'$ then
        FunctionalGroup ← 0 end
        else
            FunctionalGroup ← 1 end
        end
        for $X_i$ in $X$ do
            /* Adjust the target to depend on input at past time-steps $(t-1)$ */
            FunctionalGroup ← $FunctionalGroup.shift(-1)$
        end
        /* Remove NaN values resulting from this shift */
        $X_t$ ←$X_t[: -1, :]$
        /* Reshape the input data to a 3D array - which is the required
           format of input to the LSTM */
        $X_t$ ←$X_t[:, None, :]$
        /* Train the LSTM model with the pre-processed input and output data */
        lstm_model.train($X_t$,$y$)
        /* Retrieve hidden representations from LSTM's final hidden layer */
        $h_t \leftarrow out(lstm\_model\_dense)$
        /* Train the XGBoost classifier in the source domain using hidden
           representations obtained from the LSTM */
        xgb_model.train($h_t$,$y$)
        /* Predict turbine operational status in the source domain */
        $y_{predicted\_source} \leftarrow xgb\_model.predict(X_t)$
        /* Make inference in the new target domain through transductive
           transfer learning */
        $y_{predicted\_target} \leftarrow xgb\_model.predict(X_{targetdomain})$
```

while the remaining 207,015 samples were instances of anomalies. In this case, synthetic data was not created as the learning model was not actually re-trained in the target domain, but we were primarily interested in evaluating the effect of transferring the source domain model to the target domain towards prediction of anomalies. Thereby, the entire dataset in the target domain was utilised as the test data.

4.5 Experiments and Results

4.5.1 Training of the LSTM-XGBoost hybrid learning model utilising LDT source domain data

We utilise *TensorFlow* (Abadi et al., 2015), an open source scalable DL framework in Python to train the LSTM learning model. An 80%-20% split is utilised for the training and test datasets respectively, wherein, the training data contains 27,188 samples obtained after SMOTE and the test dataset has 4,279 samples from the original dataset of the LDT. It should be noted that the test dataset which would eventually be utilised to analyse the model's performance is a held-out segment of the original SCADA data, untouched by any oversampling. The oversampled SCADA data is used only to train the learning model for its prediction process.

Selecting the architecture and the configuration for the LSTM model depends greatly on the learning task, and varying tasks may require completely different architectures, without any given set of specific criteria for the architecture choice to the best of our knowledge. We adopt the known rules of thumb prevalent in existing literature for this purpose wherever possible, e.g. those outlined by Bengio (2012). We performed experimentations with multiple varieties of parameters and network configurations, e.g. in terms of the number of hidden layers, hidden units in the network, optimiser used etc., and finally, opted for a LSTM model architecture consisting of 2 layers wherein there are 100 hidden neurons in each layer, alongside the Rectified Linear Unit (ReLu) activation function, Adam optimiser, a learning rate of 0.001 and a training batch size of 128. The learning model was trained over a maximum of 2000 epochs for our problem. During the process of hyper-parameter optimisation, the configuration of the final model was decided based on randomised trial and error. Similarly, various comparisons were performed to evaluate and determine the learning and dropout rates, number of epochs, optimiser and its hyper-parameters etc. As our end goal was to ensure that the final model can generate predictions with a reasonably good accuracy which closely matches to the scale prevalent in present state of the art and additionally facilitate transfer learning, we opted for 2 hidden layers in the model. It should be noted that it is possible to add more hidden layers to make the network deeper, however, this leads to significant increase in the training time and complexity, while providing a negligible change in the model's performance. Thereby, the final choice of the model's architecture seemed to work reasonably well specific to our application domain and problem context. We utilised the softmax based cross-entropy as the model's loss function. The hidden representations obtained from the LSTM are finally utilised to fit (train) the XGBoost classification model, which generates the final outputs (0 denoting normal operation or 1 denoting an anomaly) in a binary context. Given that the XGBoost classifier returns a probability distribution for the obtained output classes rather than a single, discrete decision, we used a confidence threshold of 70% to classify the SCADA observations as anomaly, while in other cases, it was predicted as normal operation of the turbine.

Accuracy results To provide a meaningful comparison of the proposed model's performance with other competitive techniques which have previously been utilised for anomaly prediction in turbines, we evaluated the model's accuracy against various baselines. Given that for traditional ML models, feature selection can play a vital role in improving the model's performance, we utilised the MATLAB Classification Learner Toolbox to identify different permutations of features in the SCADA dataset, and finally applied Principal Component Analysis (PCA) to explain the features accounting for 95% of the variance in the dataset. Additionally, to prevent overfitting, 5-fold cross validation was utilised for the training dataset. Figure 4.8 describes the top 20 features for our SCADA data, alongside their factor weights which were obtained after PCA.

- The application of a Support Vector Machine (SVM) with Fine Gaussian Kernel gave us an accuracy of 95.2% for the LDT data. For the SVM model, we performed optimisation based on experiments with multiple kernel functions, box constraints belonging to the range of [0.001,1000], a varying kernel scale containing positive log-scaled values in the range of [0.001,1000], and standardisation for the dataset. The different combinations of SVM models which were evaluated are summarised in Table 4.1. Zhao et al. (2017) have previously applied SVMs for anomaly prediction in turbines, achieving an accuracy of 94%, although this is for a different data source. As an interesting result, we found that Santos et al. (2015) have previously demonstrated Linear Kernel based SVMs to outperform ANNs based on their accuracy. Our results have showcased a similar trend, with the SVM model outperforming the neural network. However, note that the Fine Gaussian SVM was the most optimal for our data in terms of its performance, possibly due to wide variations in SCADA datasets (their structure, features etc.) utilised in real world.

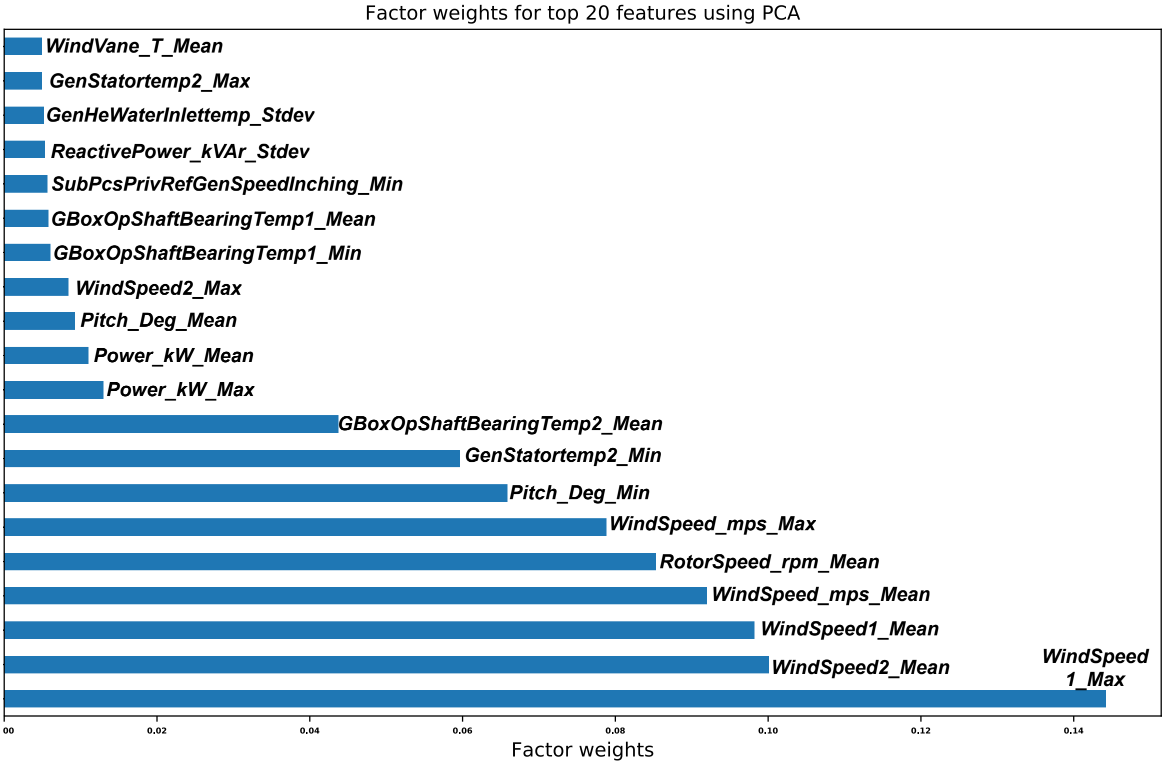


Fig. 4.8 Top 20 features obtained after PCA for the LDT SCADA data, with their factor weights – features with higher weights can better explain the variance in the data, and are thus more important

Table 4.1 Different combinations of SVMs utilised as baselines, with their key specifications

SVM Type	Kernel	Accuracy
Linear SVM	Linear	92.9%
Quadratic SVM	Quadratic	79.9%
Cubic SVM	Cubic	49.2%
Fine Gaussian SVM	**Gaussian**	**95.2%**
Medium Gaussian SVM	Gaussian	92.9%
Coarse Gaussian SVM	Gaussian	92.8%

- The application of an Artificial Neural Network (ANN) consisting of a single hidden layer and utilising the Adam optimiser provided us an accuracy of 93.81% for our dataset. We observed that adding additional hidden layers to the network led to no further improvements in the model's performance. Different types of optimisations i.e. Adam (Kingma and Ba, 2014), RMSProp (Hinton et al., n.d.) and Stochastic Gradient Descent (SGD) (Robbins and Monro, 1951), and hyper-parameter tuning was performed. Key results obtained based on the top three cases in this scenario are summarised in Table 4.2. Here, β_1 and β_2 denote the initial decay rates during estimation of the first and second moments of the gradient for the Adam optimiser. The ρ parameter for the RMSProp optimiser is used as a decay factor in computing the exponentially weighted average over the square of the gradients. The value of $Momentum = 0$ utilised for the SGD optimiser enunciates that no accelerated convergence of gradient vectors was performed, to prevent the loss function to be stuck at local minimas. Note that a total of 500 epochs was utilised to train all cases of the ANN models, as we observed that training further led to a negligible change in the model's performance.

Table 4.2 Summary of top-three optimisation parameters for the ANN

Optimiser	**Learning Rate**	**Optimisation parameters**	**Accuracy**
Adam	0.001	β_1 = 0.90, β_2 = 0.999	**93.81%**
RMSProp	0.001	ρ = 0.90	92.64%
Stochastic Gradient Descent (SGD)	0.01	Momentum = 0 (Undamped)	78.36%

Previously, Ibrahim et al. (2016) have shown that the ANN, when utilised in combination with Current Signature Analysis can provide an accuracy of 98%, although their dataset was synthetic and based on simulations, rather than real-world data in our case.

- The application of Ensemble Bagged Trees on our dataset provided an accuracy of 98.25%, which is the best accuracy achieved in our comparison with baseline models. An automatic variation in the model's learning rate was performed in the range of [0.001,1]. Table 4.3 provides a summary of the accuracy for different types of Ensemble classifiers utilised for our study. Note that Abdallah et al. (2018a) have previously reported applying Ensemble Bagged Trees for classification of root causes of failures in the turbines, but there is no evaluation and mention of accuracy in the previous study.

Table 4.4 provides a summary of comparison of the proposed learning model with the baselines. Interestingly, we observed that on training and testing the proposed model based on the original dataset before application of SMOTE, the accuracy obtained was 83.20%, which can primarily be attributed to the extreme bias which the model makes in predicting the majority of the data samples as normal operation. Given that our primary aim was to

Table 4.3 Accuracy for multiple types of Ensemble classifiers utilised

Ensemble Classifier	**Accuracy**
Ensemble Boosted Trees	95%
Ensemble Bagged Trees	**98.25%**
Ensemble Subspace Discriminant	88.4%
Ensemble Subspace KNN	95.8%
Ensemble RUSBoosted Treest	25.9%

Table 4.4 Comparison of the proposed technique's performance with related work (baselines)

Related Study	**Technique used**	**Accuracy on our data**	**Reported Accuracy in related work**
Zhao et al. (2017)	Support Vector Machine (SVM)	95.2%	94%
Ibrahim et al. (2016)	Current Signature Analysis (CSA) and Artificial Neural Network (ANN)	93.81% (Without CSA)	98%
Abdallah et al. (2018a)	Ensemble Bagged Trees	98.25%	Not reported
Proposed work	**LSTM-XGBoost hybrid model**	**96.634%**	N/A

evaluate the proposed technique's feasibility in application to an unseen, test dataset from the original SCADA logs of the turbine before applying SMOTE, clearly, the technique achieves close to state of the art performance and when compared to other DL techniques, provides a balance in the tradeoff in accuracy and transparency, which is discussed later. As a notable observation, 89.41% of faults are correctly predicted by the proposed model in the source domain, achieving a false discovery rate of 5.26%. The confusion matrix obtained based on evaluation of test data in the source domain is depicted in Figure 4.9. Table 4.5 provides more thorough details on the model's performance metrics.

Clearly, the proposed LSTM-XGBoost model outperforms conventional ANNs in terms of accuracy by up to 2.824%. Note that although our study shows that DL techniques may not always outperform more conventional ML algorithms like Ensemble Bagged Trees despite the complex, non-linear nature of SCADA data, this is likely due to smaller datasets in the wind industry, and potentially, more sophisticated models like Bi-directional LSTMs, when trained with larger samples of data (before oversampling) can help in achieving further improvements in performance of the LSTM model utilised in this chapter.

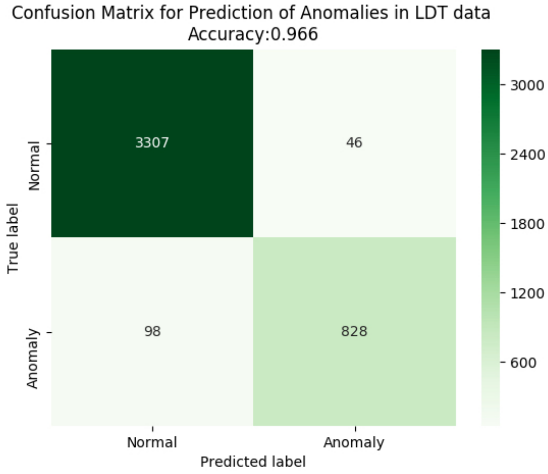


Fig. 4.9 Confusion matrix for evaluation of source domain performance with proposed model

Table 4.5 Performance metrics for the LSTM-XGBoost model used for predicting anomalies in LDT data

Measure	**Value**
TPR: Sensitivity	0.8941
TNR=SPC: Specificity	0.9862
PPV: Pos Pred Value = Precision	0.9473
NPV: Neg Pred Value	0.9712
FDR: False Discovery Rate	0.0526
ACC: Accuracy	0.9663
F1 score	0.92

Transparency results With the valuable capability of the XGBoost model in providing the importances of features on the basis of F-scores, the model can help in achieving transparency by identifying the key features in the SCADA data which are most likely contributing to any potential anomaly in the turbine's operation. Figures 4.10 and 4.11 depict this provision of the XGBoost model. It is essential to note that the feature importances for the model are arranged in descending order, i.e. features assigned a higher F-score are more likely to contribute to the identified anomaly in comparison to those outlining a lower F-score.

As an example, refer Figure 4.10, which outlines an anomaly condition in the gearbox. It can be visualised that amongst the key features contributing to this anomaly, *MainBearingtemp1_Mean* and *GearBoxTemperature_DegC_Max* are highly ranked. This is reasonable and can likely be attributed to an operational inconsistency (overheating) in the gearbox's high speed shaft bearings and in general, the gearbox housing assembly as a whole. Figure 4.11 shows another example for an anomaly in the turbine's yaw system. In this case, the highest feature importance is attributed to *NacelleOrientation_Deg_Stdev* and *RotorSpeed_rpm_Stdev*, alongside multiple other features with lower F-scores. In a wind turbine, the yaw system is located between the nacelle of the turbine and the tower, wherein, the yaw brakes play a pivotal role in holding the nacelle properly in an azimuthal oriented position (Kim et al., 2016). We can infer from this scenario a highly likely situation of disorientation of the nacelle, which directly contributes to the yaw system anomaly. Also, in situations wherein the rotor is not perpendicular directly to the prevailing wind direction, a yaw error generally occurs in the turbine (Wan et al., 2015b). This is clearly in line with the predicted feature importance, highlighting a problem in the rotational behaviour of the turbine's rotor, which is contributing to this anomaly. Note that we performed the analysis of feature importances on a randomised, but representative portion of the testing dataset containing faults in different sub-components (functional groups) including wind condition alarms, pitch system, yaw brake, gearbox, hydraulic system etc. based on ground truth in the LDT's alarm logs. We found that the proposed model generated reasonable feature importance inferences for different situations, with the most reasonable causes being identified for anomalies in the pitch system, followed by the gearbox with around 70% relevance. We however observed that in some specific cases for the hydraulic system and anomalies arising due to Moisture Vapour Transmission Rate (MVTR), the feature importances were not relevant to the results. Note that while we found the feature importances to be a reasonable analysis of the prevailing fault condition, providing an ultimate judgement on the exact cause of faults is case-specific, and given the fact that historical data for exact features which contribute to such faults is unavailable owing to the commercially sensitive nature of the alarm logs, it is integral to combine judgements based on expert decisions with the feature importances obtained to achieve most reliable results, instead of completely depending on autonomous decisions made by the model.

Based on our evaluations for the accuracy and transparency, we found that the LSTM model can support wind farm operators in identifying the situations wherein a fault would occur, based on the SCADA time-series measurements from the sensors. Additionally, with the interpretable and transparent nature of the decision tree, the turbine operators can understand the context of the predicted faults, which includes the identification of specific measurements in the SCADA data likely leading to the fault. As wind turbine engineers and technicians are trained in O&M activities with expert domain knowledge for averting failures, given any irregularities in the SCADA measurements, they can take appropriate decisions and utilise the necessary steps required to perform maintenance and repair actions for the specific turbine sub-components. This contributes to making ML a reliable source of intelligent decision support for the turbine operators, as until recently, owing to the black-box nature of conventional models (especially deep learners), techniques of such capabilities have not seen much uptake in applying them for practical purposes in wind farms.

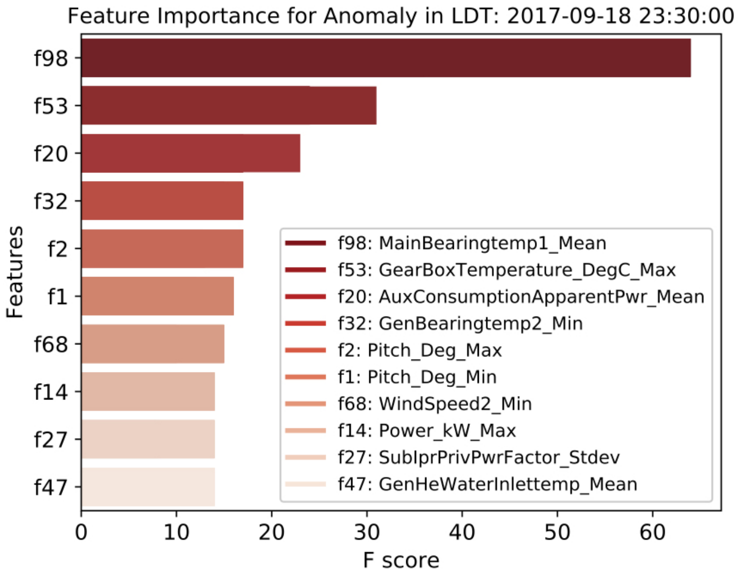


Fig. 4.10 Example 1: Feature importance plot during an actual anomaly in the turbine's Gearbox (2017-09-18 23:30:00)

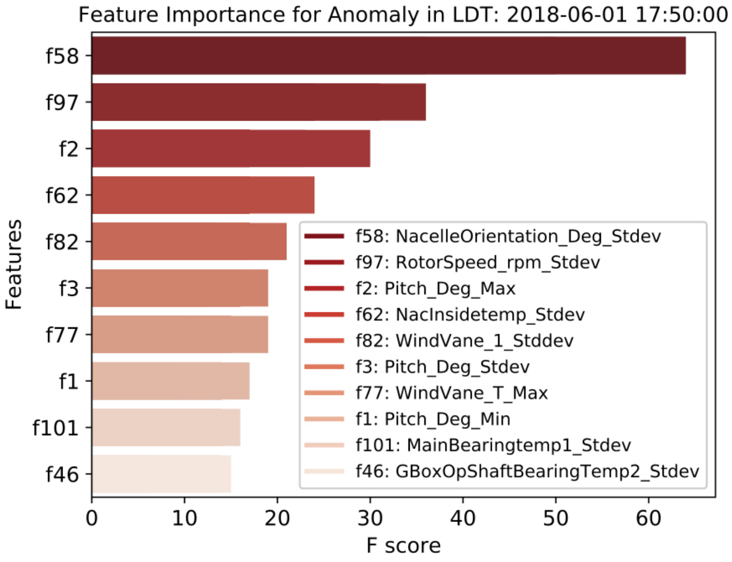


Fig. 4.11 Example 2: Feature importance plot during an actual anomaly in the turbine's Yaw System (2018-06-01 17:50:00)

4.5.2 The Essence of Transfer Learning

ML models are conventionally utilised for decision making in a specific problem task, and generally trained using a dedicated dataset in the intended target domain. However, in case of real-world applications of ML models, this generally creates key constraints on the scalability of the model, as the ML models would require re-training for new tasks with brand new datasets, even if the new target domain is very similar and related to that of the originally trained model. This particularly makes collecting new, additional data an expensive endeavour, and is often infeasible. Likewise, this holds true for the wind energy sector, as obtaining large and reliable sets of new data from new turbines and wind farms is generally not possible readily (Chatterjee and Dethlefs, 2020e). We introduce transfer learning in this section, and further discuss its applicability to the LSTM-XGBoost hybrid learning model we have proposed earlier.

The key idea behind transfer learning is to facilitate greater usability of existing knowledge in a manner that any information in one domain (the source domain) can be utilised optimally to provide a suitable generalization in a new, unseen domain (the target domain), which may be closely related or different from the source. This helps to make the model's learning process, both, more accurate as well as improving computational efficiency in cases wherein historically labelled training data is not available. There are some common distinctions made between the various types of transfer learning, notably inductive and transductive learning. In case of inductive transfer learning, the model would learn from the historically labelled examples provided in training data in the source domain and would generate predictions for new, unseen examples from test data in a different target domain (Kaboli, 2017). More specifically, inductive transfer learning aims to develop a generic trained AI model (as in traditional supervised ML) wherein, during training of the AI model, only the train data portion is encountered by the model. In our study, we focus on transductive transfer learning, wherein the source domain already has historical labels in the dataset (both for the training data as well as test data) based on which the AI model can be trained, but there are no labelled examples available (neither for train data nor test data) in the different target domain. While there is no availability of the ground truth labels for the test data beforehand in case of transductive transfer learning, the predictions made leverage the patterns learnt from the training data (with labels) and the features (in the test data) for inference during the learning process for knowledge transfer. More details on inductive and transductive learning and the distinctions between them can be found in Kaboli (2017); Tian et al. (2021); Xu et al. (2019). In formal mathematical terms, we consider a domain $\mathscr{D}$ to be formed of a specific set of features $\mathscr{X}$ in its feature space, alongside a marginal probability distribution of the domain's data denoted as $P(X)$ (Pan and Yang, 2010). In such a scenario, it is possible to represent the domain mathematically as per equation (4.23).

$$D = \{\mathscr{X}, P(X)\}, \tag{4.23}$$

Here, $X = \{x_1, \cdots, x_n\}, x_i \in \mathscr{X}$ describes the model's training dataset consisting of n samples. Consider the source domain denoted by $D_S = \left\{(x_{S_1}, y_{S_1}), \ldots, \left(x_{S_{n_S}}, y_{S_{n_S}}\right)\right\}$, wherein, individual samples in the training dataset are represented as $x_{S_i} \in \mathscr{X}_S$ and $y_{S_i} \in \mathscr{Y}_S$ represents the ground truth (actual labels) in the dataset. These samples can thereby be mapped to a corresponding target domain represented as $D_T = \left\{(x_{T_1}, y_{T_1}), \ldots, \left(x_{T_{m_T}}, y_{T_{m_T}}\right)\right\}$, where $x_{T_i} \in \mathscr{X}_T$ and $y_{T_i} \in \mathscr{Y}_T$ denotes the ground truth in the target domain, which is to be predicted by the learning model. With these conditions, transductive transfer learning can be expressed in mathematical terms as enunciated below:-

- The source and target domains, which are denoted by D_S and D_T respectively, are different. However, they are related closely to each other, thereby, $D_S \neq D_T$.
- The final goal to be achieved in both the domains for source and target is same, thereby, $\mathscr{Y}_S = \mathscr{Y}_T$.
- The distribution of features in the dataset's feature space across the source and target domains is different from each other. Thereby, $\mathscr{X}_S \neq \mathscr{X}_T$.
- The distribution of the source and target domains in terms of marginal probability is different from each other. Thereby, $P(X_S) \neq P(X_T)$.

In case of our problem task, transductive transfer learning would optimally leverage historical training data in the source domain D_S, and utilise the knowledge and experience to facilitate prediction making for the target task $\mathscr{Y}_T$ at hand, which would prevail in the new, but closely related domain D_T. This would make the learning model capable of generating predictions in scenarios wherein, there is unavailability of labelled training data in the target domain to achieve a dedicated model. We utilise the LSTM-XGBoost hybrid learning model in this chapter, and apply it for predicting anomalies in operations of multiple turbines in an onshore wind farm, with the model being only trained on historical data from a single offshore wind turbine. Figure 4.12 enunciates the transfer of knowledge between domains for the scenario outlined.

Figure 4.13 shows a graphical visualisation of the approach, wherein, the LSTM-XGBoost model trained in the source domain with the LDT data is used for predicting anomalies in the target domain with the ENGIE data.

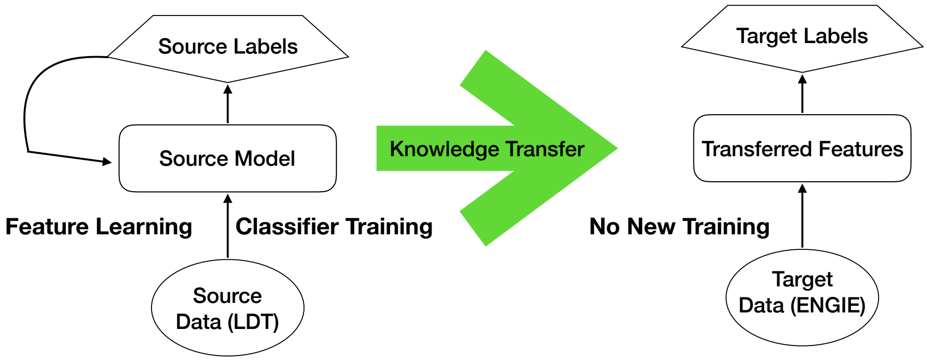


Fig. 4.12 Transfer of knowledge between different domains – Past experience and knowledge in the source domain is leveraged for making predictions in the target domain, without requiring additional training from scratch

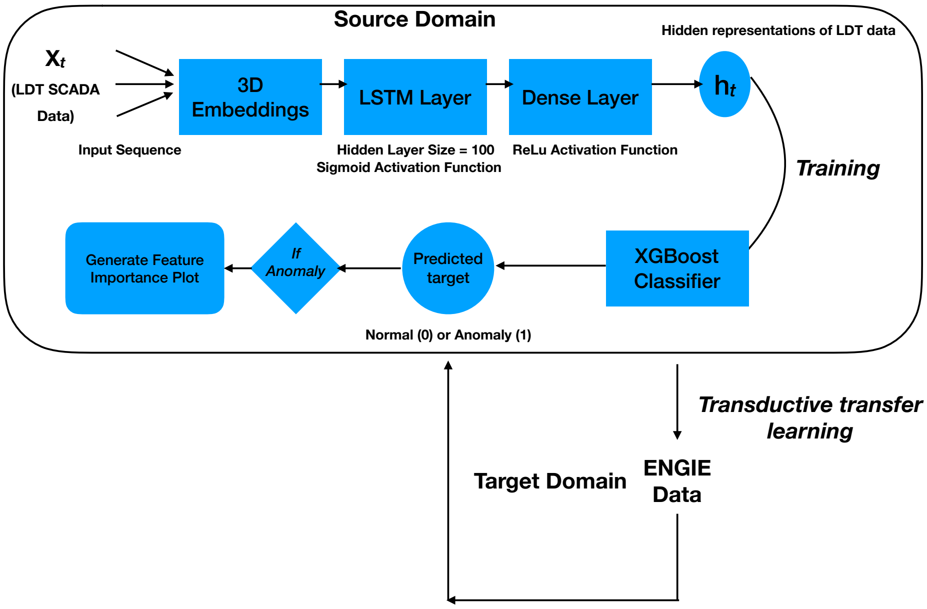


Fig. 4.13 Depiction of the proposed LSTM-XGBoost augmented model's architecture. The model trained in the source domain can be used for inference in the target domain through transductive transfer learning.

From Source to Target Domain: The Model Transfer To perform decision making for anomaly prediction in the unlabelled ENGIE dataset, the LSTM-XGBoost which was originally trained in the source domain with the LDT data is utilised to transfer any knowledge which was acquired by the model in the source domain to the target domain for ENGIE data. As the features which were learnt in the source domain are directly transferred to the target domain, this eliminates the requirements for any additional training with the ENGIE data. On comparing the predicted labels with the estimated K-Means++ labels for

ENGIE data, our model demonstrated an accuracy of 65.42% in the target domain, wherein, it could identify 85.48% anomalies correctly. Note that the lower accuracy achieved (in comparison to the source domain) is primarily due to a high rate of false alarms predicted by the transferred model – in 41.14% cases. Table 4.6 provides more thorough details on the model's performance metrics. We understand and enunciate that this rate of false alarms is quite high (as at times, false alarms can lead to redundant O&M activities and increased costs), yet the figures obtained are encouraging, given that prediction making is performed entirely on the basis of training of the model in a different domain, wherein, the ENGIE training data is not consulted at all in the learning process. The confusion matrix outlining model evaluation results in the target domain after knowledge transfer from the learnt source domain model is depicted in Figure 4.14.

Table 4.6 Performance metrics for predictions on the ENGIE data

Measure	**Value**
Sensitivity	0.9254
Specificity	0.4045
Precision	0.5886
Negative Predictive Value	0.8548
False Positive Rate	0.5955
False Discovery Rate	0.4114
False Negative Rate	0.0746
Accuracy	0.6542
F1 Score	0.7196

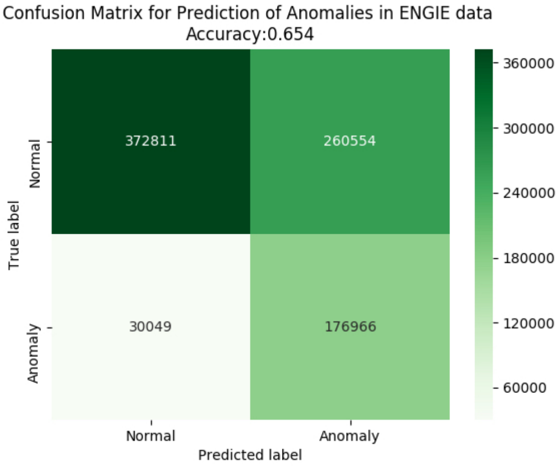


Fig. 4.14 Confusion matrix for evaluation of target domain performance with proposed model after knowledge transfer

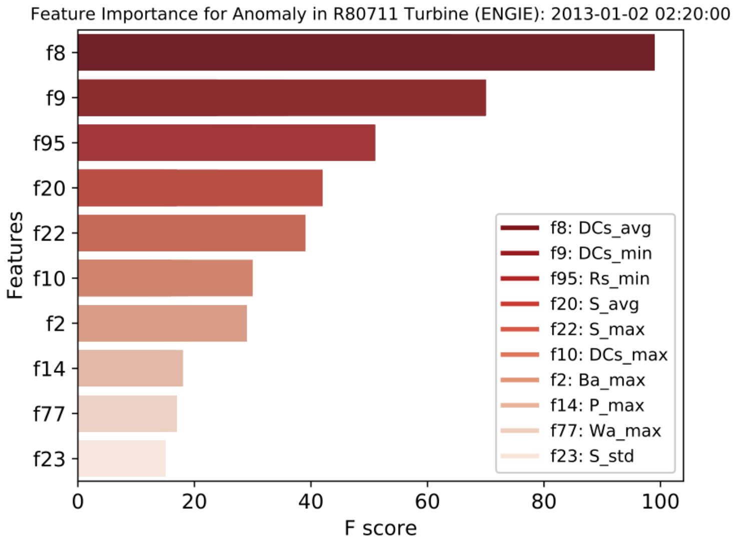


Fig. 4.15 Example 1: R80711 turbine having an anomaly as predicted by the model, alongside the feature importances (2013-01-02 02:20:00)

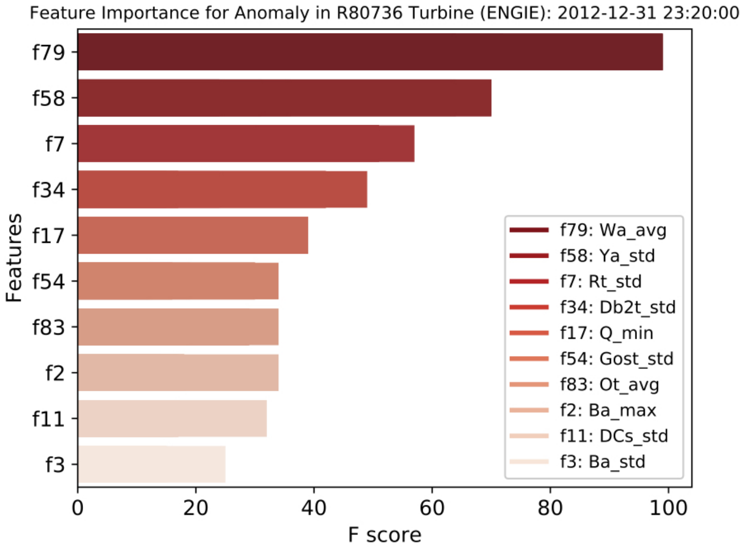


Fig. 4.16 Example 2: R80736 turbine having an anomaly as predicted by the model, alongside the feature importances (2012-12-31 23:20:00)

As outlined in the case of source domain, a similar transparency analysis was performed using the XGBoost model on the target data for identifying and extracting the feature importances for the subset of features which most likely contribute to the anomaly. Below, we discuss two different cases for feature importance, in situations of an actual anomaly in two different turbines in the target domain (R80711 and R80736) with different (unique) time-stamps. In Figure 4.15, it can be seen that *DCs_avg* and *DCs_min* are amongst the leading features contributing to the anomaly. This can most likely be inferred to be caused as a consequence of the generator converter speed having an outlier and not falling within the suitable range, which attributes an error in the turbine's generator as a direct result. In another situation as shown in Figure 4.16, the primary features contributing to the anomaly are enunciated to be *Wa_avg* and *Ya_std*. These features show that there is an unusual value in case of the absolute wind direction and nacelle angle, respectively. We can likely infer this situation to be arising due to an anomaly in the turbine's rotor, as the third feature identified in the analysis – "*Rt_std*" attributes an uneven rotor operational speed, which may be arising due to imbalance in the rotor – a situation which is directly caused due to wind shear and torque oscillation prevailing in the turbine's nacelle (KK Wind Solutions A/S), 2019).

ROC Curves The Receiver Operating Characteristics (ROC) curve is depicted in Figure 4.17, outlining the predictive performance of the learning model in case of the binary classes – normal operation (0) and anomaly (1) for the LDT source domain. Here, the Area Under Curve (AUC) provides a vital parametric measure of the overall performance of the classification model in its decision making (with the values closer to 1 being the better), which is 0.96 for class 0 and 0.90 for class 1. This shows that the model is able to learn the discriminatory features and patterns in the SCADA dataset, providing a high degree of separability between the two classes in the data. In comparison to this, the model's performance in the target domain for the ENGIE data is shown in Figure 4.17. As we can see, there is clearly a reduction in the AUC which is expected, but it is still 0.75 in case of the anomaly class. This is promising and shows that in the given scenario in the new domain, the model is able to predict anomalies with good reliability. Note that the AUC is obtained as only 0.55 in case of normal operation, as is expected due to the high false positive rate, and we believe this is an interesting problem which can be tackled in the near future, by utilising e.g. more *useful* SCADA data consisting of higher number of samples, more relevant features and a larger collection of historical alarms to train the model.

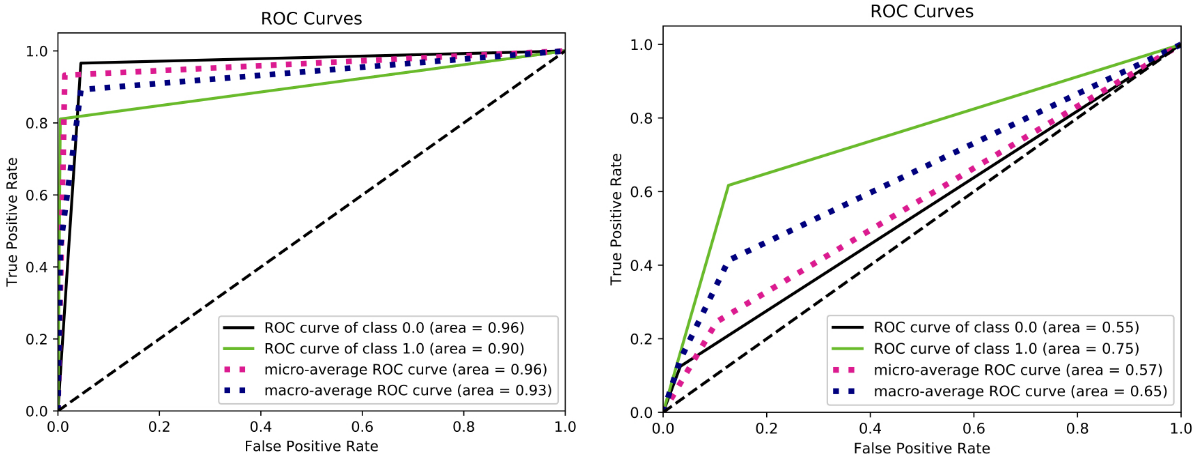


Fig. 4.17 Left: ROC curve for the LDT data (source domain), wherein, (0) denotes normal operation and (1) denotes an anomaly; Right: ROC curve obtained for the ENGIE data (target domain).

4.5.3 Discussion

The proposed model, to the best of our knowledge, is the first in the wind energy sector to explore the applications of DL tailored for anomaly prediction in wind turbines with transparent decision making, as well as extending it to a new domain through transfer learning, which eliminates the additional requirements for training the model in the new domain. We have found transfer learning to be highly promising and feasible for the wind industry, and it can play an integral role in situations wherein labelled training data is not readily available. This gives wind farm operators the ability to utilise the enormous amounts of SCADA data which are generated from multiple sensors in the turbines, but has not been effectively and fully utilised until recently for developing intelligent judgements for preventing impending faults in the turbines and their sub-components. Further, with the added capabilities of transparency, the model can aid explainable decision making and predictions, thus encouraging the uptake of AI models by turbine operators. In specific situations with lack of sufficient SCADA data for training the models (e.g. in case of new wind farms which have not been in operation for long), data from different wind farms can be utilised to facilitate decision support. This can help support the operators in feasibility evaluation of operations of new wind farms wherein SCADA data has been generated via simulations, but lack the labelled history of failures. In summary, the following key contributions are made through utilisation of the proposed model:-

- A LSTM model is utilised towards anomaly prediction in wind turbines, and achieves an accuracy of 97%. The application of this model confirms that DL based models, which can consider and capture time-series information are capable of outperforming models which only consider the local context in prediction making.

- A novel hybrid model is presented in this chapter, combining an LSTM for accurate classification of anomalies with a XGBoost classifier model for extracting the feature importances. This helps the model to overcome a critical shortcoming in conventional deep learners, which generally operate as black-boxes and do not present rationales behind their decisions. The proposed model is able to provide detailed diagnoses for cases wherein an anomaly occurred in the turbine, which we find to be relevant and in line with human analyses for expected features causing the faults in different turbine sub-components.

- Transfer learning is performed and its feasibility is demonstrated, through porting of the model learnt to data from an onshore wind farm, wherein, 85% of anomalies are correctly identified. This result is promising considering the fact that the data in the target domain is completely unseen for the purposes of decision making, and this method can thereby be utilised in situations with little or no data for making predictions towards enhancing the reliability of wind farms.

In future, the model can be extended with more sophisticated and tailored DL algorithms, and alternative algorithms can be explored towards transparency, such as attention mechanisms. Also, given that the target domain evaluations have highlighted a high rate of false alarms, minimising these false alarms can be an integral component of future study in this domain. Exploring the applications of natural language processing for generating human-intelligible error logs for turbines would be possibly the most interesting part of future work, which can be tailored for targeted individuals (engineers and technicians) possessing different levels of expert domain knowledge. In the next section, we showcase a specific application of the proposed model in facilitating offshore vessel transfer planning.

4.6 Beyond Anomaly Prediction: Deep Reinforcement Learning for Offshore Vessel Transfer Planning

In this section, we explore the promise of DL beyond accurate predictions of faults, through its wider application in decision making and offshore vessel transfer planning. In offshore wind turbines, maintenance operations present a significant challenge, as in situations wherein a failure occurs in the turbine, maintenance personnel alongside the requisite spare parts need to be transported through vessels and/or helicopters based on the type of fault and its priority (Halvorsen-Weare et al., 2013). The service actions are generally performed using Crew Transfer Vessels (CTVs), and this gives rise to an important consideration on framing appropriate strategies for vessel transfer, which can take into account the fault types and their characteristics, alongside additional metrics such as climate conditions and availability of the vessel fleet (Dalgic et al., 2015). Present-day CTVs generally make an average of around six trips for maintenance actions per year (ORE Catapult, 2019), and in each trip require careful planning and considerations for the optimal fleet size as well as appropriate vessel dispatch strategies to the targeted turbines (wherein, the anomaly/failure occured). There are some newer types of vessels, including Service Operation Vessels (SOVs), which make more trips routinely to the turbines and can sustain rougher and more harsh climatic conditions, but involve significant periods of stay in the sea for performing service operations on multiple turbines in a single transfer (Siemens, 2018). Given that planning in advance for situations of failures demands preventive maintenance actions by engineers and technicians, determination of the appropriate vessels to be assigned to the proper turbines and deciding the order of transfers in case of a wind farm with multiple turbines is a critical process involving short-term decision making (Dawid et al., 2018). Optimisation of the vessel transfers under such situations from the onshore ports/offshore stations to the appropriate turbines which require maintenance operations (Halvorsen-Weare et al., 2013) can play a pivotal role in helping wind farm operators save Operation Expenditures (OPEX), alongside multiple other costs routinely involved in the process (Dalgic et al., 2015).

Reinforcement Learning (RL) (Sutton and Barto, 2018) is a specialised branch of ML that can utilise optimisation strategies to make software agents and systems automatically determine optimal decisions to make in a given context and environment, with the end goal of maximising the performance (which is termed *reward* in the RL terminology). When DL meets RL, Deep Reinforcement Learning (DRL) (Mousavi et al., 2018) facilitates applications of neural networks in conjunction with traditional RL algorithms and has been highly successful in various applications, ranging from playing board games and controlling robots, to making decisions in autonomous vehicles (Kasim, 2016). Owing to the rough environmental conditions which offshore wind turbines face and the uncertainty in operating conditions, the application of DRL algorithms is an ideal choice for their maintenance planning.

We explore the application of DRL for data-driven maintenance planning of offshore vessel transfers, and specifically focus on the ability of DRL algorithms to learn effective policies based on the real experience which turbine operators have with their expertise and

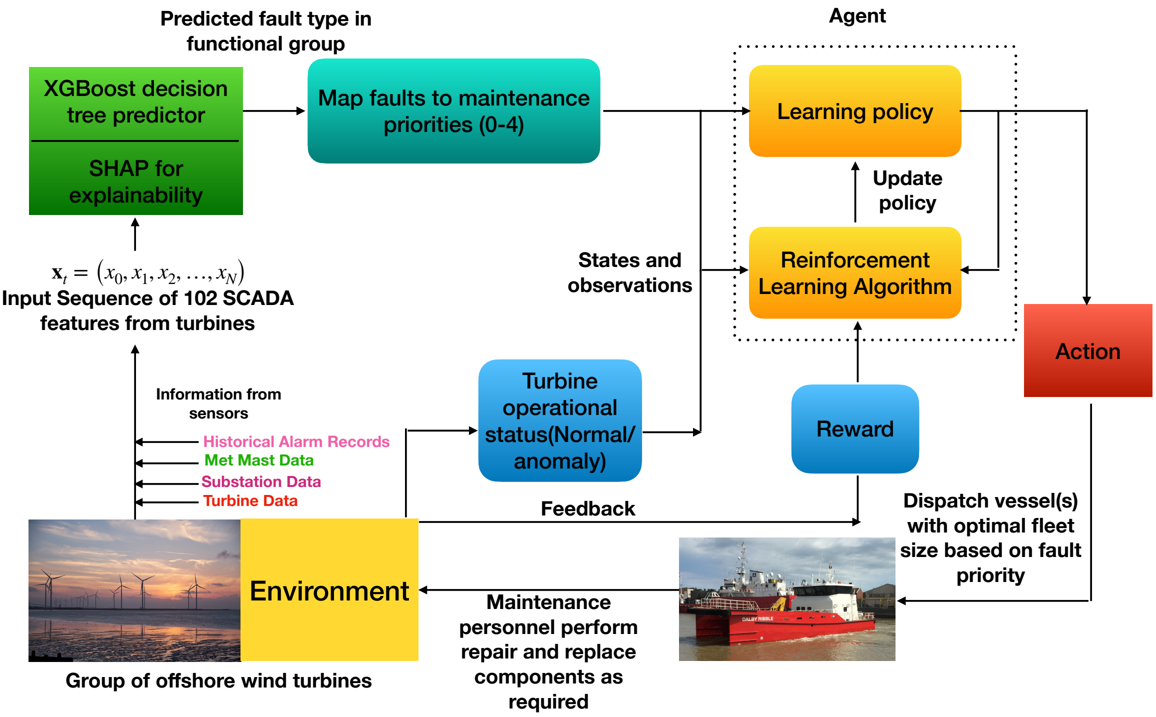

Predicted fault type in functional group
Agent
XGBoost decision tree predictor
SHAP for explainability
Map faults to maintenance priorities (0-4)
Learning policy
Update policy
Reinforcement Learning Algorithm
States and observations
$\mathbf{x}_t = (x_0, x_1, x_2, \ldots, x_N)$
Input Sequence of 102 SCADA features from turbines
Action
Information from sensors
Historical Alarm Records
Met Mast Data
Substation Data
Turbine Data
Turbine operational status(Normal/ anomaly)
Reward
Feedback
Dispatch vessel(s) with optimal fleet size based on fault priority
Environment
Maintenance personnel perform repair and replace components as required
Group of offshore wind turbines

Fig. 4.18 Depiction of the proposed framework – XGBoost + SHAP can provide fine-grained transparent and accurate prediction of anomalies in the turbine, which is utilised for mapping maintenance priorities for decision support via reinforcement learning

domain knowledge, as well as simulated experience which is learnt by the ML models during condition monitoring. In specific, the following are the key areas explored in this section:-

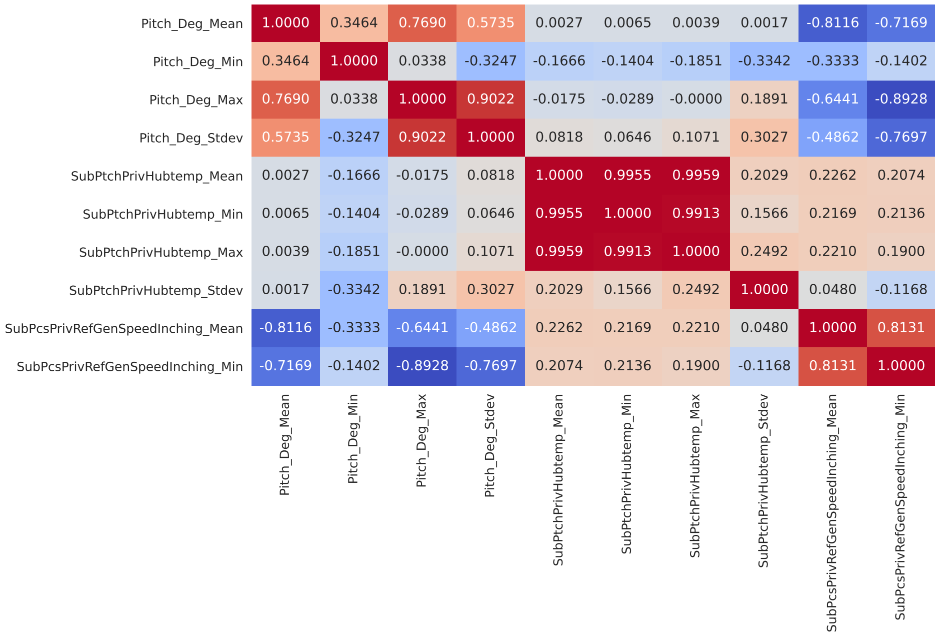
- **Explainable anomaly prediction:** With the LDT SCADA data, we create a simulated environment for an offshore wind farm to learn from historical failures while considering multiple dynamic variables (e.g. weather conditions, types of fault, priorities of repairing the turbine sub-components etc.), and utilise the XGBoost model for predicting the faults. Additionally, SHAP (Lundberg and Lee, 2017b), a novel feature explainer technique is utilised to provide more fine-grained transparency and explainability which is later mapped to maintenance action priorities to determine vessel transfer strategies.

- **Optimisation for vessel transfer planning:** A reward function is proposed to train a Deep Q-learning (DQN) algorithm, which is an extremely popular type of Q-learning algorithm (Jang et al., 2019). We use the DQN algorithm for optimising vessel transfer decisions and ensuring effective transfer strategies to turbines. The proposed algorithm is additionally compared with baseline models i.e. Q-learning (Ohnishi et al., 2019), State–Action–Reward–State–Action (SARSA) (Wen et al., 2019) and Expected-SARSA (Van Seijen et al., 2009).

The proposed framework is visually enunciated in Figure 4.18. The application of this DRL algorithm ensures minimisation of possible O&M costs which are inccurred, while additionally giving the wind farm operators the ability to manually specify vessel transfer priorities and constraints such as those based on transfer distance, fault type etc., helping facilitate the learning process in real-life. This also provides the ability to add or update the algorithm with any new constraints which may arise in the future, without performing complex mathematical modelling. With policy-dependent reward signals, this technique facilitates prioritisation of maintenance actions for critical types of faults, while eliminating or delaying transfer actions which are unnecessary or less vital. Based on the type of fault which occurs and its severity, the algorithm provides wind farm operators the provision to decide optimal transfer fleet size, which can significantly reduce O&M costs. To the best of our knowledge, this area of integration of fault prediction based AI algorithms with DRL techniques is the first to be applied in the wind industry for offshore vessel transfer planning, while considering the deployed environments of turbines as well as multiple fault types generally prevalent in present-day turbines.

4.6.1 Data utilised for the experiments

From the original LDT SCADA data described in Chapter 3 with 21,392 samples and 102 features, SMOTE was performed again, this time to hypothetically simulate data for an offshore wind farm containing five different turbines (denoted within the range 9000 to 9004, wherein, both numbers are inclusive). At this stage, since simulating a large wind farm creates challenges in terms of computational complexity and scalability, to ensure effective performance needed for scalable ML, we estimated the correlation between the different SCADA features. It was found that of the 102 features, multiple redundant features had strong correlation, of which few could be eliminated as they did not contribute to any improvements in the model's performance and accuracy metrics. We eliminated the redundant features which possessed correlation higher than 0.95. This led to a total of 233,072 samples for the study (across five different turbines), of which, each turbine had approximately 46,614 samples and 57 features specific to it. Figure 4.19 depicts a closer view of some of these features. With the key motivation in this approach being the identification of maintenance priorities for actions based on the types of faults which occurred in the turbines, the functional groups (different fault categories) in our dataset were mapped to a scale of 0 (wherein, no maintenance action is required) to 4 (urgent maintenance action is needed) on the basis of existing literature (Borchersen et al., 2012). These are finally utilised to construct the learning model, and perform further experiments. Figure 4.20 illustrates the maintenance priorities varying across the functional groups and five turbines in the wind farm through a 3D scatter plot. As can clearly be seen, there are multiple varying maintenance priorities on the scale of 0-4 for each turbine, spanning the 14 different functional groups in which anomalies occur.



	Pitch_Deg_Mean	Pitch_Deg_Min	Pitch_Deg_Max	Pitch_Deg_Stdev	SubPtchPrivHubtemp_Mean	SubPtchPrivHubtemp_Min	SubPtchPrivHubtemp_Max	SubPtchPrivHubtemp_Stdev	SubPcsPrivRefGenSpeedInching_Mean	SubPcsPrivRefGenSpeedInching_Min
Pitch_Deg_Mean	1.0000	0.3464	0.7690	0.5735	0.0027	0.0065	0.0039	0.0017	-0.8116	-0.7169
Pitch_Deg_Min	0.3464	1.0000	0.0338	-0.3247	-0.1666	-0.1404	-0.1851	-0.3342	-0.3333	-0.1402
Pitch_Deg_Max	0.7690	0.0338	1.0000	0.9022	-0.0175	-0.0289	-0.0000	0.1891	-0.6441	-0.8928
Pitch_Deg_Stdev	0.5735	-0.3247	0.9022	1.0000	0.0818	0.0646	0.1071	0.3027	-0.4862	-0.7697
SubPtchPrivHubtemp_Mean	0.0027	-0.1666	-0.0175	0.0818	1.0000	0.9955	0.9959	0.2029	0.2262	0.2074
SubPtchPrivHubtemp_Min	0.0065	-0.1404	-0.0289	0.0646	0.9955	1.0000	0.9913	0.1566	0.2169	0.2136
SubPtchPrivHubtemp_Max	0.0039	-0.1851	-0.0000	0.1071	0.9959	0.9913	1.0000	0.2492	0.2210	0.1900
SubPtchPrivHubtemp_Stdev	0.0017	-0.3342	0.1891	0.3027	0.2029	0.1566	0.2492	1.0000	0.0480	-0.1168
SubPcsPrivRefGenSpeedInching_Mean	-0.8116	-0.3333	-0.6441	-0.4862	0.2262	0.2169	0.2210	0.0480	1.0000	0.8131
SubPcsPrivRefGenSpeedInching_Min	-0.7169	-0.1402	-0.8928	-0.7697	0.2074	0.2136	0.1900	-0.1168	0.8131	1.0000

Fig. 4.19 Correlation matrix with low-level view of some features. We chose to eliminate the redundant features with $> 95\%$ correlation.

4.6.2 Learning model

Applying XGBoost with SHAP towards predicting faults

The initial goal in our approach is to predict faults in the turbines within the simulated wind farm, given a set of SCADA features. This is integral, as identifying the fault type and its severity plays a pivotal role to make appropriate decisions regarding maintenance priorities, understanding the context of a fault, and planning on whether or not to dispatch vessels. In this approach, we again utilise the XGBoost model which was outlined earlier, and alongside the predictive model, we additionally utilise SHAP, a Python library (Lundberg and Lee, 2017b) for generating fine-grained transparent decisions. Basically, SHAP utilises game theory for assigning shapley values, which help in credit allocation and provide local

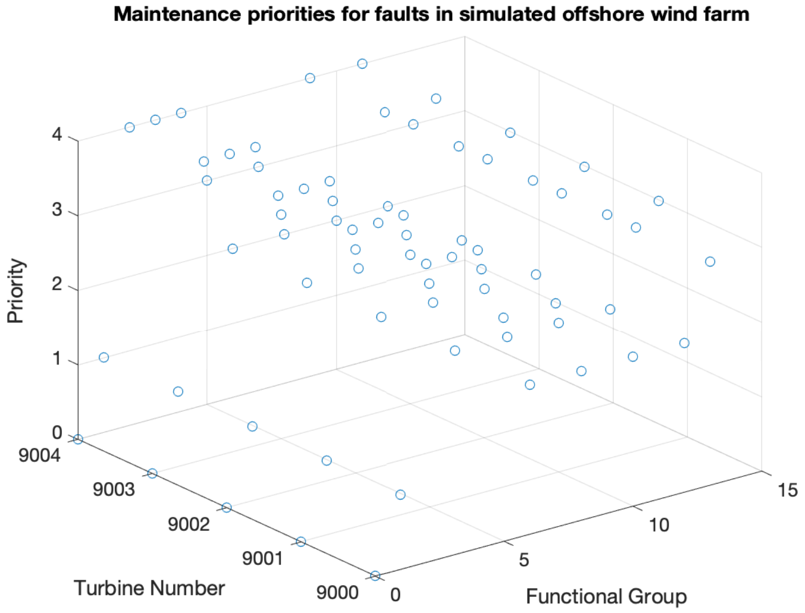


Fig. 4.20 Maintenance priorities mapped to various functional groups across five turbines. The priorities are in the range of 0 (no maintenance action required) to 4 (urgent maintenance required).

explainations, providing details and thorough analyses of contributing features which are most likely causing the predicted decision in the XGBoost model. Based on the final predictions of faults obtained from this XGBoost + SHAP model, the maintenance priorities are identified, which is ultimately used for training the DRL algorithm as discussed next.

Utilising Deep Q-Network for planning offshore vessel transfer

Deep Q-learning (Mnih et al., 2013) is a special type of RL technique based on the conventional Q-learning principle (Watkins and Dayan, 1992), which basically aims to maximise the performance of the learning agent within the given virtual (simulated) environment (Jang et al., 2019; Ohnishi et al., 2019). During Q-learning, the key principle relies on action values (referred to as Q-values), which are continuously updated iteratively for ensuring that the rewards in case of a given episode are maximised. The Deep Q-learning technique provides an additional refinement in the efficiency and effectiveness of Q-learning algorithms, by utilising DNNs in conjunction with the conventional reward functions in Q-learning, in order to approximate the values of rewards and preserve the relative importance during the learning process. The development of the Deep Q-Network (DQN) utilised in this chapter is briefly described below:-

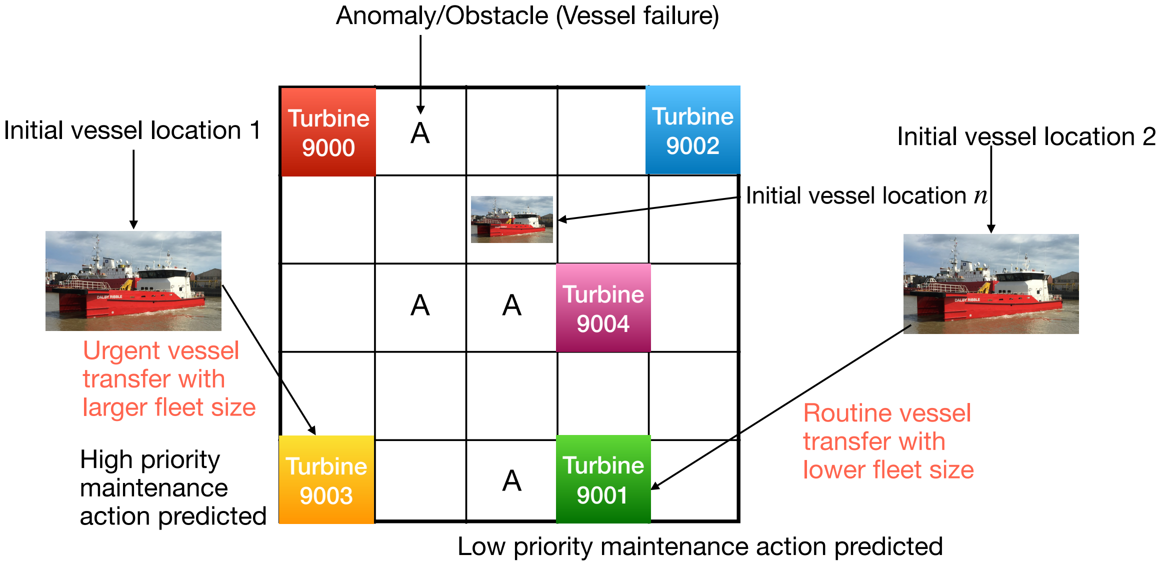

Anomaly/Obstacle (Vessel failure)
Initial vessel location 1
Initial vessel location 2
Initial vessel location n
Turbine 9000
Turbine 9002
Turbine 9004
Turbine 9003
Turbine 9001
A
A
A
A
Urgent vessel transfer with larger fleet size
High priority maintenance action predicted
Routine vessel transfer with lower fleet size
Low priority maintenance action predicted

Fig. 4.21 Depiction of the RL environment, in the form of a grid map – the proposed approach considers transitions between states based on the faults predicted in the simulated wind farm

1. Designing the RL environment for decision making: For development of the proposed DRL based learning algorithm, we utilise OpenAI Gym (Brockman et al., 2016), an open-source library in Python for performing simulation of RL scenarios, and make use of the SafeCab Taxi environment[1] as the basis to modify the environment for our specific problem aimed at vessel transfer planning. In case of the original Taxi environment, the key goal is to navigate a taxi within the grid environment to accomplish minimal time of travel to the intended destination. However, unlike the routing/navigation problem in the original environment, our goal spans towards a vessel dispatch planning problem. The agent in our learning task needs to determine the optimal time to send the vessels alongside the determination of turbines which need urgent maintenance, according to which the vessel is dispatched for picking up or dropping off the maintenance personnel. As wind turbines in a wind farm are generally ordered systematically within a clustered formation, with due mathematical optimisation for their arrangement (Giebel and Hasager, 2016), we utilise a grid map for the purpose of constraining the environment within the offshore turbines cluster boundary as outlined in Figure 4.21. This gives the agent the ability to directly access any grid location from the starting initial point, being subject to the constraint that the vessel reaches the right turbine at the right time, without facing any anomaly/obstacles (leading to stopping or failing).

[1] https://github.com/openai/gym

For this purpose, the five turbines in our grid are assigned different locations (which are designated with different colours). Every incorrect decision/vessel dispatch which is performed is considered an obstacle in the learning process (thereby an anomaly) denoted as A. We assume for this scenario that the vessel is initially present at a random grid location for picking up the maintenance personnel from the offshore port or onshore station. Consider up to 20 different initial locations for the vessel within the environment (a vessel may already be situated in the sea when the training of the algorithm is started), five different possible turbines (destinations for maintenance) and five initial locations for the maintenance personnel, leading to total $20x5x5 = 500$ states. Every state in the environment is denoted as s, with the DQN agent capable of making one of seven different possible discrete (deterministic) actions according to the following:-

- Action 0: Pick up the maintenance personnel from their original (present) location
- Action 1: Move the maintenance vessel to turbine no. 9000
- Action 2: Move the maintenance vessel to the turbine no. 9001
- Action 3: Move the maintenance vessel to the turbine no. 9002
- Action 4: Move the maintenance vessel to the turbine no. 9003
- Action 5: Move the maintenance vessel to the turbine no. 9004
- Action 6: Drop-off the maintenance personnel back to their original (starting) position

2. Initialisation of the environment states: The total size of the learning space for our problem is 500 states $*$ 7 actions = $3,500$. With the pair of state-action values which are denoted as (s,a), the Q-values corresponding to them, represented as $Q(s,a)$ are initialised with random locations, assuming that the turbines in the grid are placed at hypothetical locations and are located at some distance from each other. For example, $[(0,2),(2,1),(2,2),(4,2)]$ denotes a random and hypothetical set of states of the turbine and the corresponding maintenance vessel for initialisation of the agent's learning process. The observations which are utilised for the environment are the maintenance priorities, which were initially mapped in the scale of 0-4 based on the fault type and priority in Section 4.6.1. These observations are randomly permuted within the environment, with the key aim of the algorithm to learn an effective policy for vessel transfer during faults in the turbines, wherein, more severe/critical categories of fault (e.g. in the gearbox of the turbine) are assigned a higher priority, while less severe/crucial faults (such as wind condition alarms) are given lower priority. An additional optimisation constraint is the location of the turbines in the grid, wherein, the turbines with anomalies located closer to the available vessels are assigned a higher priority for the vessel transfer.

3. Developing the reward function for feedback: Given that the learning agent for RL begins initialisation of states with random values, it is integral to provide a measure for evaluating and assessing the agent's performance during every iterative episode throughout the learning process. A reward function is utilised to this end. For our problem, a reward function $R(s,a)$ is considered, as described below:-

- A hypothetical reward of +1 is assigned for every discrete action made by the agent
- Correct movements to the appropriate turbine are assigned a +100 reward
- Any illegal operations executed by the agent towards "Pick up" and "Drop off" (e.g. movement of the vessel outside the grid constraints) are assigned -30 reward
- In cases wherein the agent performs wrong transfers (such as moving the vessel to an incorrect turbine) or causes an interaction of the vessel with an obstacle/anomaly constraint (e.g. the vessel fails unexpectedly in the grid), a reward of -50 is assigned, finally terminating the episode

For every decision which is taken by the learning agent, the goal is to maximise the value of reward outlined in equation 4.24 (Jang et al., 2019).

$$Q(s,a) = r(s,a) + \gamma \max_{a} Q\left(s', a\right) \tag{4.24}$$

Where, γ represents the discount factor, which is a value in the range $[0,1)$ factoring the weights which are assigned by the agent to future rewards on the basis of the reward function's present value.

4. Training the Deep Neural Network in the algorithm: As a part of the final step in the algorithm's learning process, a DNN is utilised and fed with initial values of states which are learnt by the agent, with the network returning different possible Q-values for the discrete values corresponding to actions in the present circumstances of the learning environment, which is outlined in equation 4.25.

$$\begin{aligned} Q\left(s_t, a_t\right) = Q\left(s_t, a_t\right) + \\ \alpha\left[r_{t+1} + \gamma \max_a Q\left(s_{t+1}, a\right) - Q\left(s_t, a_t\right)\right] \end{aligned} \tag{4.25}$$

The neural network is specified with a learning rate of α, with the model being trained for a specific number of episodes in the learning process, which is iteratively repeated until the network converges to a minimal value of loss, as per equation 4.26 (Lin et al., 2019).

$$\text{loss} = \left(\mathrm{r} + \gamma \max_a \hat{Q}\left(s, a'\right) - Q(s, a)\right)^2 \tag{4.26}$$

The network explores various types of possible paths for optimisation during training, through exploration and exploitation by initially making random decisions with the probability (ε) and then greedy decisions with the probability $(1-\varepsilon)$. Although initially, completely random decisions are made by the learning agent, the performance improves continuously generally up to a certain number of episodes, for which a fixed exploration rate is finally set by the agent. During this learning process, the neural network's weights are updated continuously, with iterative computations of approximations for the state-action values.

5. Decision making and prediction of vessel transfer/dispatch: On the basis of the state-action values predicted, the DQN agent makes an optimal decision regarding the specific action to choose out of the seven discrete actions, and ensures maximisation of the network's score (represented by the reward) with the termination of the training schedule, and reaching an optimal value of (ε). Consider the situation when any specific turbine in the wind farm suffers from a fault condition. The DQN agent's key goal here would be to ensure dispatch of the vessel from its initial location to the appropriate turbine in a timely manner, without facing constraints that affect the transfer (such as anomaly/obstacles due to failures or incorrect vessel movements to an incorrect turbine). In our approach, a text-based simulation is generated for the different possible actions which our learning model recommends, which can play an intrinsic role in helping engineers and technicians decide maintenance actions and priorities for vessel transfer planning, alongside the spare parts needed and optimal fleet size for the dispatch.

4.6.3 Experiments

Fault prediction

In the experimentation for fault prediction in the simulated wind farm dataset based on the LDT SCADA data, 233,072 data samples were utilised as obtained after pre-processing. A 70-30% train-test split was utilised. The XGBoost model applied consisted of an ensemble of 100 estimators, with a learning rate of 0.1, early stopping and softprob based objective function (a common variant of softmax activation in XGBoost). In addition, SHAP was utilised to aid in transparency of the model's decisions, through the library's additive feature explainer and probabilistic model output evaluation through assumption of independent features. Using this approach, explainable summary plots were obtained to provide a low-level view of the key features relevant to maintenance planning, alongside additive force plots, which can explain the exact values of features (e.g. exact gearbox oil temperature) contributing to the turbine anomaly. The model predictions were then mapped into the scale of 0 (normal operation) to 4 (urgent maintenance required), and provided as feedback to the DRL algorithm for decision making and optimisation.

Reinforcement learning for decision making and optimisation of vessel transfers

Towards training the DQN algorithm for decision making and maintenance planning, TensorFlow was utilised for implementing the learning model on the Google Compute Engine (with NVIDIA Tesla K80 GPU on the cloud). Baseline models utilised were the Q-learning algorithm, SARSA and Expected-SARSA. The key hyper-parameters utilised for all the models experimented with are enunciated below. All models were trained up to a maximum of 50 episodes (epochs in neural network terminology), as it was observed that training beyond this did not lead to any change in the model's performance at the cost of computational time.

- For the Q-learning algorithm, SARSA and Expected-SARSA, the initial momentum utilised was (ε) of 1, discount factor (γ) of 0.99 and γ was decayed in the learning process to an end value of 0.875.
- For the DQN algorithm, 2 hidden layers were utilised in the neural network, alongside an initial (ε) value of 1 and end (ε) value of 0.001. A learning rate of 0.001 was used, along with a batch size of 32 and discount factor γ specified as 0.99.

4.6.4 Results

Fault prediction

The performance of the XGBoost + SHAP model was evaluated on the test dataset, and it was observed that the model could identify turbine faults given SCADA features with an accuracy of up to 98.8%, and demonstrated an F1 score of 0.99. It was seen that our learning model outperformed (in terms of accuracy) various other algorithms prevalent in the state of the art including Ensemble Bagged Trees (Abdallah et al., 2018a) by up to 0.55% and hybrid DL models such as a combination of LSTM with XGBoost (Chatterjee and Dethlefs, 2020e) discussed earlier in this chapter by up to 2.17%. These results are in line with existing literature, which has shown that for large structured datasets with tabular format of features, XGBoost is highly efficient (Li et al., 2020). We believe that the ability of the XGBoost model to identify complex patterns in the non-linear SCADA features helps to optimally map fault types, while SMOTE enables the model to eliminate bias during prediction making, as otherwise, the model would have mostly predicted the majority class (in our case, normal operation). These fault types predicted at this stage are utilised to infer maintenance action priorities and facilitate decision support which is discussed later in Section 4.6.4.

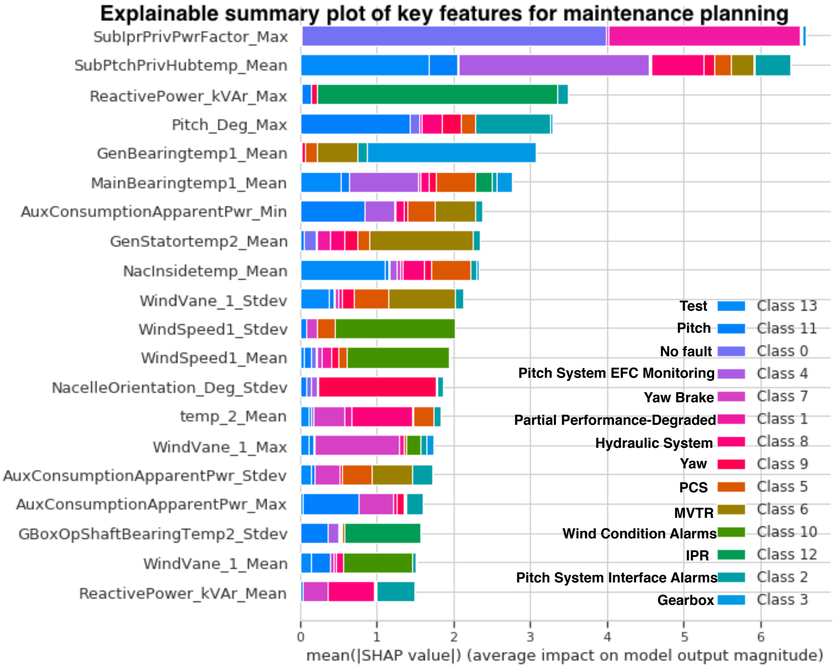


Fig. 4.22 Key features contributing to the anomaly alongside their explainable summary

Explainable summary of predicted faults and their categories By using SHAP, we were able to obtain explainable summary plots for the key features which are most likely to contribute to a specific type of fault. Some interesting results in this aspect are outlined in Figure 4.22. As an example, for an anomaly arising due to wind condition alarms, the key features (with higher values of SHAP scores) are the wind speed *(WindspeedI_Mean)* and wind direction *(WindVane_1_Mean)*, which is feasible and highly relevant, as any change in the normal values of wind speed/direction are a common cause of generally prevalent SCADA alarms (Qiu et al., 2012). Another example is outlined for the Pitch System Interface Alarms, which are attributed to an inconsistency in the pitch angle alignment values *(Pitch_Deg_Mean)*. This inference is relevant and in line with existing literature (Elosegui et al., 2018), as any abnormal deviations in the turbine's pitch angle can directly cause an anomaly in the pitch system.

Low-level analysis of the anomaly through additive force plots To generate further insights into the SCADA features which contribute to the anomaly in the turbine, additive force plots (Chen et al., 2020a) were developed in our study using SHAP. Additive force plots are extremely helpful to understand the feature values in the dataset at runtime (during prediction making), and can outline how various features in the SCADA data vary and push the expected model's output (base value under conditions of normal operation) towards the predictions which the model actually makes during an anomaly. Figure 4.23 provides an interesting example for an additive force plot generated during an anomaly in the turbine's gearbox, wherein, an inconsistency in the gearbox's temperature values directly causes performance degradation of the drive train as well as the generator, which affects the overall power output performance and operational behaviour of the turbine. The analysis and inference obtained were also found to be in line with expert judgements based on domain knowledge, and existing literature in this area (Jin et al., 2018)..

Performance evaluation of the RL algorithms

For evaluating the Deep Q-learning technique's peformance in our learning task and comparing it with the other baseline models utilised, the metric of average reward (score) obtained was used during our analysis, along with the computation time of the algorithm. Here, a higher reward would represent a better performance of the RL agent in updating its policy and learning maintenance action priorities on the basis of the fault types and their severity.

The evaluation goal here is to consider the average total reward achieved by each model. The highest end reward as depicted in Figure 4.24 is obtained by the DQN, which shows that

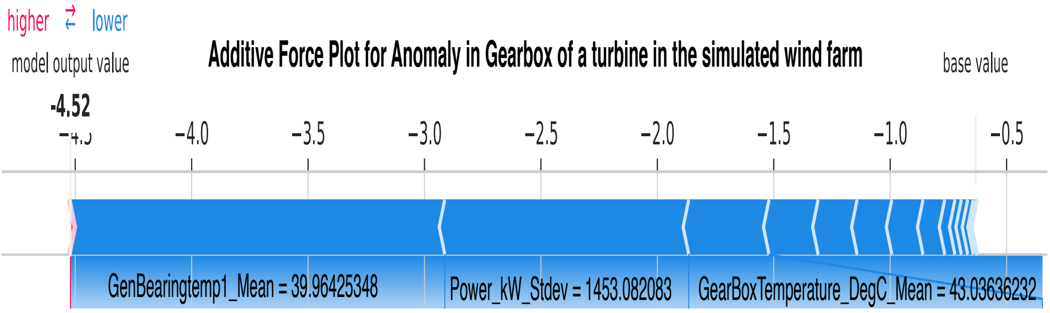


Fig. 4.23 Additive force plot analysis during a predicted anomaly in the turbine's gearbox – the outlined features contribute to the anomaly due to deviation from their expected (base values) during normal operation

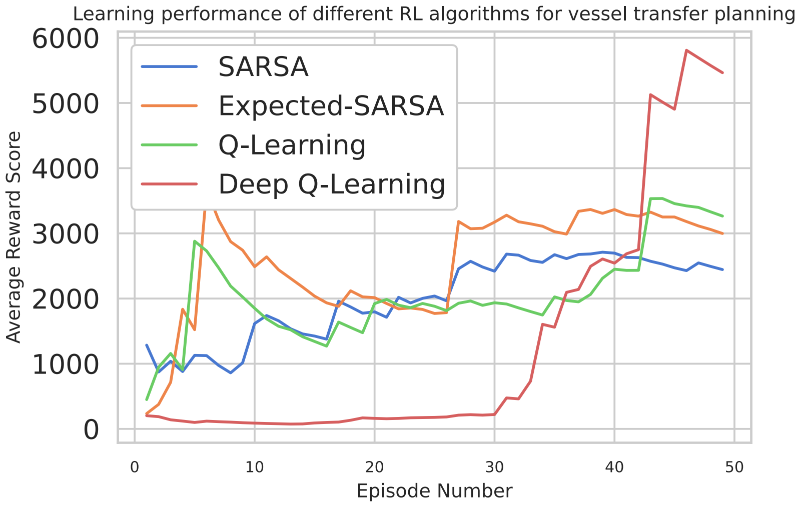


Fig. 4.24 Performance comparison of various RL algorithms utilised, wherein, a higher reward value would represent a better model performance.

Table 4.7 Performance evaluation with key metrics for the different RL algorithms utilised over 50 episodes. The best performing algorithm is highlighted with bold face.

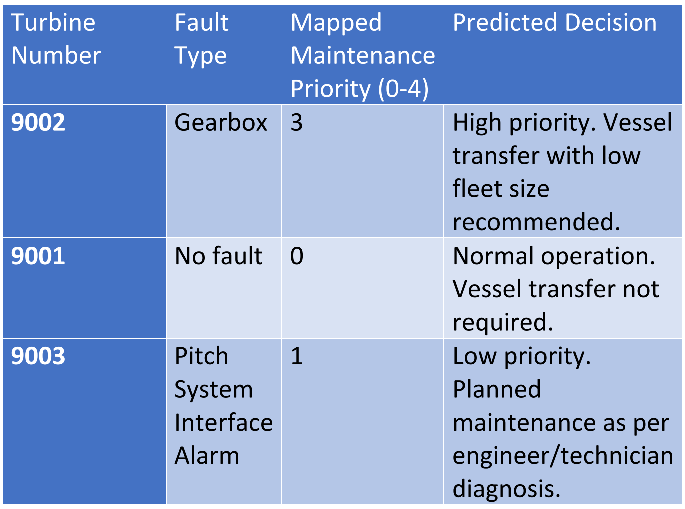
RL algorithm	**Total average reward**	**Total Computation time (Approx. On GPU)**
Q-learning	3264.14	1.3 min
SARSA	2444.32	**1.19 min**
Expected-SARSA	2997.82	1.5 min
Deep Q-learning	**5464.60**	2.2 min

it is the best performing model. While it can be seen that the reward for the DQN algorithm slightly decreases from its achieved peak value near completion of the training episodes, this is due to priority replay (Liu and Zou, 2018), which assigns a higher probability of sampling to the environment's states outlining prioritised experience. The Q-learning algorithm is the second best performing technique, although it scores below the DQN by a large margin. The performance metrics and key measures for all four algorithms are summarised in Table 4.7. Note that in terms of computation time, the SARSA algorithm achieves the best performance, while DQN takes the longest time for computations, primarily owing to the utilisation of the neural network in the DQN for updating the agent's policy, with better performance coming at the cost of increased computation time.

Decision making for vessel transfer planning

With the primary goal of developing an intelligent decision support system for offshore vessel transfer planning in this chapter, which can help in determining vessel dispatch schedules as well as optimal fleet size (either larger or smaller) based on fault type and severity, a number of hypothetical examples are outlined in Table 4.8. Here, the fault type is mapped to the corresponding maintenance priority (based on its severity), and we demonstrate the feasibility of decision making by turbine engineers and technicians alongside wind farm operators using the proposed DQN algorithm.

Table 4.8 Some hypothetical scenarios outlining the decision making process in the simulated wind farm environment.



Turbine Number	Fault Type	Mapped Maintenance Priority (0-4)	Predicted Decision
9002	Gearbox	3	High priority. Vessel transfer with low fleet size recommended.
9001	No fault	0	Normal operation. Vessel transfer not required.
9003	Pitch System Interface Alarm	1	Low priority. Planned maintenance as per engineer/technician diagnosis.

The XGBoost model here contributes to providing accurate predictions of faults in the turbine, while the DQN optimally leverages the maintenance priorities (for our case, on a scale of 0-4) to maximise the network's rewards (score), and thereby helps in decision making.

The code for the technique and the video outlining the generated text-based simulation are made openly available at [2].

4.7 Conclusions

In this chapter, we have proposed a novel application of DL techniques for the wind industry, in use-cases spanning anomaly prediction for an offshore wind turbine, transfer learning for predicting faults in new turbines without labelled historical training data, and planning for offshore vessel transfers through the blessings of DRL. Our study shows that DL algorithms, such as LSTMs show immense promise in performing anomaly prediction given SCADA data, but suffer from the drawback of lack in transparency, which can be overcome through utilisation of more traditional ML algorithms such as XGBoost. This chapter demonstrates that Deep Q-learning is a promising and highly feasible technique which can be utilised by the wind industry, and using other techniques in conjunction (such as SHAP based on game theory for explainable predictions) can help engineers understand not only the fault which occurs, but also its type and context. We believe that this can play an intrinsic role in determination of offshore vessel transfer plans, as timely decisions on averting/fixing faults can be made beforehand while minimising the costs (anomalies, obstacles and failures). Additionally, more severe fault types can be given higher priority of maintenance, while unnecessary transfers can be delayed and minimised or smaller fleet sizes can be utilised for such purposes. Finally, an important conclusion which this chapter makes is on the nature of DL for the wind industry – while it can indeed be a highly promising area, but using DL algorithms standalone has several drawbacks (such as lack in transparency, and in some cases with smaller datasets – a lower model performance). This motivates us towards developing hybrid models for decision making rather than using DL algorithms standalone, tailoring them for domain-specific issues which the wind industry presently faces. In the upcoming chapters, we would build upon the promise of DL, and explore further variations in achieving more fine-grained decisions in AI models. The next chapter (Chapter 5) will focus on identifying causal relationships in SCADA data for discovery of novel associations and hidden predictive variables, which can help build further trust in the prediction process of DL models.

[2] https://github.com/joyjitchatterjee/DeepRLTurbineVesselTransfer

Chapter 5

Temporal Causal Inference in SCADA Data for Explainable AI

> I believe that whatever we do or live for has its causality; it is good, however, that we cannot see through to it.
>
> Albert Einstein

This chapter is based on the work previously published by the author as an article in the *Journal of Physics: Conf. Series* (Chatterjee and Dethlefs, 2020c) and at the *KDD Fragile Earth* workshop (Chatterjee and Dethlefs, 2020a).

5.1 Introduction

While the wind industry has experienced growing interest in utilising SCADA data for data-driven decision making (Zhao et al., 2017), such data entails a major challenge owing to its complex and non-linear nature – multiple features present in SCADA data can share a common cause (*confounders*) leading to a predicted fault in the turbine, as well as consist of several hidden relationships. With the application of AI models (especially deep learners) to SCADA data for decision making, the black-box nature of the models hampers transparency in the predicted decisions, limiting their utilisation for reliable and autonomous decision support in the industry. To achieve reliable decision making and predictions, discovering hidden knowledge within the data in the form of causal associations is integral, and can help design data-driven systems based on effective policies to either prevent or produce the desired outcomes (Kleinberg, 2013).

According to Pearl (2019), there is a significant role of actions, interventions and decisions made by predictive models, wherein the utilisation of causal reasoning can be pivotal in achieving human-level intelligence in generally opaque predictions. Causal reasoning has seen successful application in various domains, such as utilisation for medical treatments (Hazlett, 2018), understanding the context of forecasts made by AI models in stock markets (Zhang et al., 2014) etc. In a notable vein of research in this domain, Wang and Blei (2019) have previously utilised probabilistic modelling for performing causal inference through an algorithm called the *Deconfounder*, which has been applied to real-world datasets towards achieving reliable decision-making. However, such algorithms only model the identified effects on an outcome based on univariate causes in the dataset, and ignore such situations in which multiple features can be present and contribute towards a specific outcome. This is exactly the challenge with SCADA datasets for wind turbines.

In this chapter, we propose the application of temporal causal inference for the wind industry using SCADA data towards accurately modelling relationships (especially hidden associations) within the SCADA features as well as providing an estimate of the temporal delay between the cause (the contributing SCADA features) and the effect (a specific anomaly in the operation of the turbine). To this end, we utilise Convolutional Neural Networks (CNN) with attention mechanism for discovering such hidden associations through temporal causal inference (Nauta et al., 2019). The proposed model is applied to time-series of SCADA features and historical alarm logs in the turbine's operation and we provide a comparison of the model's performance with other state of the art methods (Hazlett, 2018). The causal inference technique proposed in this chapter can support wind turbine O&M by discovering novel insights in relationships between SCADA features. Note that not all of the identified relationships through causal inference in SCADA data may be obvious and there can be presence of some features (which might have potentially been disregarded as insignificant/noise), which can contribute integrally to the learning process of the AI algorithms. We demonstrate the role of causal reasoning in enhancing reliable decision support for wind turbines, by helping make black-box neural network models more transparent, interpretable and thereby robust. The primary objectives of this chapter are outlined below:-

- **Causal inference:** To learn a temporal causal graph from SCADA data based on past values of the observed time-series and historical alarm logs.
- **Attention interpretation:** To determine the causal relationships between features during faults in turbine sub-components using a DL-based attention mechanism.

- **Delay identification:** To identify the delay between cause and effect, i.e. the time-steps lag at which a change in one feature affects another feature.

The chapter is organised as follows: Section 5.2 demonstrates the application of traditional statistical techniques for causal inference and discusses their drawbacks. Section 5.3 introduces the concept of DL for temporal causal inference and discusses the proposed learning model. Experiments are discussed in Section 5.4 by applying DL for causal inference in real-world wind turbine SCADA data. Results are shown in Section 5.5 and detailed analysis of causal reasoning across different fault types in turbines are discussed. Finally, Section 5.6 concludes the chapter, providing the background for further fine-grained explainations in the form of natural language in forthcoming chapters.

5.2 Traditional statistical methods for causal inference

Before diving into DL for causal inference, it is integral to demonstrate the challenges and drawbacks posed by traditional statistical causal reasoning algorithms. To this end, we utilise a commonly applied causal inference technique based on conventional statistical computations – the Bayesian Structural Time-Series (BSTS) (Kay H. Brodersen et al., 2015) [1] for our SCADA data. The BSTS is a state-space model for time-series data based on a Markov chain Monte Carlo algorithm for posterior inference, which consists of a regression component that can provide conterfactual predictions for causal attribution. The model is particularly adept at accommodating multiple types of variation in datasets, including local trends, seasonality and time-varying aspects of contemporaneous covariates (Brodersen et al., 2015) which are generally prevalent in SCADA datasets. We use the BSTS as our baseline model in this chapter – see Brodersen et al. (2015) and Kay H. Brodersen et al. (2015) for more details regarding BSTS.

We wanted to explore the role of causal inference in identifying the *hidden relationships* between the causing/intervening variable on the response/outcome variable. As an example, we demonstrate a relationship which is interpreted as random and spurious by traditional methods, but is identified as a hidden confounder by the DL-based causal inference model in Section 5.3.

Figure 5.1 depicts the statistical effects of variation in the *turbine's nacelle angle (intervention)* on the *generator stator temperature (response)* during an actual anomaly in the Yaw system. Here, the pre-intervention period shows the predicted time-series of the response

[1] https://github.com/dafiti/causalimpact

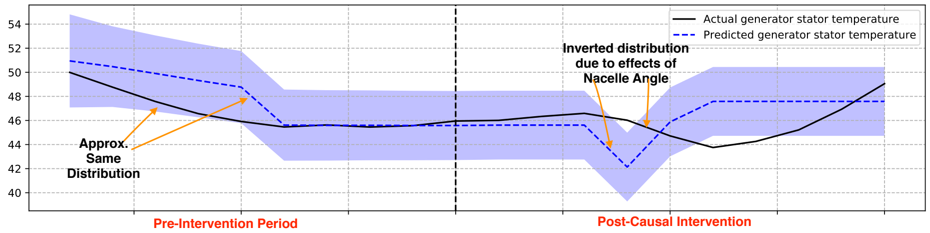


Fig. 5.1 Causal effects (non-obvious) of nacelle angle (X) on generator stator temperature (y) during anomaly in Yaw system using BSTS Approach (Statistically Insignificant). Probability of random occurrence p = 24.88% and 95% interval of percentage change in X and y = [-3.44%, 1.73%].

variable using the BSTS model which we would have expected without the causal event occurring, while the post-intervention period highlights how the variable changes as a result of the intervention caused by nacelle angle. Though we see that the intervention seems to have a negative effect on the predicted response variable (as the actual time-series distribution is approximately inverted after the intervention caused by nacelle angle), this is found to be very minor (a -3.44% change in nacelle angle causes a 1.73% change in generator stator temperature in the 95% confidence interval). We found that the BSTS technique discovered little significance in how these metrics are causally related, and interpreted the associations as statistically insignificant and possibly spurious. To confirm this inference, we calculated the auto-correlation for this metric as shown in Figure 5.2. The auto-correlation components are clearly non-zero for the variations of the lagged time-series, outlining that the variable is in fact not random, but has a temporal nature, motivating us to look for alternate causal inference models.

5.3 Deep learning for causal inference

5.3.1 Learning Task

Consider a multivariate time-series of various features in the SCADA dataset which are represented as $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\} \in \mathbb{R}^{N \times L}$, wherein, N denotes the total number of available SCADA features in the dataset and L represents the total number of samples (observations) for each SCADA time-series feature. In this scenario, every ith sample of individual SCADA features (wherein, $i \in N$) is denoted as per the notation $\mathrm{X}_1 \ldots \mathrm{X}_L$. The learning task for our problem in this chapter lies in identifying the casual relationships between these N SCADA features (wherein, it is also possible for individual features to share

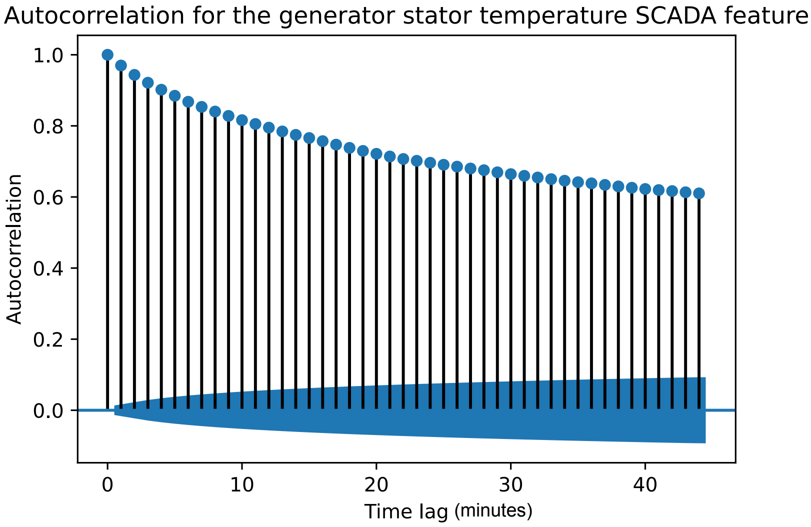


Fig. 5.2 Auto-correlation for the generator stator temperature time-series. The components are significantly non-zero, signifying non-random nature of the metric.

cause-effect relations with multiple other features), the time delay at which the relationship gains prominence (cause and effect temporal delay), and finally construct an intuitive and visual representation of these relationships in the form of a temporal causal graph.

As this problem pertains to identifying the directionality between the cause-effect relationships in SCADA features during different types of faults, a directed causal graph (Lewis and Kuerbis, 2016; Nauta et al., 2019) represented as G_{SCADA} = (V, E) is to be constructed consisting of a collection of vertices and edges with associated relationships. Here, each vertex $v_i \in V$ would denote an individual SCADA feature, while the edge connections represented as $e_{a,b} \in E$ would signify a causal transition (association) from vertex v_a to v_b. Further, we refer to the notation utilised by Nauta et al. (2019) for datasets with multivariate time-series observations, according to which our additional goal reflects determining the temporal delay $d\left(e_{i,j}\right)$ between the cause and effect's occurrence i.e. the number of time-steps in the dataset (observations or samples, wherein, each of our SCADA features is generally at 10-minute interval) after which one SCADA feature would causally affect another feature leading to a fault/anomaly in the turbine's operation. Figure 5.3 represents the structure of the directed causal graph which we intend to construct with the SCADA features and fault records in this chapter.

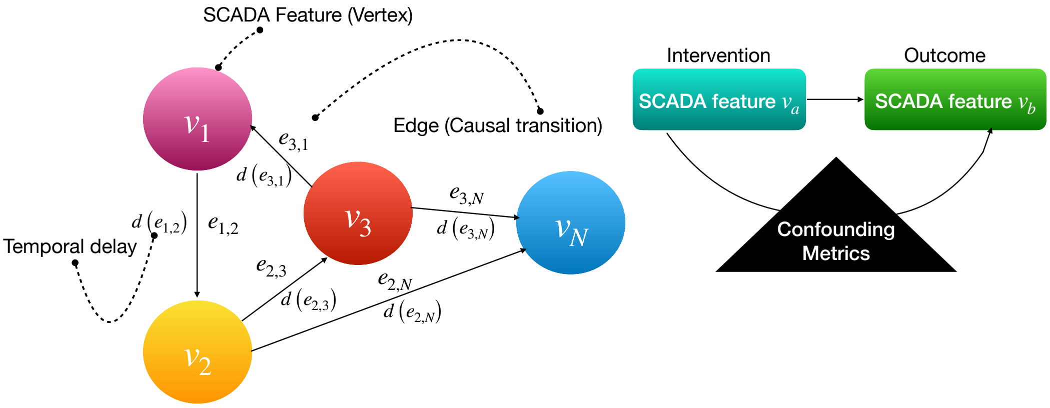


Fig. 5.3 General structure of directed causal graph for inference with SCADA data – we aim to construct directed relations (edges) describing causal transitions between SCADA features to identify confounding metrics

5.3.2 Learning Model

For our learning task, we propose the utilisation of a specialised DL model for temporal causal inference, namely the Attention-based Convolutional Neural Network (CNN), which was first proposed towards successful application in causal inference for financial and neuroscientific domains by Nauta et al. (2019). Our key focus in developing the learning model is to make it suitable for learning with continuous sequences of SCADA time-series features alongside the historical logs of alarms which have occurred in the turbine. For this purpose, we modify the original model's architecture towards identifying hidden confounders from SCADA data in an operational turbine. The various components of this learning model are described below:-

1. **Predicting the SCADA features' time-series:** The initial step in our proposed causal inference model pertains to predicting the values of different time-series of SCADA features in the dataset. As described before, in our context, given a sequence of N individual SCADA features denoted in the range $\mathrm{X}_1 \ldots \mathrm{X}_N$, wherein, each feature contains L different samples (observations signifying total length of the time-series), the goal here is to predict the future values $\hat{X}_i$ of each SCADA time-series feature on the basis of its present and past values. We utilise a CNN architecture for this purpose.

   CNNs are a specialised type of feed-forward neural network and have seen immense success in applications for computer vision (Alom et al., 2019), and have more recently demonstrated the ability to handle datasets of sequential nature such as time-series (Du et al., 2018). There are a variety of convolutional layers in a typical CNN model with sliding kernels, which gives them the ability to discover novel patterns in the time-series features as well as predict the target time-series' future values. Note that a kernel here refers to a filter (in mathematical terminology, basically a matrix with multiple associated weights) which is computed by performing a convolution operation (dot product) of the filter weights W at the present time-step with the numeric value of the input time-series. Consider that for each SCADA feature X_i in the dataset, the future j values of their time-series is to be predicted. This signifies a total of $L - j$ samples to be used as input for training the CNN, and utilisation of a kernel size S (generally determined experimentally while performing hyper-parameter optimisation). This operation is computed as per equation 5.1.

$$\mathscr{W} \odot \left[X_i^{j-S+1}, X_i^{j-S+2} \ldots, X_i^{j-1}, X_i^{j}\right] \tag{5.1}$$

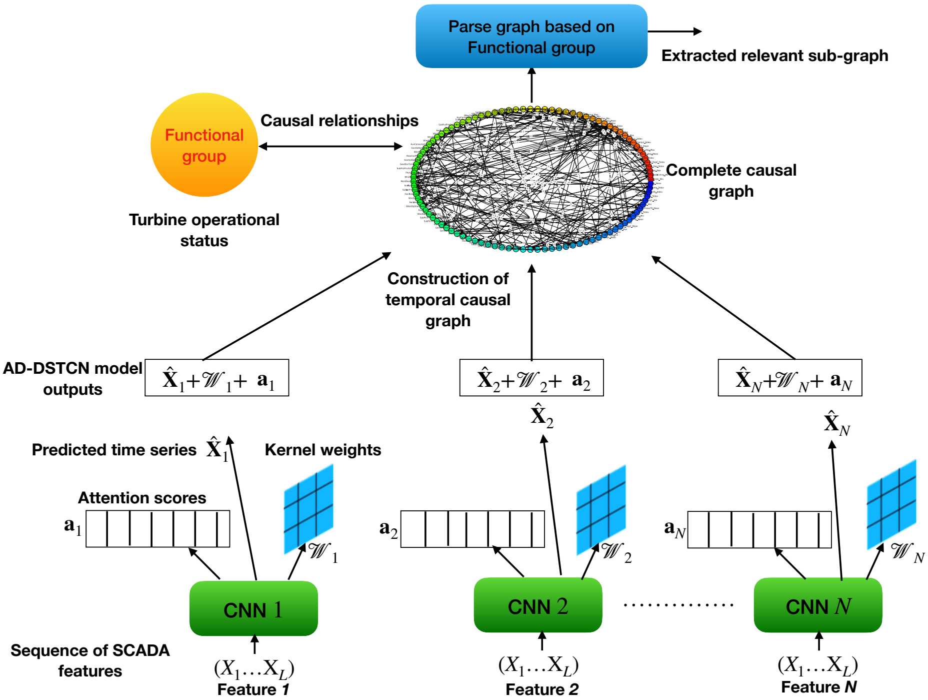


Fig. 5.4 The proposed CNN with attention mechanism for performing temporal causal inference with SCADA data. The model outputs the kernel weights and predicted time-series of SCADA features, which are finally used to construct the causal graphs during different cases of faults.

This process pertains to predicting the value of an individual (univariate) SCADA time-series feature. However, given that we have a collection of multiple SCADA features (multivariate time-series), we utilise N independent CNN architectures, which are represented as $\mathscr{C}\mathscr{N}\mathscr{N}_1 \ldots \mathscr{C}\mathscr{N}\mathscr{N}_N$. Here, the individual CNNs $\mathscr{N}_j$ predict the time-series $\hat{X}_j$ corresponding to its input SCADA data (i.e. the first SCADA feature is predicted by $\mathscr{CNN}_1$, the second SCADA feature is predicted by another $\mathscr{CNN}_2$ etc.). Besides the predictions of the time-series features, note that the CNNs also output the kernel weights $\mathscr{W}_j$. This phenomenon is depicted in Figure 5.4.

Conventionally, CNNs with sliding kernels utilise an equal number of time-steps (termed receptive field) as the kernel dimension or are specified by the user. However, to perform temporal causal inference, it is integral for the network's receptive field to be larger than the temporal delay between the identified cause-effect relationships. For this purpose, a dilation mechanism is incorporated into the CNN to increase the dimensions of the model's receptive field, and facilitate the identification of causal associations even under situations wherein the temporal delay is large between multiple causes.

The dilation rate here is represented by K, wherein, a value of $K = 1$ would signify the application of a conventional CNN, $K = 2$ would skip 1 time-step (observation) in the time-series, $K = 3$ would skip two time-steps and $K = n$ would eventually skip $n - 1$ time-steps. This mechanism is effective in modelling prediction cases for SCADA features when temporal causal delay is large, and additionally makes the computational process less expensive in performing updates of the sliding kernel's weights over time (He and Sun, 2015). Given a set of different features from the original dataset (exogenous SCADA time-series), the goal here is to predict the target values of the time-series, ensuring that the loss L achieved across the ground truth values ($\mathbf{X}_j$) and the predictions ($\hat{\mathbf{X}}_j$) is minimised. Figure 5.5 shows an example illustration of some of our exogenous time-series based features used in constructing a complete causal graph.

Note that while CNNs can perform the predictions of time-series data with high accuracy (Du et al., 2018), similar to other DL models, they fail to provide reasoning of causal associations utilised by the model in making predictions. To tackle this issue, we further incorporate an attention mechanism which is discussed next.

2. **Attention mechanism for transparency in learning:** As an integral step for causal inference, an *attention mechanism* is utilised and incorporated on top of the conventional CNN architecture to facilitate discovery of specific features in the SCADA dataset which likely contribute to the target predictions and are causally related. The essence of the attention mechanism lies in identifying the key features which the CNN model focuses on during its learning process. For this purpose, we develop our mechanism based on Tang et al. (2016) and further integrate it with the causal inference

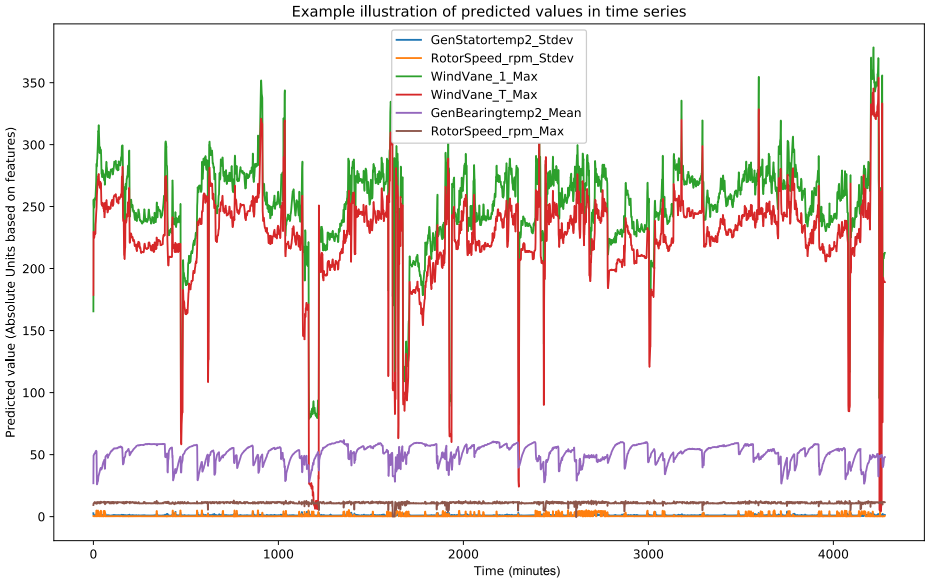


Fig. 5.5 Example depiction of some SCADA features utilised to construct the temporal causal graph – the values of each feature are predicted by the corresponding independent CNN models

framework proposed by Nauta et al. (2019). Consider N individual features in our SCADA dataset across which temporal causal inference is to be performed.

In mathematical terms, the attention vector here is a row vector possessing the same dimensions as the number of SCADA elements utilised (of dimensions $1 \times N$) and for each distinct time-step L in the SCADA time-series features (i.e. belonging to time-step 1 to time-step L), attention scores are computed by matrix multiplication of the input feature $\hat{\mathbf{X}}_j$ with the updated version of kernel weights in equation 5.1. This leads us to obtain individual attention scores for each SCADA feature and is described in equation 5.2, wherein j denotes the number of time-steps possessed by each SCADA feature.

$$\mathbf{a_j} = \left[a_{1,j}, a_{2,j}, \ldots, a_{N,j}\right] \tag{5.2}$$

These vectors are finally utilised to generate a square matrix with attention scores, by concatenating the individual attention row vectors in accordance with equation 5.3. Considering the N SCADA features in our dataset, this phenomenon can be simply represented by assuming an $N \times N$ square matrix with attention scores (similar to adjacency matrices), wherein, matrix element $a_{i,j}$ represents the attention scores for

SCADA feature X_i to be a likely cause of feature X_j. The diagonal elements of this matrix (in cases where $i = j$) signify self-causation (Nauta et al., 2019).

$$\mathbf{A_{i,j}} = \begin{bmatrix} \mathbf{a}_{1,1} & \mathbf{a}_{1,2}\cdots & \mathbf{a}_{1,N} \\ \mathbf{a}_{2,1} & \mathbf{a}_{2,2}\cdots & \mathbf{a}_{2,N} \\ \cdots & & \\ \mathbf{a}_{N,1} & \mathbf{a}_{N,2}\cdots & \mathbf{a}_{N,N} \end{bmatrix} \tag{5.3}$$

The attention scores above outline a measure similar to conditional probability in statistics (Jia, 2019), but are a function of the CNNs' hidden states and show the relative importance of a feature towards obtaining a specific output. Thereby, when the model is performing time-series prediction of a SCADA feature, the relative importance of other features is also obtained, with higher scores signifying a stronger (more likely) cause-effect relationship. During experimentation, the user can manually specify the number of parameters to utilise from the attention score matrix, which is represented by the top k values. Alongside the kernel weights obtained in step (1), the attention scores are utilised in conjunction to finally obtain the temporal causal graph as described next in step (3).

3. **Developing the temporal causal graph based on kernel weights and attention scores:** By integrating the CNN with dilation mechanism discussed above in step (1) with the attention mechanism for transparency in (2), we obtain the final causal inference model, namely Attention-based Dilated Depthwise Separable Temporal Convolutional Network (AD-DSTCN) (Nauta et al., 2019). Till this stage, we had utilised N independent CNN models for time-series prediction of the N different SCADA features, which can also output the kernel weights and attention scores. These parameters are utilised in performing temporal causal inference over the SCADA features and construction of the complete causal graph, as depicted in Figure 5.4. However, as development of the causal graph based on all parameters obtained can likely lead to obtaining too many relationships (many of which may not make sense and can possess low scores), we utilise a *significance measure* $s(c,e)$ for this purpose to compute the causal effects of cause c on effect e (Huang and Kleinberg, 2015).

   The significance measure parameter (which belongs to range $[0,1]$) determines the instances wherein the model's increase in loss is significant enough in between the causing feature (also called the intervened time-series which experiences sudden changes/-variations in time) and the outcome (which is causally affected) due to changes in the intervened time-series feature, to finally label the potential causes (after validation) as true (Nauta et al., 2019). Note that a higher significance measure puts lesser constraints on the model towards identifying hidden confounders. With the complete causal graph constructed, the relevant sub-graphs can simply be extracted based on the different types of faults in the turbine (across multiple functional groups). Whenever a fault occurs, if a SCADA feature is contributing to it, then any changes in its value would lead to changes in the confounding features, and such instances of association would be included in the graph. Conversely, for features which are not significant, any variations in the time-series would not cause variations in the confounding features, and such instances of association would be excluded from the graph.

The process described above is enunciated in the form of pseudo-code in Algorithm 3.

Algorithm 3: Pseudo-code for the proposed temporal causal inference model

```
Input: Multivariate time-series of 102 SCADA features
Output: Temporal causal relationships
Data: SCADA data $X = \{x_1, \cdots, x_n\}, x_i \in \mathscr{X}$ and historical categorical labels for
      faults $y$
/* Feed individual SCADA features into independent CNNs                    */
for $X_i$ in $X$ do
    $cnn_i\_input \leftarrow X_i$
/* Training of the CNNs for causal inference                               */
for $cnn\_model$ in $cnn_i$ do
    $cnn\_model \leftarrow train(\mathrm{cnn}_i\_input, y_i)$
/* Retrieve attention scores, kernel weights and predict corresponding
   time-series towards causal inference                                     */
$att\_score, kernel\_weights, predicted\_timeseries \leftarrow \mathrm{cnn\_model}(X_i)$
/* Calculate final AD-DSTCN model outputs                                   */
$causal\_factors \leftarrow sum(\mathrm{att\_score}, \mathrm{kernel\_weights})$
/* Construct and parse temporal causal graph based on fault types           */
$graph\_causality \leftarrow parse(\mathrm{causal\_factors}, FunctionalGroup)$
```

5.4 Experiments

We utilised 21,392 samples from the LDT SCADA data with 102 features and 14 functional groups of faults (as described in Chapter 3), for the purpose of temporal causal inference.

The AD-DSTCN learning model for causal inference is implemented using PyTorch (Paszke et al., 2019). A learning rate of 0.01 is used for training the model, with kernel dimensions 2x2, dilation coefficient of 2, Adam optimiser for learning and a train-test split of 80-20%. The above parameters were obtained after extensive optimisation of the hyper-parameters and performing empirical tuning. As discussed before, we utilise the predicted features from the trained CNN model for discovering the cause-effect associations in SCADA time-series parameters pertaining to various types of faults in turbine sub-components. To this end, the top 20 attention weights are utilised in development of the final temporal causal graph. Given that our goal here is to discover novel insights and relationships (rather than obvious confounders), we choose the significance measure based on existing literature (Nauta et al., 2019; Wang and Blei, 2019). Previously, these studies have demonstrated that a significance measure of $s = 0.8$ is ideal for achieving good results, and can help suitably bridge the gap between the number of hidden counfounders identified and their relevance to the fault types.

The model is evaluated in accordance with the Mean Absolute Scaled Error (MASE) obtained during time-series prediction of SCADA features as well as average standard deviation, and is compared with a state of the art baseline model, the Deconfounder proposed by Wang and Blei (2019). The Deconfounder is based on statistical learning for causal inference and utilises probability factor modelling towards determining causal effects across different groups of the population through controlled study. Finally, the temporal causal graph is obtained using the kernel weights and attention scores. Relevant sub-graphs for instances of different types of faults are extracted from this complete causal graph and used to interpret the identified confounders. This is discussed in more detail in Section 5.5.

5.5 Results

Table 5.1 describes the experimental results for causal inference. The proposed AD-DSTCN model with a single hidden layer achieved a MASE of 1.066 and a standard deviation of 2.948, demonstrating the best performance. In comparison to this, the Deconfounder (Wang and Blei, 2019) baseline model demonstrated a MASE of 3.901 – which is 72.67% worse than the proposed causal inference model.

From the complete causal graphs, based on the multiple types of faults (in the functional groups), the subsets of the graphs were extracted. In accordance with this, we utilised the discovered causal associations to identify hidden insights during anomalies in multiple turbine sub-components. Further, the AD-DSTCN model is capable of identifying the temporal delay in the causing feature leading to the identified effect in the SCADA time-series (wherein, 1

Table 5.1 AD-DSTCN Model parameters and their performance evaluation during temporal causal inference – the best performing model is shown in bold face

Layers in Depthwise Convolution	Optimiser	Epochs	MASE	Std. Deviation
1	Adam	500	3.292	4.731
		1000	**1.066**	**2.948**
		2000	1.212	2.985
	RMSprop	1000	2.493	3.829
2	Adam	1000	2.609	2.356

time-step is equivalent to generally a 10-minute interval in the SCADA data). It is essential to mention here that unlike other forms of data such as time-series/text/images and video signals etc., the temporal causal graphs are not prevalent in Euclidean space. Additionally, these graphs do not have a fixed structure, which generally makes both, the development of the relationships and their quantitative analysis more challenging. However, the identified relationships can be analysed by turbine engineers to provide better understanding of the decisions made by the AI learning models. The proposed DL approach for causal inference can help discover novel insights in SCADA data during different types of faults, and shed light on certain key associations in SCADA features which may have otherwise been potentially disregarded as noisy and spurious.

Given that we did not have the ground truth for identified hidden confounders, it was not possible to quantitatively evaluate these relationships. However, to overcome this challenge to the best possible, we performed a qualitative evaluation of the identified hidden conofunders in the SCADA data across multiple types of faults (specifically in the 14 different functional groups). Note that there are some relationships which are obvious (e.g. the causal association in active power mean value to the active power minimum value), but may not be significant in terms of their relevance to a given fault's context. For the purpose of qualitatively evaluating the temporal causal graphs, we thereby only consider those associations as valid which are relevant to the fault type and in line with its context.

Note that in some cases, the causal reasoning model was not able to identify any associations which were relevant to the context of the fault. The percentage evaluation is performed over a random (but representative) portion of the test SCADA data, and represents the likelihood of a causal association to be relevant to the fault type based on domain understanding. It is imperative to mention that this reflection may differ from the process utilised by the machine in interpreting such associations, and should thereby only be considered a reflection of human domain understanding. The percentage relevance of causal associations (wherever identified) under different types of faults (based on functional groups) is summarised in Table

Table 5.2 Summary of relevance based on qualitative evaluation of hidden confounders. The cases with $\geq 60\%$ relevance scores are outlined in bold face.

Functional Group	Hidden Conf. discovered	Avg. % Relevance
No Fault	✓	**86.95%**
Partial Performance-Degraded	✓	45.45%
Pitch System Interface Alarms	✓	**63.15%**
Gearbox	✓	33%
Pitch System EFC Monitoring	✓	**60%**
PCS	✓	54.54%
MVTR	×	N/A
Yaw Brake	✓	**60%**
Hydraulic System	✓	58%
Yaw	✓	44.44%
Wind Condition Alarms	×	N/A
Pitch	✓	**66.66%**
IPR	×	N/A
Test	✓	28.57%

5.2. [2] Note that temporal causal graphs are evaluated based on a significance measure of 80%, which was found to be most reasonable in terms of providing inference of a sufficient number of hidden associations, as well as being relevant to the fault's context.

As outlined in Table 5.2, our learning model identified temporal causal relations being most relevant for the turbine's normal operation (when there is no fault), followed by the functional groups for pitch system, the pitch interface alarms, pitch system EFC monitoring and yaw brake, with all cases demonstrating greater than 60% relevance to the fault's context. The learning model failed to identify hidden confounders for Wind Condition Alarms and Moisture Vapour Transmission Rate (MVTR), which can likely be attributed to lack of muliple confounding parameters (other than the obvious wind speed in case of the former, and vapour transmission related parameters for the latter). In some cases such as Test Rig and Gearbox, the model performed poorly (with relevance scores as low as 28.57%), likely due to the complexity in the nature of such faults (and errors in data generated by the test rig during anomaly).

Below, we discuss some cases of hidden temporal causal associations identified by the learning model which were interesting. Multiple cases for additional causal graphs during different types of faults besides those discussed are available on GitHub[3]. Note that our discussion is focused on relevant hidden confounders identified, which can be qualitatively

[2] The interested reader is referred to the Platform for Operational Data (https://pod.ore.catapult.org.uk) for more details on the Functional Group acronyms.

[3] Multiple cases of temporal causal graph with SCADA data: https://github.com/joyjitchatterjee/TurbineSCADACausality/

interpreted and understood by a human rather than all confounders which the AI model looks at during its predictive process for which the evaluation is difficult to establish. We attribute the irrelevant parameters to specific cases, which can further be investigated by turbine engineers and technicians in better understanding *why* the AI learning model makes use of the causal associations between the SCADA features in some instances of faults.

Note that for the described cases of temporal causal graphs, the relationships should be interpreted as per the following notation: any change in feature Y (outcome of causation) occurs following a variation (intervention) in feature X (cause), wherein, the relationships inferred are $Y->X$. We utilised this notation owing to its simplicity and suitability for describing SCADA features. The temporal delay $d(y,x)$ represents that the outcome feature experiences a variation in its time-series d time-steps after a change occurs in the causing (intervening) feature. For instance, a temporal causal association from rotor speed (Y) to the pitch angle feature (X) (denoted by $Y->X$) occurring at a delay of 1 time-step would signify that the rotor speed time-series is causally affected by a variation in the pitch angle, and this relationship takes prominence after 1 time-step (10-minute interval) after the pitch angle varies. Note that the minimum time-delay ahead of which these causal relations can be predicted is 10 minutes. While our causal inference model is able to identify the temporal delays for the SCADA features, it cannot provide the average time after which a fault in one sub-system affects another sub-system.

Case 1: Identified hidden confounders for normal operation of turbine, with a significance measure of 0.80 A temporal causal graph outlining the relations during an instance of normal turbine operation is depicted in Figure 5.6. It can be seen that most of the relationships identified are as expected, which includes associations such as turbine rotor speed being related to pitch angle, nacelle angle being related to rotor speed as well as wind direction, active power being related to the grid voltage etc., all possessing an instantaneous temporal delay *(0 time-steps)*. These relations are reasonably identified and are vital for turbine operation, especially during the phases of power limitation and optimisation control, making them integral for ensuring optimal performance (Balijepalli et al., 2018), as turbines vary their rotational speed by altering the pitch angle during power generation. Note that there are multiple relationships of the same type of time-series which share causal associations, such as *reactive power mean value* being related to the feature's *standard deviation* relation, *wind direction minimum value* being related to the feature's standard deviation etc.), mainly due to the presence of similar patterns in these features (as the 10-minute averages of such time-series from a similar context would share hidden confounders).

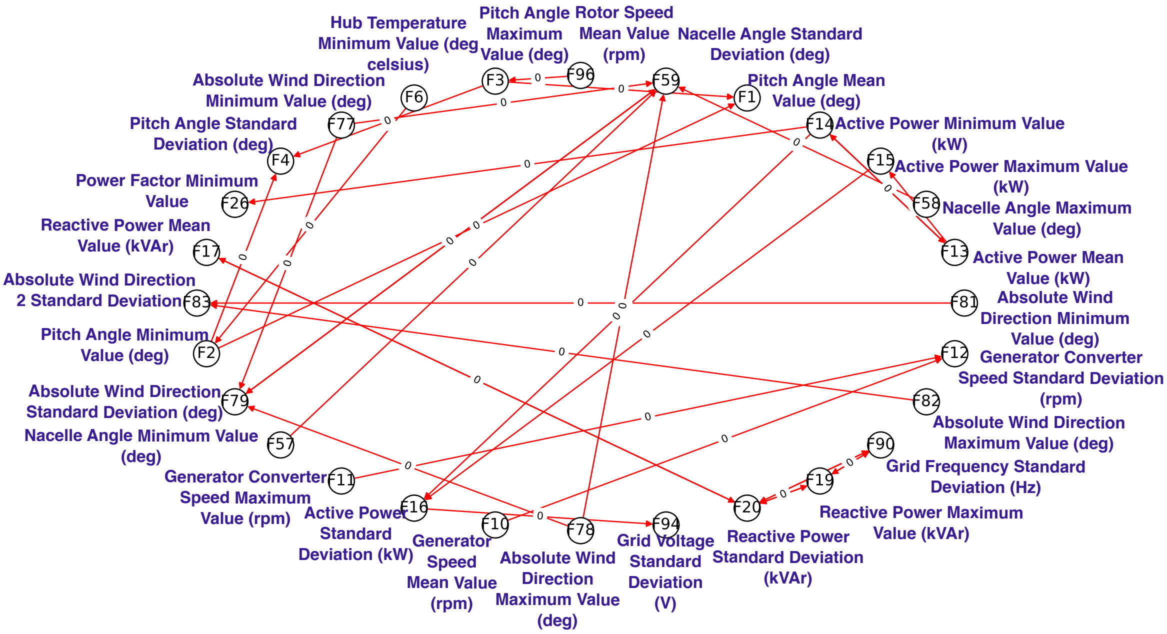

Hub Temperature Minimum Value (deg celsius)
F6
Pitch Angle Maximum Value (deg)
F3
Rotor Speed Mean Value (rpm)
F96
Nacelle Angle Standard Deviation (deg)
F59
Absolute Wind Direction Minimum Value (deg)
F77
Pitch Angle Mean Value (deg)
F1
Pitch Angle Standard Deviation (deg)
F4
Active Power Minimum Value (kW)
F14
Power Factor Minimum Value
F26
Active Power Maximum Value (kW)
F15
Nacelle Angle Maximum Value (deg)
F58
Reactive Power Mean Value (kVAr)
F17
Active Power Mean Value (kW)
F13
Absolute Wind Direction 2 Standard Deviation
F83
F81
Absolute Wind Direction Minimum Value (deg)
Pitch Angle Minimum Value (deg)
F2
F12
Generator Converter Speed Standard Deviation (rpm)
Absolute Wind Direction Standard Deviation (deg)
F79
F82
Absolute Wind Direction Maximum Value (deg)
Nacelle Angle Minimum Value (deg)
F57
F90
Grid Frequency Standard Deviation (Hz)
Generator Converter Speed Maximum Value (rpm)
F11
F19
Reactive Power Maximum Value (kVAr)
Active Power Standard Deviation (kW)
F16
F20
Reactive Power Standard Deviation (kVAr)
F10
Generator Speed Mean Value (rpm)
F78
Absolute Wind Direction Maximum Value (deg)
F94
Grid Voltage Standard Deviation (V)

Fig. 5.6 Hidden confounders identified during normal turbine operation, with a 80% significant increase in loss between the original features' time-series and the intervened AD-DSTCN model output.

Case 2: Identified hidden confounders for yaw system anomaly, with a significance measure of 0.80 The hidden confounders identified during a yaw system anomaly are shown in Figure 5.7. There is a causal association between the absolute wind direction and nacelle angle. This inference is reasonable, given that the wind direction time-series is highly variable, and depending on the nacelle measuring instrument's recorded wind direction, appropriate yawing is directed by the turbine's control system. As the wind direction continously varies, there is a change in the yaw system's operation with frequent start/stop of the system in order to keep the turbine aligned appropriately to the wind, directly leading to rotor torque fluctuations and resistance torque variations, which affect the loading in the yaw system and cause load fluctuations (Wan et al., 2015a). This can directly affect the nacelle and cause it to vibrate, owing to the rise in speed fluctuations and thus making the system unstable. Nacelle angle, by definition represents the angle between the wind turbine's rotor axis and the true north (Mittelmeier and Kühn, 2018), and the prevailing causal effect *(occurring at a delay of 0 time-steps)* possibly signifies high aerodynamic yaw loads, leading to the nacelle's spillage. The turbine's efficiency to generate power is thus also affected directly (Reddy et al., 2017), leading to production losses shortly after the occurrence of the anomaly *(a temporal delay of 1-2 time-steps)*. This is identified through causal relations between power characterisation metrics in the graph.

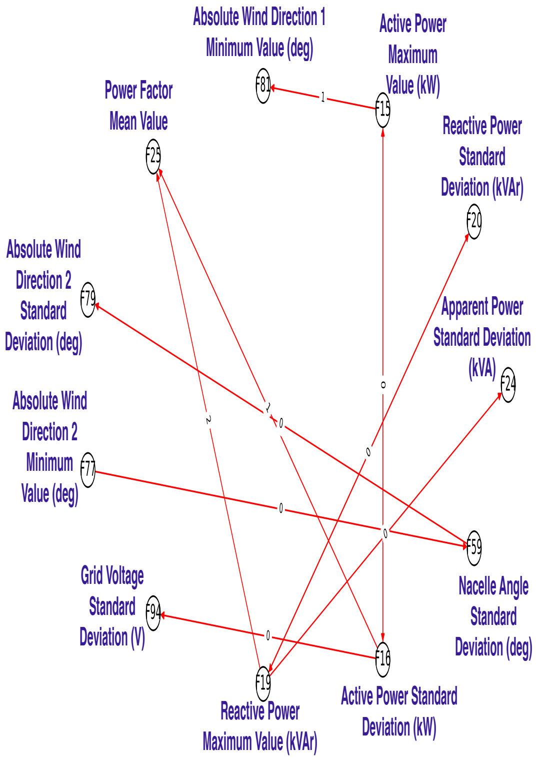

Absolute Wind Direction 1 Minimum Value (deg)
F81
Active Power Maximum Value (kW)
F15
Power Factor Mean Value
F25
Reactive Power Standard Deviation (kVAr)
F20
Absolute Wind Direction 2 Standard Deviation (deg)
F79
Apparent Power Standard Deviation (kVA)
F24
Absolute Wind Direction 2 Minimum Value (deg)
F77
F59
Nacelle Angle Standard Deviation (deg)
Grid Voltage Standard Deviation (V)
F94
F19
Reactive Power Maximum Value (kVAr)
F16
Active Power Standard Deviation (kW)
1
2
1
0
0
0
0
0
0

Fig. 5.7 Hidden confounders identified during anomaly in yaw system, with a 80% significant increase in loss between the original features' time-series and the intervened AD-DSTCN model output.

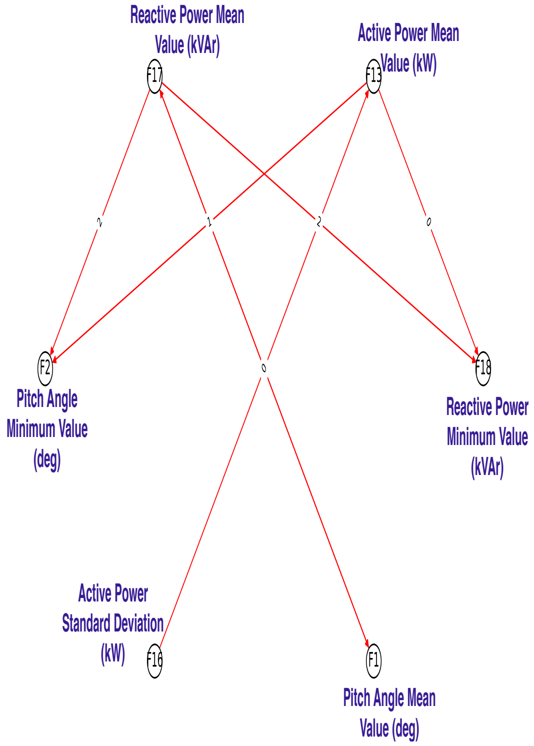

Reactive Power Mean Value (kVAr)
F17
Active Power Mean Value (kW)
F13
F2
Pitch Angle Minimum Value (deg)
F18
Reactive Power Minimum Value (kVAr)
Active Power Standard Deviation (kW)
F16
F1
Pitch Angle Mean Value (deg)
2
1
2
0
0

Fig. 5.8 Hidden confounders identified during anomaly in pitch system, with a 80% significant increase in loss between the original features' time-series and the intervened AD-DSTCN model output.

Case 3: Identified hidden confounders for pitch system anomaly, with a significance measure of 0.80 The hidden confounders identified during an anomaly in the pitch system are depicted in Figure 5.8. There is a causal association in between turbine pitch angle and reactive power. This inference is reasonable, as a deviation from the optimum value of the pitch angle (which is generally predefined) affects the power dynamics (including active and reactive power) (Godwin and Matthews, 2013). As per existing literature (Rezaeiha et al., 2017), for a vertical axis turbine, the variation in pitch angle shifts the instantaneous moment on turbine blades, thereby affecting the power performance and aerodynamics (Reddy et al., 2017) during the reactive power control mode, as is identified by our model. In addition, it can be seen that the relation takes effect during an actual operational inconsistency (outlined by the Pitch System Interface Alarm), outlining a significant operational outlier in the pitch angle's dynamic response, which potentially causes the control system to fail.

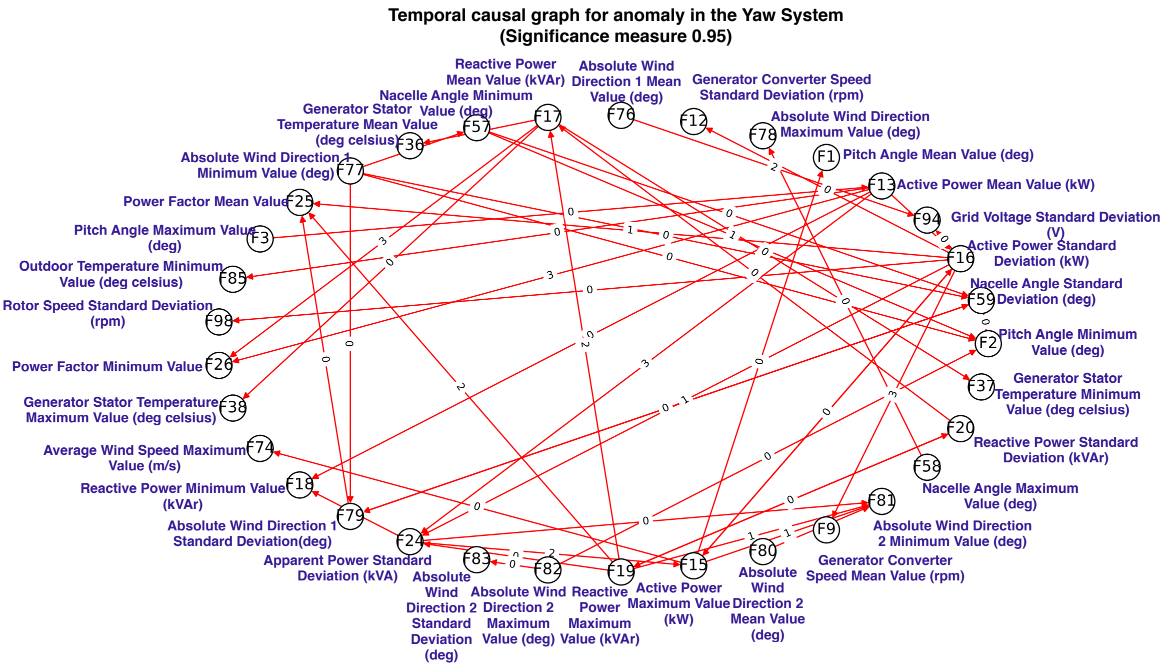

Temporal causal graph for anomaly in the Yaw System
(Significance measure 0.95)
Reactive Power Mean Value (kVAr)
Absolute Wind Direction 1 Mean Value (deg)
Generator Converter Speed Standard Deviation (rpm)
Nacelle Angle Minimum Value (deg)
Generator Stator Temperature Mean Value (deg celsius)
Absolute Wind Direction Maximum Value (deg)
Pitch Angle Mean Value (deg)
Absolute Wind Direction 1 Minimum Value (deg)
Active Power Mean Value (kW)
Power Factor Mean Value
Grid Voltage Standard Deviation (V)
Pitch Angle Maximum Value (deg)
Active Power Standard Deviation (kW)
Outdoor Temperature Minimum Value (deg celsius)
Nacelle Angle Standard Deviation (deg)
Rotor Speed Standard Deviation (rpm)
Pitch Angle Minimum Value (deg)
Power Factor Minimum Value
Generator Stator Temperature Minimum Value (deg celsius)
Generator Stator Temperature Maximum Value (deg celsius)
Reactive Power Standard Deviation (kVAr)
Average Wind Speed Maximum Value (m/s)
Nacelle Angle Maximum Value (deg)
Reactive Power Minimum Value (kVAr)
Absolute Wind Direction 2 Minimum Value (deg)
Absolute Wind Direction 1 Standard Deviation(deg)
Generator Converter Speed Mean Value (rpm)
Apparent Power Standard Deviation (kVA)
Absolute Wind Direction 2 Standard Deviation (deg)
Absolute Wind Direction 2 Maximum Value (deg)
Reactive Power Maximum Value (kVAr)
Active Power Maximum Value (kW)
Absolute Wind Direction 2 Mean Value (deg)
F57
F17
F76
F12
F78
F1
F36
F77
F13
F25
F94
F3
F16
F85
F59
F98
F2
F26
F37
F38
F20
F74
F58
F18
F81
F79
F9
F24
F83
F82
F19
F15
F80

Fig. 5.9 Hidden confounders identified during anomaly in yaw system, with a 95% significant increase in loss between the original features' time-series and the intervened AD-DSTCN model output.

Case 4: Identified hidden confounders for yaw system anomaly, with a significance measure of 0.95 As can be seen from Figure 5.9, while the learning model identifies the highest number of hidden confounders (which is curated by the higher significance measure utilised), this in turn comes at the cost of less credibility/explainablity for certain SCADA features. Amongst the reasonable relationships identified: there is causal association between reactive/apparent power and power factor [4], active power being causally related to wind speed [5], and the active power experiences variation with a change in generator converter speed.

Note that this temporal causal graph also shares some relations which were identified in Case 2. However, there are several inferred relationships which either appear to be

[4] By definition, the power factor is the ratio of the cosine angle between the apparent power to the active power.

[5] A variation in the turbine's rotor speed leads to a change in active power, which is a reasonable inference, given that the anomaly in yaw system (yaw error) causes inconsistencies in running operations of the turbine including the rotor speed and active power generated. (Wan et al., 2015a).

non-credible or are difficult to explain, such as: causal association between the generator stator temperature to nacelle angle and wind direction, relation from outdoor temperature to the active power, or more reasonable causations from pitch angle to reactive power. The relationships signify that any change in the nacelle angle leads to a variation in the generator stator temperature time-series, which when seen stand-alone is a meaningless relation. However, from the graph, it can be seen that the nacelle angle is additionally causally related to the pitch angle (wherein, the generator stator temperature time-series feature indirectly has this common cause-effect relationship with the pitch angle), which can be attributed to the situation of the yaw error affecting the operation of the pitch control system, thereby causing the generator stator temperature to share the above relationship. [6] These relations can be interpreted as situations wherein a co-located anomaly took prominence in a different turbine sub-component at the same instance of time (either arising as an effect of the occurring anomaly, or being an independent anomaly), which caused variations in the statistical characteristics of the multiple SCADA features (either as a direct consequence or owing to causation amongst multiple hidden confounders). Note that we do not claim the credibility of all identified causal associations to human domain experts and some relations can be hard to explain or interpret (Vasilyeva et al., 2018), but we believe that the inference can play a vital role in probing the predictions made by the AI learning model and thereby contributing to enhanced transparency in decisions. Engineers and technicians can explore these parameters further to better analyse the working of the AI model during the predicted faults, and possibly incorporate more focus on the identified confounders.

5.6 Discussion and Conclusion

In this chapter, a novel application of DL for temporal causal reasoning is proposed towards transparent decision making utilising operational wind turbine SCADA data. By using the operational status of the turbine as a target parameter, the learning model can discover novel associations and hidden predictor variables, which, to the best of our knowledge, are neglected by models in existing literature. Felgueira et al. (2019) have previously utilised Autoregressive Normal Behaviour Models for performing temporal causal inference, but their study neglects multiple cause-effect relationships which can be integral to the AI model's learning process and are identified by temporal causal graphs. It has also been observed that

[6] According to Li and Wang (2019), there is a correlation which is prevalent between the pitch angle and generator stator temperature time-series signals, particularly during dynamic monitoring of operational inconsistencies/anomalies in the pitch system, making these features related to a certain significant degree.

while the selection of a lower significance score makes the causal inference model identify fewer hidden confounders, more of these associations are relevant to the wind energy domain for O&M.

The proposed temporal causal inference model, despite its several advantages in aiding transparency to the black-box natured AI models, is not perfect, and we do not claim that all hidden confounders identified would directly be relevant or understandable by the turbine operators. Nonetheless, the proposed approach paves a clear path towards Explainable AI for the wind industry, with the internal parameters utilised by the black-box learners being clearly revealed through causal inference. We envisage that the causal reasoning model can be further improved in future, through utilisation of larger datasets and optimisation with human domain knowledge.

Finally, we demonstrate that causal inference is promising for the wind industry in benefiting from the advantages of Explainable AI, and can help inspire more wind farm operators in adopting AI models for decision support. In Chapter 6, we will focus on further fine-grained explanations for alarms in turbines by generating human-intelligible alarm messages and maintenance action strategies through Natural Language Generation (NLG) techniques to help engineers fix/avert the failures.

Chapter 6

Natural Language Generation for Explainable Decision Support

> The power of words is immense. A well-chosen word has often sufficed to stop a flying army, to change defeat into victory, and to save an empire.
>
> Emile de Girardin

This chapter is based on the work previously published by the author at the *International Joint Conference on Neural Networks* (Chatterjee and Dethlefs, 2020d) and the *NeurIPS Climate Change AI* workshop (Chatterjee and Dethlefs, 2019a).

6.1 Introduction

While the wind industry has witnessed significant interest in applying ML techniques directly to SCADA data for predicting anomalies in O&M (Si et al., 2017; Zaher et al., 2009), there is a clear paucity of data-driven intelligent decision support systems which cannot only predict occurrence of impending faults with high levels of accuracy but also generate clear and concise human-intelligible diagnoses of their cause(s). For engineers and technicians, the utility of present-day decision support systems is very limited, as predicting faults is of little use unless these systems can propose effective maintenance actions to avert/fix the failures (Chatterjee and Dethlefs, 2019a).

In this chapter, we apply Natural Language Generation (NLG) techniques to explore their feasibility in generating *informative* alarm messages alongside corresponding maintenance actions by using SCADA data to facilitate autonomous decision support in the wind industry. For any anomaly in the turbine's operation (deviation from normal behaviour), our goal is to **(1)** generate informative alarm messages outlining the details of the fault which occurs and the subcomponent(s) affected **(2)** propose most appropriate maintenance actions which can help engineers to fix (or avert) the prevailing fault. Given the nature of our task wherein the input spans long sequences of SCADA features with continuous values and very different combinations of such features can be relevant to an occurring fault, we propose the application of Transformers (Ashish Vaswani et al., 2017) towards performing sequence-to-sequence generation. Earlier work in utilising Transformers (Vaswani et al., 2017b) has shown that the provision of a multi-head attention mechanism and their ability to compute attention weights can contribute to making the model both faster in learning over long sequences as well as more accurate for some tasks, given that the output predictions are independent of the order in which the input sequences are processed (Vaswani et al., 2017b).

Our generation task is decomposed into 2 stages (**(i)** alarm and **(ii)** maintenance), wherein, a Transformer model is utilised for each stage. The Transformer 1 takes in as input a sequence of continuous-valued SCADA features and generates corresponding alarm messages for anomalies. Additionally, the multi-head attention mechanism in the model helps to obtain the sequence of relevant features which most likely contribute to the corresponding alarm. The model in second stage (Transformer 2) takes as input the sequence of features contributing to the fault along with the alarm type (i.e. the model outputs from Transformer 1), and finally generates the maintenance strategies which are most effective for the prevailing fault, in the form of natural language phrase templates, which are trained by utilising maintenance strategy samples which we have authored based on our domain knowledge.

The proposed Transformer 1 model utilised for alarm message generation is compared against conventional models for sequential tasks: the LSTM-based Seq2Seq NLG techniques with and without attention mechanisms. The Luong attention mechanism is utilised for the latter (Luong et al., 2015a). An evaluation of the percentage of alarms correctly predicted demonstrates that the Transformer learning model achieves an accuracy of up to 96.76%, outperforming baseline networks (by up to 18.76%) and additionally achieves higher computational speed (by up to 28.78%) in comparison to a vanilla encoder-decoder model with attention mechanism. It is found that the feature importance scores which are derived through the attention scores of the Transformer model are more reasonable as well as relevant in comparison to the baseline models. For the Transformer 2 model which is utilised for generating maintenance strategies, it is seen that the model achieves an accuracy of up to 75.35%, outperforming the baseline Seq2Seq(Att) model by up to 18%.

The chapter is organised as follows: Section 6.2 describes the concept of data-to-text NLG and discusses the motivation for applying them in wind turbine CBM. The proposed DL model for generating alarm messages tailored for wind turbine O&M is introduced in Section 6.3. Section 6.4 discusses the utilisation of Dual-Transformer models, a novel technique for generating human-intelligible maintenance actions corresponding to predicted alarms in the turbine. The development of a human-authored corpus for maintenance action templates is discussed in Section 6.5. Section 6.6 demonstrates the experiments performed for assessing the feasibility and application of NLG in the wind industry. Section 6.7 shows the results for evaluating the accuracy of predicted alarm types and generated maintenance action messages. Finally, Section 6.8 concludes the chapter and provides the background for utilising multimodal Knowledge Graphs towards NLG of comprehensive O&M reports in Chapter 7.

6.2 Data-to-text generation techniques in NLG

Data-to-text generation techniques have been applied to a multitude of domains, and have seen immense success in areas such as generating weather forecasts (Sripada et al., 2004), knowledge entities (Vougiouklis et al., 2018), towards sports commentaries (Chen et al., 2010) etc., amongst many others. Additionally, there is often an NLG component in dialogue systems which facilitates transformation of semantic inputs to utterances, which can ultimately be presented to the end users (Hastie et al., 2016; Mairesse et al., 2010).

6.2.1 Deep Learning for NLG

Most of the work prevalent in statistical NLG pertains to Sequence-to-Sequence RNNs (Sutskever et al., 2014), which perform the learning process by mapping non-linguistic representations of the input (e.g. a collection of semantic slots which are to be expressed, entities from meteorological data, pixels present in images etc.) into a sequence of words which describe the input in the form of natural language. There have been several extensions proposed for such vanilla models for sequence-to-sequence generation e.g. through semantically conditioning outputs towards increasing the semantic accuracy (Wen et al., 2015), pre-processing inputs appropriately for increasing the quality of semantic outputs (Nie et al., 2018), applying copy actions (Gehrmann et al., 2018) or modified techniques such as Pointer Sentinel networks (Merity et al., 2016) towards increasing the slot transfer accuracy, or incorporating linguistic information into purely learnt existing models through injection, which can be helpful in capturing more fine-grained domain-specific lexico-syntactic regularities (Dethlefs, 2017; Dusek and Jurcicek, 2016).

An alternative to the Sequence-to-Sequence NLG models are Variational Autoencoders (VAEs), which are generally utilised for encoding input representations and then finally, decoding them back while ensuring minimal loss. According to the realisation of recent models, it has been seen that by incorporating “noise" into the inputs e.g. information on target semantic slots, VAEs have the ability to transform semantic input sequences into corresponding lexico-syntactic surface forms (Freitag and Roy, 2018), in a manner very similar to other NLG architectures. Semeniuta et al. (2017) have previously demonstrated that a combination of VAEs with convolutional and deconvolutional techniques can significantly help to reduce inconsistency in the generated outputs.

6.3 Learning models

We first introduce our baseline models, the Seq2Seq architecture with and without Luong attention and then develop the Transformer model from these descriptions and discuss the relevance of each for our generation scenario.

6.3.1 Standard Seq2Seq Model

Sequence-to-Sequence models were first proposed by Cho et al. (2014b); Sutskever et al. (2014) and have been extensively popular for handling sequence-to-sequence prediction problems such as machine translation, NLG, and dialogue, see Section 6.2. The model consists of two key components, an encoder and a decoder, which map a source input sequence to a target output sequence of variable length. Encoder and decoder are normally modelled as RNNs, such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997b) models or Gated Recurrent Units (GRUs) (Cho et al., 2014b) and have been shown to successfully learn from sequences with long-term dependencies, making them particularly suited to time-series data.

As the SCADA data from a wind turbine is a typical time-series with multiple features measured from various sensors at generally 10-minute intervals, the model would learn to map a sequence of features to predict a corresponding alarm message describing operational behaviour of the turbine at any given interval. Figure 6.1 depicts the architecture of the learning model.

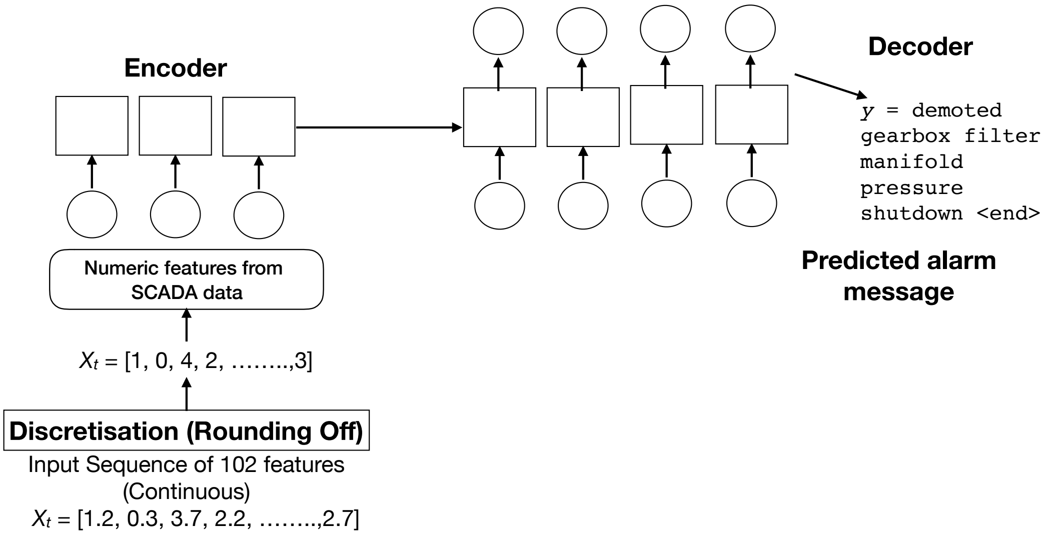

Encoder
Decoder
y = demoted
gearbox filter
manifold
pressure
shutdown <end>
Predicted alarm message
Numeric features from SCADA data
Xt = [1, 0, 4, 2, ...,3]
Discretisation (Rounding Off)
Input Sequence of 102 features
(Continuous)
Xt = [1.2, 0.3, 3.7, 2.2, ...,2.7]

Fig. 6.1 Architecture of the Seq2Seq model – an encoder and a decoder module in the model utilise LSTM networks to learn fixed dimensional hidden vector representations of the inputs for computing the output.

The Seq2Seq model takes as input a sequence of SCADA features $\mathbf{x}_t = (x_1, \ldots, x_N)$ and outputs a sequence of words describing the internal state of the turbine $\mathbf{y}_t = (y_1, \ldots, y_M)$, where t is a time-step. To accomplish this, an LSTM computes a hidden representation $\mathbf{h}$ through iterative updates to a non-linear activation function $f(\mathbf{x}_t, \mathbf{h_{t-1}})$. The goal is to minimise the loss function:

$$L(\mathbf{x}, \mathbf{y}) = -\frac{1}{N} \sum_{n \in N} x_n \log y_n, \tag{6.1}$$

using e.g. (as in our case) sparse categorical cross-entropy. The LSTM will compute $\mathbf{h}$ given a number of gates that control the retention or elimination of new information, see e.g. Graves (2013) for details. In short, we compute LSTM updates as:

$$\mathbf{h}_t = LSTM(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t+1} + \mathbf{b}_h), \tag{6.2}$$

with weight matrix $\mathbf{W}$ and bias term $\mathbf{b}$. Although the vanilla Seq2Seq model presented is fairly competent in mapping an input sequence to its corresponding target, in recent years, it has witnessed a popular refinement with the introduction of *attention* (Bahdanau et al., 2015; Luong et al., 2015a). In contrast to the standard model which uses a fixed dimensional hidden vector representation to compute an output, the attention mechanism helps the model to specifically focus on only the most important parts of the input sequence during decoding.

6.3.2 Seq2Seq Model with Luong Attention

For our study, we use the Luong attention mechanism (also referred to as the multiplicative approach) (Luong et al., 2015a), an expanded version of Bahdanau attention (popularly referred to as the additive approach) (Bahdanau et al., 2015). The two approaches differ as follows:

1. While Luong attention uses three different types of scoring functions i.e. **dot, general and concat**, Bahdanau uses only **concat**.
2. Luong attention uses the previous output of the decoder to estimate the alignment vector, whereas the Bahdanau uses the previous time-step output for the estimation.

We define two additional vectors in our vanilla model – the context vector **c** and alignment vector $\alpha_t = (\alpha_1, \ldots, \alpha_T)$. The latter is of the same length as the source sequence and represents a word's weight contributing towards the loss.

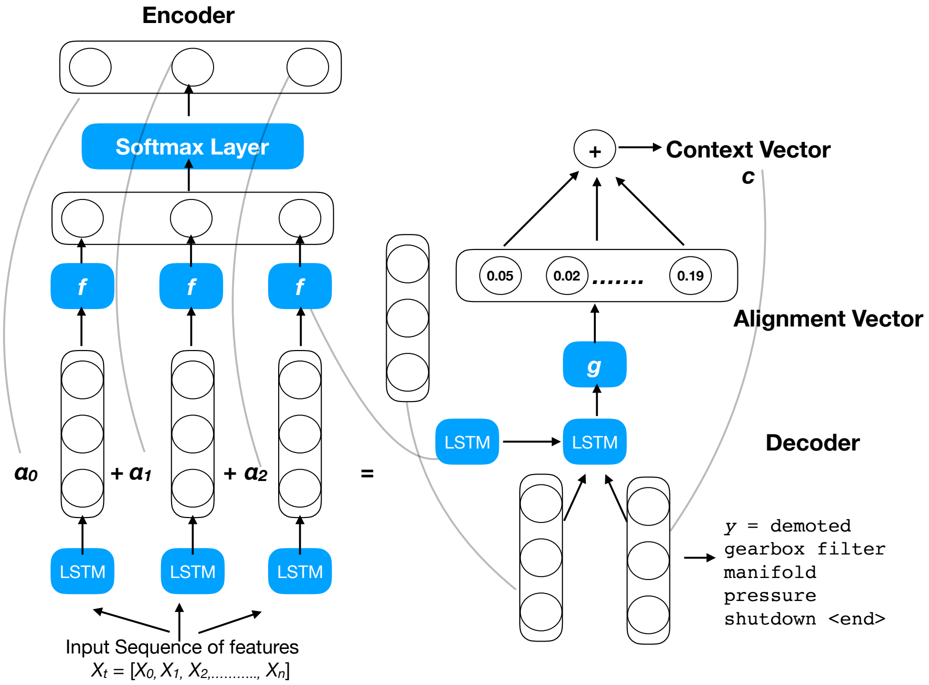


Fig. 6.2 Seq2Seq model with Luong attention – the encoder and decoder module utilise LSTM learners to map the input sequences to the corresponding targets, while estimating attention weights through the alignment and context vectors to facilitate the model to focus on vital parts of the input data.

Assuming successive hidden states of the encoder to be denoted as $e_{t'}$, we estimate:

$$\alpha_{t'} = f\left(h_{t-1}, e_{t'}\right) \in \mathbb{R} \qquad \text{for all } t', \tag{6.3}$$

where $f\left(h_{t-1}, e_{t'}\right) \mapsto \alpha_{t'} \in \mathbb{R}$ is a real-valued function dependent on the hidden states h_t and $e_{t'}$. We also compute $\alpha = \mathrm{softmax}(\alpha)$, where a softmax function is used for normalising the alignment vector sequence and computing the weighted average of the encoder's hidden states, as outlined by Genthial (2017). Finally,

$$c = \sum_{t'=0}^{n} \alpha_{t'} e_{t'} \tag{6.4}$$

represents the context vector, which estimates the weighted average of the encoder's prevailing output, and is used to evaluate the final sentences at the decoder end. Context vector **c** is used for predicting target sequence **y** and is also able to provide the scores for the features in our SCADA data which are likely contributing to the predicted alarm message, as with any attention module. Figure 6.2 illustrates the architecture of a Seq2Seq model with Luong attention.

6.3.3 Transformers for NLG

Transformers are more recent models for NLG tasks, which have which have been shown to outperform conventional sequence-to-sequence architectures in a variety of tasks in the last few years, such as machine translation (Vaswani et al., 2017b), question-answering, natural language understanding, common sense inference (Devlin et al., 2019), video captioning (Zhou et al., 2018) etc. The key idea in a Transformer architecture focuses on eliminating the requirements for recurrence and convolutions, and instead computing attention weights over input sequences by using positional embeddings. In a Transformer, the conventional attention mechanism (wherein, the output sequence attends to the input) is extended through self-attention, facilitating the inputs and outputs to both attend to themselves as well as the target attending to the source (Vaswani et al., 2017b). This mechanism has seen greater success in some domains in comparison to the conventional attention-based Seq2Seq model. Additionally, the elimination of recurrence lowers the computational cost, which potentially makes Transformers highly attractive for real-world and real-time data-to-text generation tasks.

Few studies have demonstrated promising applications of Transformers for data-to-text generation in the domain of decision support and recommendation systems. For instance, in Gehrmann et al. (2018)'s entry to the 2018 E2E challenge, the authors applied Transformers to generate restaurant recommendations directly from discrete semantic slots and achieved good results, though the Transformer was not their best performing model for the task. Transformers have also been utilised for video caption generation (Zhou et al., 2018) and have been found to outperform multiple competitive LSTM baseline models. Chen et al. (2019b) have previously trained NLG system for a new domain by utilising small amounts of labelled data and an extensive pre-trained language model to facilitate inheritance of general-purpose linguistic knowledge. The basic model in this case is a LSTM sequence-to-sequence architecture in combination with a field-gating encoder (refer Liu et al. (2018)), but the pre-trained language model is a Transformer consisting of 12 layers (Radford et al., 2018). Feng et al. (2020a) proposed a hierarchical CNN-Transformer model with explicit attention for generating informative rationales during clinical decision support with the English MIMIC-III dataset. Some other studies have utilised Transformers for Argument Mining (AM) towards relation-identification (e.g. of clinical terms) in long pieces of text in the healthcare domain (Mayer et al., 2020), and have shown that they significantly outperform conventional end-to-end AM systems (such as those utilising LSTM, GRU, CRF etc.).

Similar to Seq2Seq models, the Transformer architecture consists of an encoder and a decoder. However, it contains multiple attention constraints, referred to as multi-head attention, which constitute the essence of its performance (Thiruvengadam, 2019; Vaswani et al., 2017b):

- Encoder's self attention, wherein the source sequence attends to itself.
- Decoder's self attention, wherein the target sequence attends to itself.
- The target attending to the source sequence (i.e. conventional attention as in Mikolov et al. (2010); Sutskever et al. (2014)).

In contrast to traditional Seq2Seq models, Transformers do not depend on RNNs for processing the sequential data, giving them several advantages which include higher computational efficiency. As RNNs read sequences one word at a time along with their sequential nature of processing data makes them less efficient in performing parallel GPU processing in comparison to Transformers, which primarily utilise feed-forward networks and matrix multiplication in the learning process (Tubay and Costa-Jussà, 2018).

Figure 6.3 illustrates a simplified Transformer architecture which consists of an encoder and a decoder module, see Vaswani et al. (2017b) for further details. As can be seen, the Transformer consists of a number of layers which can be stacked together (shown as N_x) to form a deeper architecture.

Encoding The encoder module consists of three stages, wherein, the first stage projects the input sequence into vector space. Positional embeddings of the input sequence are incorporated to record token positions in the input and account for the absence of a RNN. Positional embeddings PE are computed as:

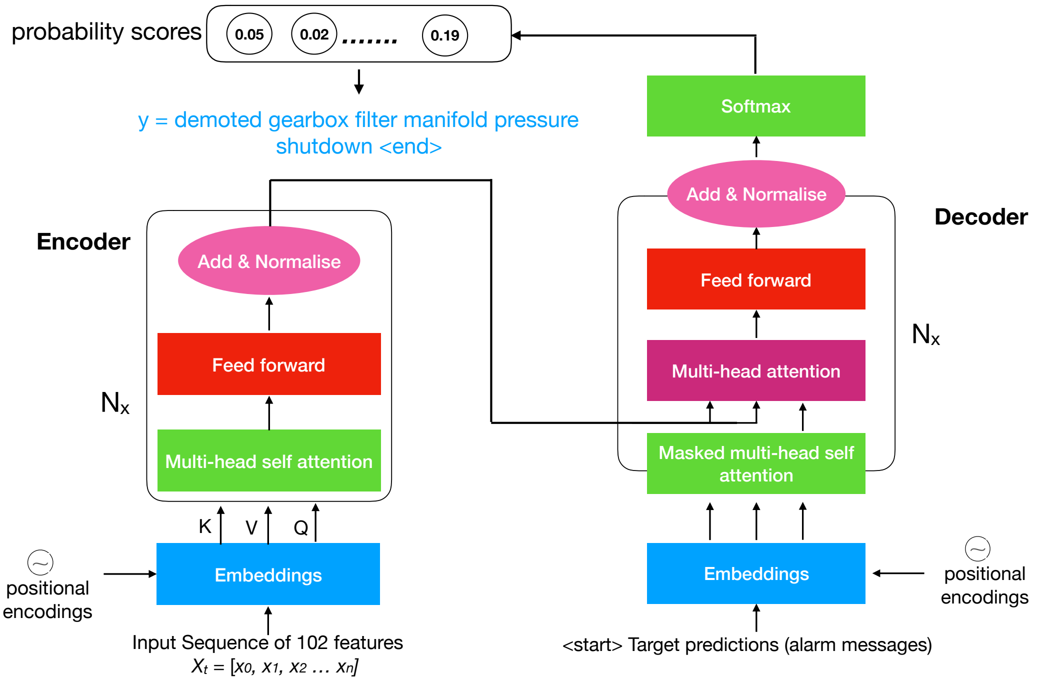

probability scores
0.05
0.02
....
0.19
y = demoted gearbox filter manifold pressure shutdown <end>
Softmax
Add & Normalise
Decoder
Encoder
Add & Normalise
Feed forward
Nx
Multi-head attention
Feed forward
Nx
Multi-head self attention
Masked multi-head self attention
K
V
Q
positional encodings
Embeddings
Embeddings
positional encodings
Input Sequence of 102 features
$X_t = [x_0, x_1, x_2 \dots x_n]$
<start> Target predictions (alarm messages)

Fig. 6.3 Simplified architecture of the Transformer – the model consists of an encoder and a decoder incorporated with positional embeddings of the input sequence, along with a multi-head attention mechanism that facilitates learning in the absence of recurrence and convolutions.

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \tag{6.5}$$

and

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{\text{model}}}\right), \tag{6.6}$$

where d_{model} denotes the depth of the Transformer.

The multi-head attention stage is directed by three key parameters – *Query* Q, denoting the component that pays attention and represented as a vector of the semantic tokens corresponding to a specific input word, along with key-value pairs consisting of all the words in the sequence, represented as *Key* K and *Value* V – which signify all the initial input word vectors which the model ends up attending to. For encoding, V corresponds to the same word sequence as Q. The overall goal is to compute a weighted sum over values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key (Allard, 2019):

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{6.7}$$

where $\sqrt{d_k}$ is a scaling constant equal to the square root of the dimension of keys. QK^T computes a similarity matrix between queries Q and keys K, which is applied to the source sequences during encoding, but Q is drawn from the target sequences at the decoding stage. A position-wise feedforward neural network then identifies the input projections which should be eventually used at the decoder end.

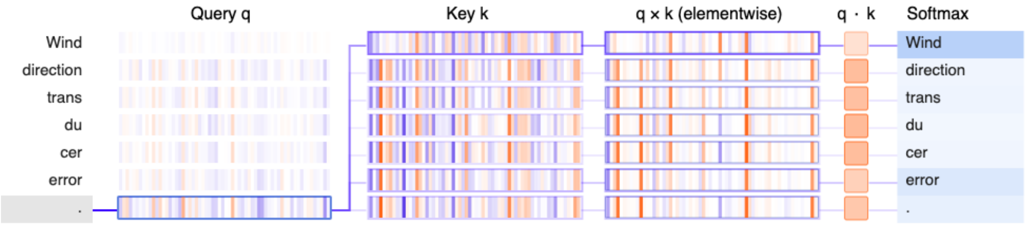


Fig. 6.4 Neuron visualisation within the key and query vectors in the Transformer architecture.

An example visualisation of the multi-head attention mechanism is provided in Figure 6.4. In this mechanism, the individual neurons in the key and query vectors are utilised for computing the final attention weights for the generated alarm messages developed through BertViz (Vig, 2019). The model's final output is generated during the decoding process by utilising the past and the final hidden representations of the source sequence that are received from the encoder. Note that a softmax layer is also utilised in the process for computing the scores of the words which are to be predicted in the target sequence, which in case of our problem are the alarm message predictions. More details on the Transformer architecture can be found in Vaswani et al. (2017b).

Decoding The decoder module contains five discrete stages, of which the initial two are the same as in the encoder (with embeddings offset by one position). However, masked multi-head self-attention is used instead which restricts the decoder to only focus on past words in the sequence and prevents it from looking ahead. At the third stage, the multi-head attention block captures past and final hidden representations received from the encoder. The fourth layer is similar to the encoder's feedforward network, and finally the softmax layer is used to capture the scores of words to be predicted in the target sequence.

In this chapter, we focus on extending the original Transformer architecture towards data-to-text generation by utilising continuous numeric inputs from SCADA data. Particularly, our goal is to appropriately utilise the Transformer's attention scores to facilitate transparency in the model's decisions besides achieving accuracy in predictions.

6.4 Dual-Transformer model for intelligent decision support in wind turbines

We propose the modelling of our NLG system for intelligent decision support in O&M of wind turbines by utilising two stages – **Stage (1):** The input is a sequence of SCADA features and the output generated is the alarm descriptions of fault events. **Stage (2):** The most appropriate (and effective) maintenance action strategies are predicted in the form of natural language phrases. For the Stage 1, given that the SCADA data from turbines typically consist of multiple time-series features from various sensors (generally measured at an interval of 10 minutes), the Transformer would learn to map the sequence of SCADA input features to corresponding alarm messages which describe the turbine's operational behaviour at any given interval of time. Stage 2 is analogous to a sequence generation task, wherein, each maintenance strategy can consist of multiple actions (which can occur in different orders) to fix/avert any occurring faults.

Below, the Transformer's role in each stage is described towards identification of faults and alarm message generation (Stage 1) and performing content selection of appropriate maintenance action strategies (Stage 2).

6.4.1 Stage 1: Alarm message generation

In the first stage, a sequence of SCADA features $\mathbf{x}_t = (x_0, x_1, x_2, \ldots, x_N)$ is provided as input, and the internal state of the turbine is predicted as the output in the form of a sequence of symbols $\mathbf{y}_t = (y_0, y_1, y_2, \ldots, y_N)$, wherein, t denotes a time-step. This phenomenon is represented in Figure 6.5. In this case, the Transformer's role is to learn when a fault would likely occur (to facilitate an early alarm to be raised) as well as estimate and provide most likely causes of the occurring fault. The model's attention weights are utilised for achieving the latter, refer to Figure 6.8 for an example.

The integer-encoded class of the alarm (representing the type of alarm message) predicted in Stage 1 along with indices of the identified relevant contributing features are passed on further to Stage 2 towards content selection and generation of maintenance action strategies, which can eventually play an integral role in fixing/averting a fault.

6.4.2 Stage 2: Content selection of maintenance action strategies

The second stage of the proposed network is modelled as a separate Transformer model towards performing content selection and generation of appropriate maintenance strategies.

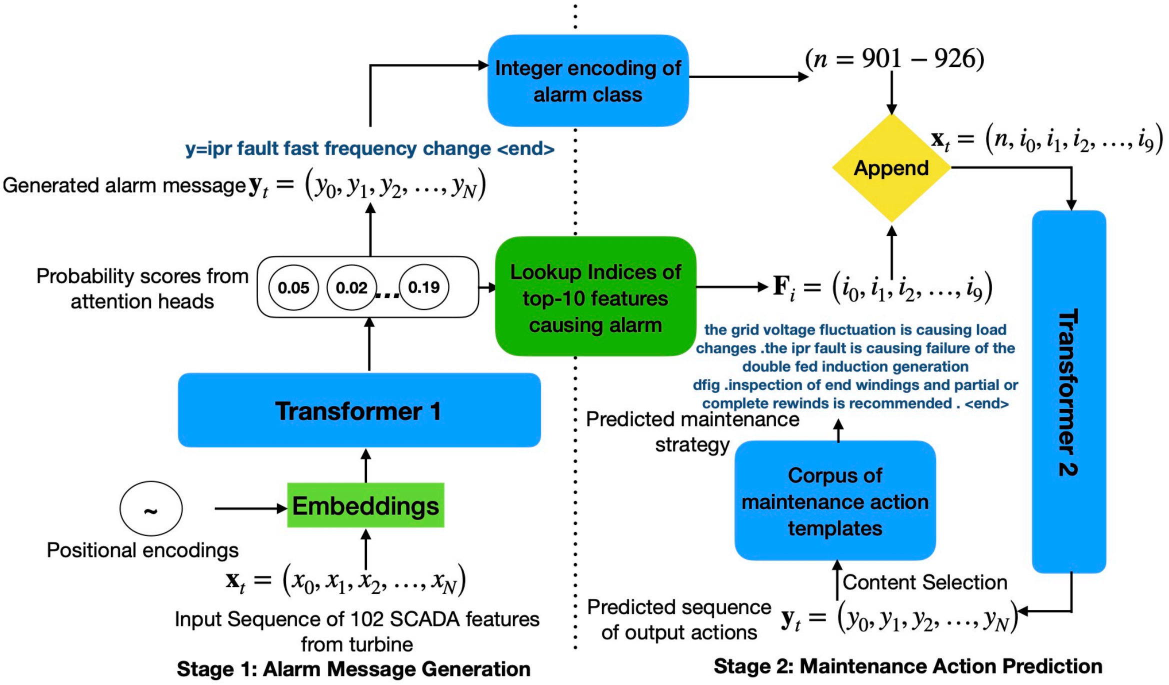

Integer encoding of alarm class
$(n = 901 - 926)$
y=ipr fault fast frequency change <end>
Generated alarm message $\mathbf{y}_t = (y_0, y_1, y_2, \ldots, y_N)$
Append
$\mathbf{x}_t = (n, i_0, i_1, i_2, \ldots, i_9)$
Probability scores from attention heads
0.05
0.02
0.19
Lookup Indices of top-10 features causing alarm
$\mathbf{F}_i = (i_0, i_1, i_2, \ldots, i_9)$
the grid voltage fluctuation is causing load changes .the ipr fault is causing failure of the double fed induction generation dfig .inspection of end windings and partial or complete rewinds is recommended . <end>
Transformer 1
Transformer 2
Predicted maintenance strategy
Corpus of maintenance action templates
Embeddings
~
Positional encodings
$\mathbf{x}_t = (x_0, x_1, x_2, \ldots, x_N)$
Input Sequence of 102 SCADA features from turbine
Content Selection
Predicted sequence of output actions
$\mathbf{y}_t = (y_0, y_1, y_2, \ldots, y_N)$
Stage 1: Alarm Message Generation
Stage 2: Maintenance Action Prediction

Fig. 6.5 Combining Transformer 1 for alarm message generation and Transformer 2 for content selection of maintenance actions – the integer-encoded class of the alarm message predicted in Stage 1 along with indices of the contributing features identified through the attention mechanism are utilised in Stage 2 for content selection.

The working of Transformer 2 is exactly similar to that of Transformer 1 – the model would predict a sequence consisting of a collection of maintenance action messages (which together form a complete strategy) based on the input sequence of alarm and the identified features in Stage 1 which likely lead to the alarm. This process is depicted in Figure 6.5.

More specifically, an alarm message generated by Transformer 1 is initially integer-encoded into one of 26 different possible categories of alarms $(n = 901 - 926)$. Next, the predicted alarm's class n is appended together with a list of the top-10 relevant features $(\mathbf{F}_i = (i_0, i_1, i_2, \ldots, i_9))$ which likely cause the alarm as identified via the attention mechanism of Transformer 1. This gives rise to a new sequence of features $\mathbf{x}_t = (n, i_0, i_1, i_2, \ldots, i_9)$. The updated sequence is utilised as an input to Transformer 2, which finally provides the most effective maintenance actions by performing content selection of relevant templates from a collection of key-value pairs of actions present in a dictionary. More details on the phrase-based natural language templates utilised for content selection in Stage 2 is described in Section 6.5. The topology of combining Transformer 1 and Transformer 2 to finally generate the maintenance strategies is depicted in Figure 6.5.

6.5 Development of maintenance action templates

In order to produce the final output in the proposed learning model, we develop a corpus of maintenance action templates which correspond to multiple types of faults that historically occurred in the turbine. With the presence of 26 discrete classes of alarms in our LDT SCADA dataset, we developed 167 sub-phrases spanning across these classes based on our domain knowledge, where each sub-phrase details a specific maintenance action strategy (which we ultimately aim to generate). Each strategy is made up of a sequence of multiple actions i.e. a sequence consisting of multiple sub-phrases. Note that it is possible for different alarms to share the same maintenance strategy (in cases wherein the same actions would fix a fault), or conversely, it is also possible for the same alarm message to require different types of maintenance actions (in cases wherein the underlying causes leading to the alarm differ). This phenomenon is visually depicted in Figure 6.6 through an example Circos plot (Kassebaum, 2020). As can be clearly visualised from this plot, there is no independence across multiple alarm classes in terms of the maintenance strategies required to fix the faults, and an intersection is present amongst the actions for multiple different cases (e.g. for the maintenance action required for an alarm in the yaw brake (type 1), there is also a causal association with the maintenance action pertaining to the blades alarm (type 19)). Thereby, given that it is possible for two different alarms to share the same maintenance actions, amongst the 167 templates, there were 102 sub-phrases completely unique to any of the individual alarm classes. The human-authored maintenance action templates used in this chapter are made publicly available [1]. The template sub-phrases average to 6.073 words.

In our approach, a complete maintenance strategy would be a text output which consists of sub-phrases selected by Transformer 2. There was a major challenge with our maintenance action templates – training with a small corpus and the prevalence of class imbalance (with some maintenance actions being more prominent than others) can affect the generated predictions by incorporating unwanted bias in the model. To overcome this problem, the human-authored corpus of maintenance actions was over-sampled via SMOTE across the given SCADA features and corresponding alarms, leading to an overall dataset with 1,055 samples.

The maintenance action sub-phrase templates were collected and stored in the form of multiple key-value pairs in a dictionary, as shown in Figure 6.7. Given the indices of the top-10 most likely features contributing to the alarm as identified by Transformer 1 and the process of final content-selection performed by Transformer 2, the proposed model can

[1] Turbine Maintenance Action Templates:
https://github.com/joyjitchatterjee/TurbineMaintenanceTemplates

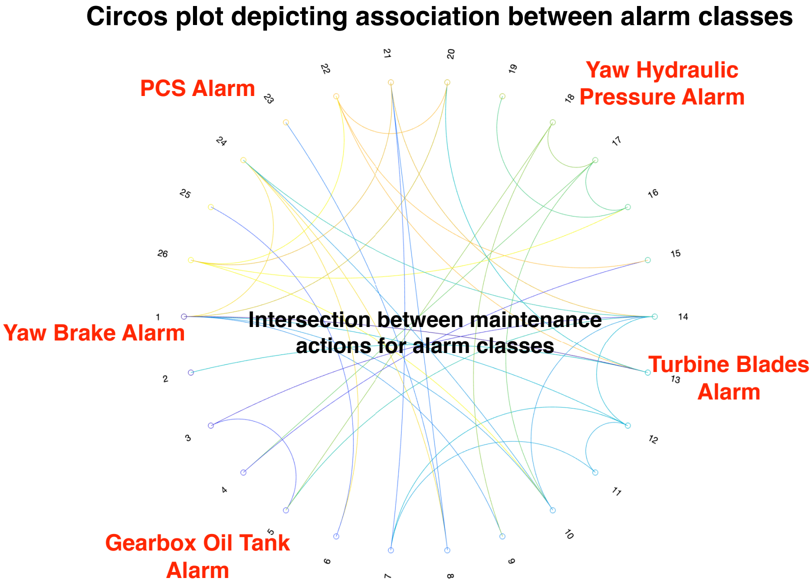
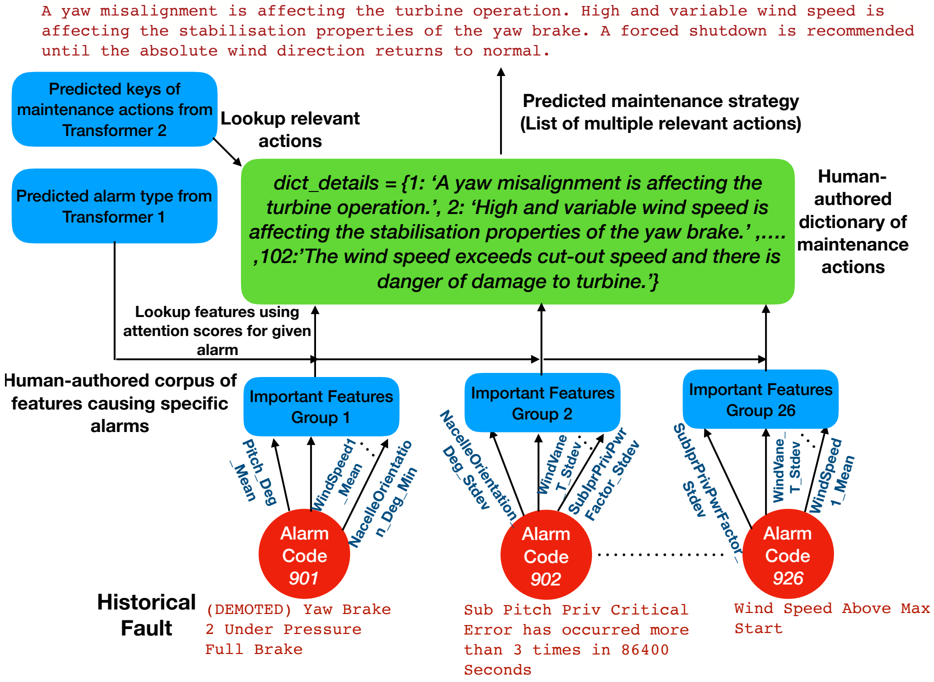


Fig. 6.6 Circos plot example visualising the relationship between different alarm classes. The alarm classes are clearly not independent, and it is thereby possible to have shared maintenance actions.


A yaw misalignment is affecting the turbine operation. High and variable wind speed is affecting the stabilisation properties of the yaw brake. A forced shutdown is recommended until the absolute wind direction returns to normal.

dict_details = {1: 'A yaw misalignment is affecting the turbine operation.', 2: 'High and variable wind speed is affecting the stabilisation properties of the yaw brake.' ,…. ,102:'The wind speed exceeds cut-out speed and there is danger of damage to turbine.'}

Predicted keys of maintenance actions from Transformer 2
Lookup relevant actions
Predicted maintenance strategy (List of multiple relevant actions)
Predicted alarm type from Transformer 1
Human-authored dictionary of maintenance actions
Lookup features using attention scores for given alarm
Human-authored corpus of features causing specific alarms
Important Features Group 1
Important Features Group 2
Important Features Group 26
Pitch_Deg_Mean
WindSpeed1_Mean
NacelleOrientation_Deg_Min
NacelleOrientation_Deg_Stdev
WindVane_T_Stdev
SubIprPrivPwrFactor_Stdev
SubIprPrivPwrFactor_Stdev
WindVane_T_Stdev
WindSpeed1_Mean
Alarm Code 901
Alarm Code 902
Alarm Code 926
Historical Fault

(DEMOTED) Yaw Brake 2 Under Pressure Full Brake

Sub Pitch Priv Critical Error has occurred more than 3 times in 86400 Seconds

Wind Speed Above Max Start

Fig. 6.7 Templates for content selection of maintenance actions in Transformer 2

utilise the human-authored maintenance sub-phrases to generate the alarm type predictions as well as suggest the most appropriate maintenance strategy to avert/fix the fault.

6.6 Experiments

6.6.1 Stage 1: Generation of alarms

As the proposed Dual-Transformer model's first stage focuses on identifying specific faults which occur in the turbine before the model can predict any maintenance actions, we utilise the sequence of LDT SCADA features as input along with the historical labelled alarm messages to train the model. TensorFlow (Abadi and et al., 2015) is utilised to develop the models for all of the three different learning algorithms which we intend to compare:-

- **Seq2Seq**: The Seq2Seq model based on the vanilla LSTM (Sutskever et al., 2014). 200-dimensional word embeddings are utilised for the model, with 64 hidden neurons, learning rate of 0.001, a dropout of 0.1, Adam optimisation and trained over 200 epochs.
- **Seq2Seq (Att)**: The Seq2Seq model based on the vanilla LSTM with Luong attention (Luong et al., 2015a), with the same hyperparameters as Seq2Seq. For the attention mechanism, we utilise the *concat* score function towards computing alignment vectors, together with the *dot* and *general* which are already prevalent in the Luong attention mechanism.
- **Transformer**: The Transformer model consisting of multi-head attention mechanism, refer Section 6.3.3 for more details. For the model, 8 multi-head attention heads are utilised, along with a model size of 64 with 3 dense layers for each head. The model's learning rate was decayed through the *WarmupThenDecaySchedule* class available in TensorFlow.

The dataset is split into training and test sets with a 80%-20% ratio, and a batch size of 32 is utilised.

6.6.2 Stage 2: Generation of maintenance strategies

For the second stage of the Transformer, a mapping is learnt from the input sequence consisting of the alarm type with the list of relevant features to the output sequences which consist of maintenance actions. In this case, the latter are represented in the form of integers which point to the developed dictionary made up of text-based templates. For this stage, a model size of 128 is utilised along with a vocabulary size of 1,300 words. Similar to Stage 1, a train-test split ratio of 80%-20% was used with a batch size of 16 in this case. For Stage 2, 500 training epochs were used.

The Seq2Seq(Att) model based on the vanilla LSTM with Luong attention was utilised as the baseline in this stage, with 128 hidden neurons, a learning rate of 0.001, dropout of 0.1, Adam optimisation, and training over 500 epochs. A vanilla Seq2Seq model without attention mechanism was not utilised for comparison in this stage, as the list of important features are necessary for the model in Stage 2 to perform content selection of appropriate maintenance actions, which can only be feasible with an attention mechanism.

All alarm messages were tokenized using *fit_on_texts* based on the Keras (Chollet et al., 2015) method to embed the strings into integer sequences. As the models would need all sequences to be consistent dimensionally, we further used zero padding to equate the lengths of all the sequences. We used masking for the multi-head attention mechanism to eliminate attention on the padded zero values, as they do not explicitly contribute to the model's predictions.

6.7 Results

6.7.1 Evaluation of Stage 1

The results obtained through objective evaluation of Stage 1 based on the percentage of alarm types which were correctly predicted are shown in Table 6.1, along with the performance metrics in terms of average precision, recall and F1 score as well as the total computation time. Our models were able to learn reasonable feature representations of the SCADA data, and thereby generate alarm messages corresponding to expert judgements in the historically labelled data.

Table 6.1 Stage 1 Results in terms of performance metrics for predicted alarm types and computation time. The best performing model is shown in bold-face.

Stage 1 Model	Percentage of Alarms Correctly Predicted	Computation time	Avg. Precision	Avg. Recall	Avg. F1 Score
Seq2Seq	78%	**1.42 min**	0.847	0.853	0.850
Seq2Seq (Att)	79.2%	4.25 min	0.862	0.853	0.857
Transformer	**96.76%**	3.30 min	**0.967**	**0.99**	**0.978**

Objective Evaluation

Based on the percentage of alarm types which are predicted correctly by the models, it was seen that the **Transformer** clearly outperformed the other two models. Also, the Transformer attained the highest F1 score (up to 0.978) in comparison to the baseline models, which have very similar performance with the **Seq2Seq** models achieving a slightly better score. This is likely caused due to the relatively short length of the messages in our data – 5.49 symbols on average, consisting of a maximum of 14 symbols in the training dataset and a minimum length of 1 symbol. This shows that the attention mechanism is likely not integral to learn representations corresponding to shorter sequences.

It is also seen (unsurprisingly), that the **Seq2Seq** model achieves the shortest computation time, with the **Transformer** being the second fastest performing model. These computation times were obtained on the NVIDIA Tesla K80 GPU on Google's Compute Engine.

6.7.2 Subjective Evaluation

To confirm our objective metrics for Stage 1, we also conducted a human rating study via Amazon Mechanical Turk (AMT). This was done to assess the semantic correctness of our generated alarm messages as well as their fluency. We decided to evaluate the former metric in terms of *semantic similarity* and asked humans to assign ratings on a 1-5 Likert scale, where 1 means *not similar at all* and 5 means *identical*. Similarly, we used a 1-5 Likert scale to assess *fluency*, where 1 denotes *not fluent at all* and 5 means *human fluency*. We generated 200 random messages from the test set for each model to assess the generated alarm messages. They were assessed by 87 unique human judges leading to a total of 1,200 messages.

Table 6.2 Results in terms of BLEU score with standard deviation, human ratings (median ratings are shown in parentheses) and computation time for all three models. The best performing model is shown in bold-face.

Model	**BLEU-4**	**Semantic similarity**	**Fluency**	**Computation time**
Seq2Seq	0.454 ± 0.220	3.65 (4)	3.14 (3)	**1.42 min**
Seq2Seq (Att)	0.443 ± 0.226	3.59 (4)	3.17 (3)	4.25 min
Transformer	**0.492 ± 0.196**	**3.96 (4)**	**3.36 (4)**	3.30 min

We can observe from Table 6.2 that the **Transformer** clearly outperforms its baselines in terms of semantic similarity, therefore arguably generating the most correct alarm messages overall. It also scores highest on the fluency metric. Further, the BLEU ranking for the baselines is confirmed with **Seq2Seq** being ranked higher for *semantic similarity* than **Seq2Seq (Att)**. The models scored similarly in terms of *fluency*, with **Seq2Seq (Att)** slightly ahead. All models perform similarly in terms of their median rating.

Table 6.3 Examples of alarm messages generated in Stage 1 along with remarks pertaining to their viability. Clearly, the Transformer model outperforms other baselines in terms of all metrics other than computation time.

Reference messages	1. (DEMOTED) Gearbox oil tank 2 level shutdown. **Alarm Code: 905** 2. Wind direction transducer error 1&3. **Alarm Code: 912** 3. Sub pitch priv fatal error has occurred more than 3 times in 86,400 Seconds. **Alarm Code: 902**
Seq2Seq	1. Demoted oil tank shutdown. **Alarm Code: 905** 2. Wind direction transducer error . **Alarm Code: 912** 3. Error occurred more than fatal in $NUM seconds.**Alarm Code: 919**
Seq2Seq (Att)	1. Demoted gearbox oil tank under pressure full brake. **Alarm Code: 901** 2. Pitch system fatal error. **Alarm Code: 919** 3. Sub pitch priv critical error has occurred more than $NUM times in $NUM seconds. **Alarm Code: 925**
Transformer	1. Demoted gearbox oil tank level shutdown. **Alarm Code: 905** 2. Wind direction transducer error .**Alarm Code: 912** 3. Sub pitch priv fatal error has occurred more than $NUM times in $NUM seconds. **Alarm Code: 902**

Analysis of errors and outputs for Stage 1

The example messages generated by each of the models are shown in Table 6.3, along with the corresponding human references. It can be seen that for the **Seq2Seq** model, while there are reasonable messages output for the two initial situations, despite correctly identifying an error in the last situation, the model's output is not coherent. For the **Seq2Seq (Att)** model's generated outputs, the first example has an incorrect reference to the faulty sub-component wherein the oil tank is confused with the yaw brake. In case of the second example, there is a similar error, wherein, the message confuses the pitch system with the wind direction transducer. The final message shown is acceptable in terms of its relevance, but the error is highlighted as “critical" instead of “fatal". Finally, it can clearly be seen that the **Transformer** generates the most correct messages for the problem task, but there are still nuances from human references which the model misses out on, which includes the exact tank which is shut down as well as the relevant alarm codes.

To further analyse and inspect each of the model's behaviour during the prediction process, the attention scores of the top-10 relevant features for an example alarm in the gearbox (*"demoted gearbox oil tank level shutdown"*) are shown in Figures 6.8 and 6.9.

As can be seen, the **Transformer** infers the *GBoxOpShaftBearingTemp1_Max* and *GBoxOpShaftBearingTemp1_Min* SCADA features to be amongst the most important contributing

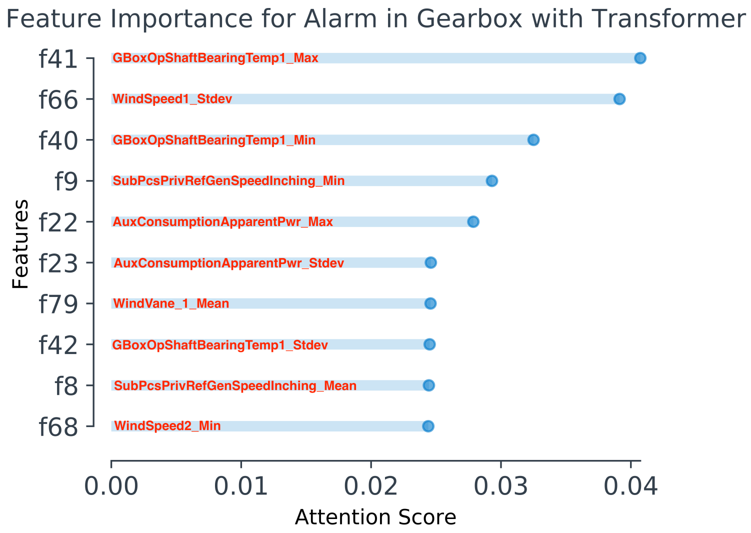


Fig. 6.8 Feature importance for relevant contributing features as obtained using attention weights of the **Transformer** model

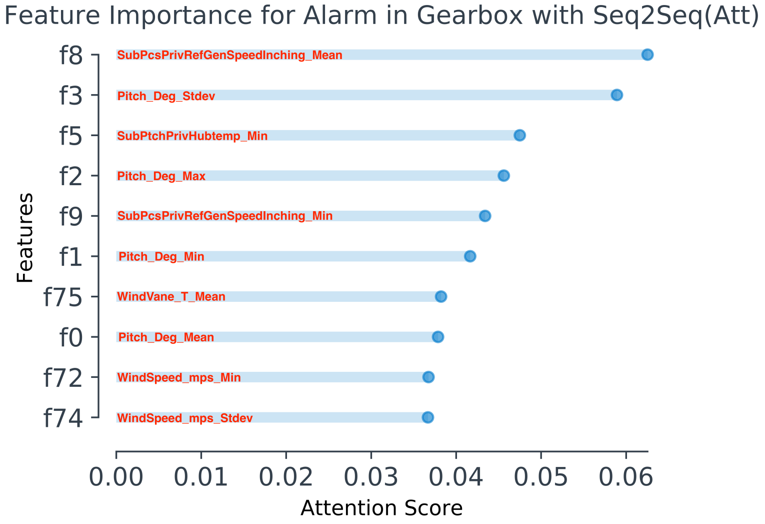


Fig. 6.9 Feature importance for relevant contributing features as obtained using attention weights of the **Seq2Seq (Att)** model

factors, likely attributed to the overheating of the high speed gearbox shaft bearings and the gearbox housing. This leads to the alarm towards shutting down the oil tank owing to an irregular increase in the gearbox temperature. Also, the other higher ranked features such as *SubPcsPrivRefGenSpeedInching_Min* and *AuxConsumptionApparentPwr_Max* are found to be significantly relevant as per existing literature (Feng et al., 2013), as rotational speed and apparent power are vital parameters which can indirectly affect the efficiency of the turbine's control system, and thereby cause an anomaly in the gearbox's operation.

In contrast, the **Seq2Seq (Att)** model's attention scores provide a fairly reasonable interpretation of the features contributing to the gearbox alarm. However, clearly, the features are not as relevant as those identified by the **Transformer** model. Note that *SubPcsPrivRefGenSpeedInching_Mean* specifically signifies an issue pertaining to the high speed generator inching, which occurs as an indirect consequence owing to contact with the generator. However, the model misses some of the key features which are relevant and integral to the operational status of the gearbox. There are additionally some higher ranked features such as *Pitch_Deg_Stdev* which are incorrectly identified, given that any disorientation arising in the turbine's pitch angle generally has little relevance and relationship with operation of the gearbox.

6.7.3 Stage 2: Content-selection of maintenance strategies

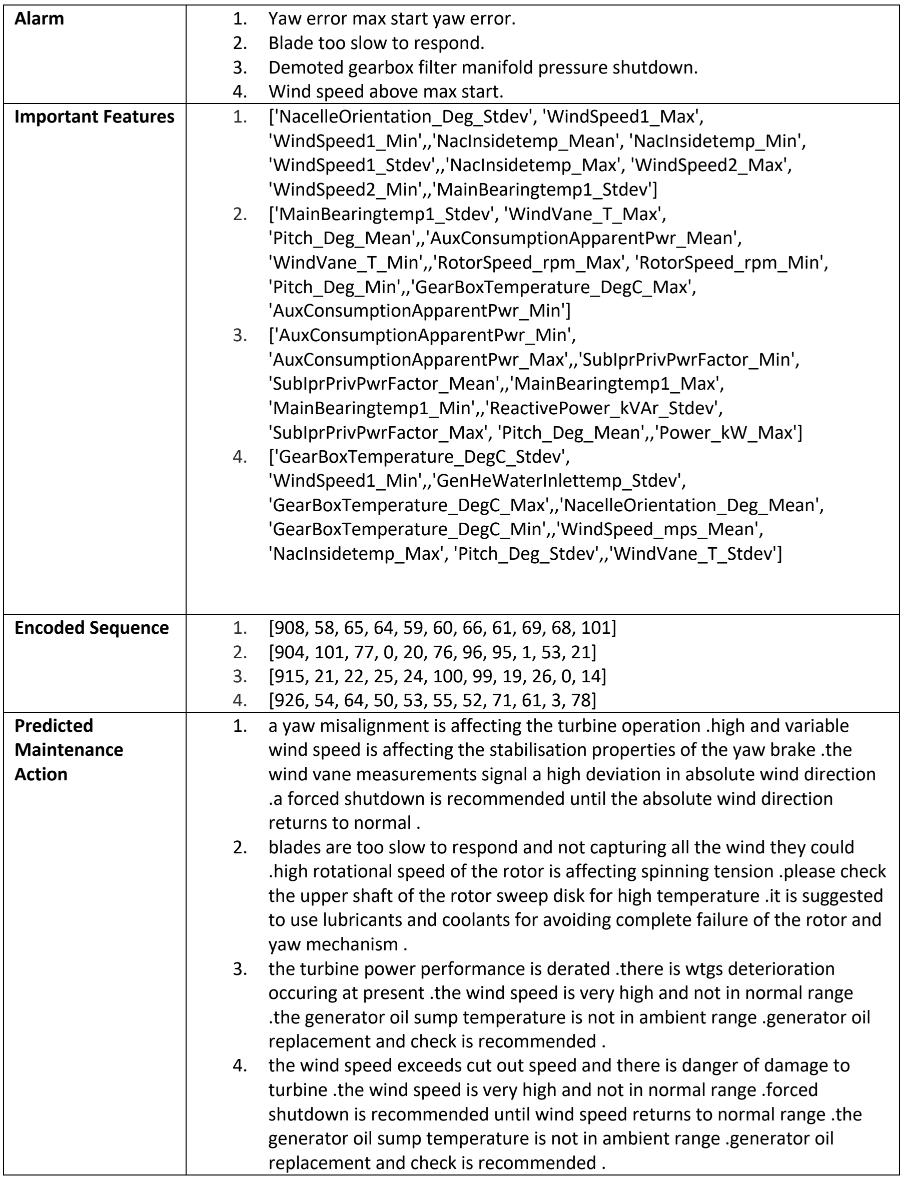
For performance evaluation of the Stage 2 model which plays a key role in predicting the most effective maintenance actions to fix/avert faults in the turbine, 20% (211 samples) are held out as the test set from the over-sampled Stage 2 data set (consisting of 1,055 samples). The Stage 2 Transformer model is accordingly compared against the Seq2Seq(Att) baseline model. For a given sequence of integer-encoded alarm type and the relevant contributing features, the Stage 2 model is evaluated on the basis of the number of maintenance actions correctly chosen for each of the held out samples.

Table 6.4 Results for Stage 2 evaluation in terms of the percentage accuracy of maintenance actions correctly predicted. The model with best performance metrics is outlined in bold-face.

Stage 2 Model	Percentage of maintenance actions correctly predicted
Seq2Seq (Att)	57.34%
Transformer	**75.35%**

The model's final output at Stage 2 consists of a collection of multiple natural language templates which were selected based on the alarm type and contributing features predicted at Stage 1. The results for the Stage 2 model evaluation are outlined in Table 6.4. As is clearly evident, the Transformer model achieves the best performance, predicting correct maintenance actions with an accuracy of up to 75.35% and outperforms the Seq2Seq(Att) model by up to 18%.

Table 6.5 Some examples of maintenance actions predicted by the Dual-Transformer model – for each predicted alarm type and important features identified by Transformer 1, the Transformer 2 model selects appropriate maintenance actions from our human-authored corpus of O&M strategies.



Alarm	1. Yaw error max start yaw error. 2. Blade too slow to respond. 3. Demoted gearbox filter manifold pressure shutdown. 4. Wind speed above max start.
Important Features	1. ['NacelleOrientation_Deg_Stdev', 'WindSpeed1_Max', 'WindSpeed1_Min',,'NacInsidetemp_Mean', 'NacInsidetemp_Min', 'WindSpeed1_Stdev',,'NacInsidetemp_Max', 'WindSpeed2_Max', 'WindSpeed2_Min',,'MainBearingtemp1_Stdev'] 2. ['MainBearingtemp1_Stdev', 'WindVane_T_Max', 'Pitch_Deg_Mean',,'AuxConsumptionApparentPwr_Mean', 'WindVane_T_Min',,'RotorSpeed_rpm_Max', 'RotorSpeed_rpm_Min', 'Pitch_Deg_Min',,'GearBoxTemperature_DegC_Max', 'AuxConsumptionApparentPwr_Min'] 3. ['AuxConsumptionApparentPwr_Min', 'AuxConsumptionApparentPwr_Max',,'SubIprPrivPwrFactor_Min', 'SubIprPrivPwrFactor_Mean',,'MainBearingtemp1_Max', 'MainBearingtemp1_Min',,'ReactivePower_kVAr_Stdev', 'SubIprPrivPwrFactor_Max', 'Pitch_Deg_Mean',,'Power_kW_Max'] 4. ['GearBoxTemperature_DegC_Stdev', 'WindSpeed1_Min',,'GenHeWaterInlettemp_Stdev', 'GearBoxTemperature_DegC_Max',,'NacelleOrientation_Deg_Mean', 'GearBoxTemperature_DegC_Min',,'WindSpeed_mps_Mean', 'NacInsidetemp_Max', 'Pitch_Deg_Stdev',,'WindVane_T_Stdev']
Encoded Sequence	1. [908, 58, 65, 64, 59, 60, 66, 61, 69, 68, 101] 2. [904, 101, 77, 0, 20, 76, 96, 95, 1, 53, 21] 3. [915, 21, 22, 25, 24, 100, 99, 19, 26, 0, 14] 4. [926, 54, 64, 50, 53, 55, 52, 71, 61, 3, 78]
Predicted Maintenance Action	1. a yaw misalignment is affecting the turbine operation .high and variable wind speed is affecting the stabilisation properties of the yaw brake .the wind vane measurements signal a high deviation in absolute wind direction .a forced shutdown is recommended until the absolute wind direction returns to normal . 2. blades are too slow to respond and not capturing all the wind they could .high rotational speed of the rotor is affecting spinning tension .please check the upper shaft of the rotor sweep disk for high temperature .it is suggested to use lubricants and coolants for avoiding complete failure of the rotor and yaw mechanism . 3. the turbine power performance is derated .there is wtgs deterioration occuring at present .the wind speed is very high and not in normal range .the generator oil sump temperature is not in ambient range .generator oil replacement and check is recommended . 4. the wind speed exceeds cut out speed and there is danger of damage to turbine .the wind speed is very high and not in normal range .forced shutdown is recommended until wind speed returns to normal range .the generator oil sump temperature is not in ambient range .generator oil replacement and check is recommended .

6.7.4 Analysis of Stage 2 errors and outputs

Table 6.5 shows some example cases of maintenance actions which are predicted by the Dual-Transformer model during different scenarios of faults. It can clearly be seen that all output messages are grammatically coherent and fluent in nature, which is owed to the fact that instead of requiring the learner's ability to generate long sub-phrases, the second stage Transformer needs to simply perform content selection of the relevant and most-appropriate sub-phrases from the corpus. This makes the learning process both simpler as well as more effective and computationally efficient. However, note that there were some cases for which the model provided false representations of actions which are either not relevant to the fault type or are not suitable to overcome the fault. Additionally, the relevance of the output messages is affected by incorrect ordering of maintenance actions extracted during the content selection process. Below, some interesting cases from Table 6.5 are discussed:-

- As is evident from the first example, there is a fault occurring due to yaw misalignment in the turbine. On the basis of the relevant features contributing to the fault which are identified by the Stage 1 Transformer, the model is able to make effective predictions regarding the fault to be likely attributed to extreme variations in wind speed, which directly causes an inconsistency in the yaw brake's operation. Finally, the model recommends a forced shutdown.

- In case of the second example, the fault is attributed to a slow response of the turbine's blades. Based on the relevant features identified by the Stage 1 Transformer, utilisation of lubricants is suggested towards stabilising the bearings. The relevance of the above action is reasonable in terms of rotor speed and shaft temperature being the key contributing features. As the rotor hub holds the blades of the turbine for ensuring aerodynamic efficiency, any anomalous behaviour and inconsistency occurring in the blade movements can most likely be attributed to a difficulty in lifting the turbine blades for making the rotor spin.

- As evident from the third example, the model completely fails to identify both the list of relevant contributing features as well as the appropriate maintenance action. The model makes incorrect predictions for the maintenance strategies on the basis of derated turbine power performance, while the alarm actually occurred in the gearbox and is irrelevant in the context of this case.

- For the final example, the message outlines that the fault occurs due to the wind speed value falling above the maximum cut-off speed limit (which is necessary in order to sustain the turbine operation). The model correctly recommends forced shutdown of the turbine until wind speed returns to normal as an action. However, in addition, note that the model outputs an irrelevant sub-phrase targeted at the generator oil sump temperature. We believe that this is likely caused due to close association across the feature indices corresponding to wind speed and generator oil sump temperature in the SCADA data, and in this case, the stage 2 model marginally misses the correct template index for wind speed, and instead picks up the nearest template based on positioning order in the predicted sequence.

It is therefore clearly evident that while the Dual-Transformer model is very effective in recommending appropriate maintenance actions to be taken by turbine engineers and technicians, it fails to identify the most relevant actions on some occasions which correspond to a given sequence of SCADA features and the prevailing alarm type. While this may not be sufficient to achieve real-world decision support in the wind industry, the results obtained are nevertheless promising as turbines are complex engineering systems for which automated decision making through NLG is still in the stages of its infancy. We believe that the proposed technique can be improved in the future by incorporating larger datasets with better annotations of human-authored maintenance actions and performing dynamic optimisation by incorporating human feedback into the model's decisions.

It is thereby integral for the content selection stage of the model to eliminate (in some cases) the non-relevant sub-phrases predicted in the output, and only provide the most appropriate sub-phrases to be more effective in performance. This can be possibly achieved in the near future through incorporation of maintenance manuals alongside other unstructured documents in the learning process.

6.8 Conclusion

In this chapter, the feasibility and promise of applying data-to-text generation techniques to support O&M in the wind industry has been demonstrated. The proposed NLG approach for decision support can assist turbine engineers and technicians to better understand the context of impending faults, and thereby, potentially prevent a catastrophe caused by expensive events and alarms. Based on experiments in which the Transformer is compared against Seq2Seq model with and without attention, it is seen that the Transformer significantly outperforms the other models in terms of performance and computation speed, making it highly promising for real-time industrial applications. Most importantly, the attention scores of the Transformer are significantly more aligned to expert judgements based on existing literature in the wind industry compared to the Seq2Seq and Seq2Seq(Att) models. The second stage Transformer model provides the valuable capability to predict effective maintenance actions on the basis of likely causes of failures identified in the first stage. A brief video which explains this approach is included with this chapter [2]. Note that while the Dual-Transformer learns reasonable feature representations in most cases, a variety of meaningful numbers and symbols are lost during the generation of alarm messages. This problem can possibly be addressed by using Pointer networks or copy actions in the future. Note that a key drawback of the proposed technique is the possibility for the model to identify factually incorrect maintenance actions during content selection, which we aim to address in the next chapter. In Chapter 7, we will focus on further refinement in the quality of the predicted maintenance actions by utilising multimodal domain-specific information in the wind industry to develop an interactive question-answering system based on Knowledge Graphs (KGs), providing engineers and technicians an ambient interface to query O&M facts in natural language.

[2] Brief explanation of Dual-Transformer model for predicting alarm messages and maintenance actions: https://youtu.be/HSUFzBr_mVQ

Chapter 7

Automated Question-Answering for Interactive Decision Support

> Part of being successful is about asking questions and listening to the answers.
>
> Anne Burrel

This chapter is based on a preprint (Chatterjee and Dethlefs, 2021a) and an article (Chatterjee and Dethlefs, n.d.) presently in submission to a journal.

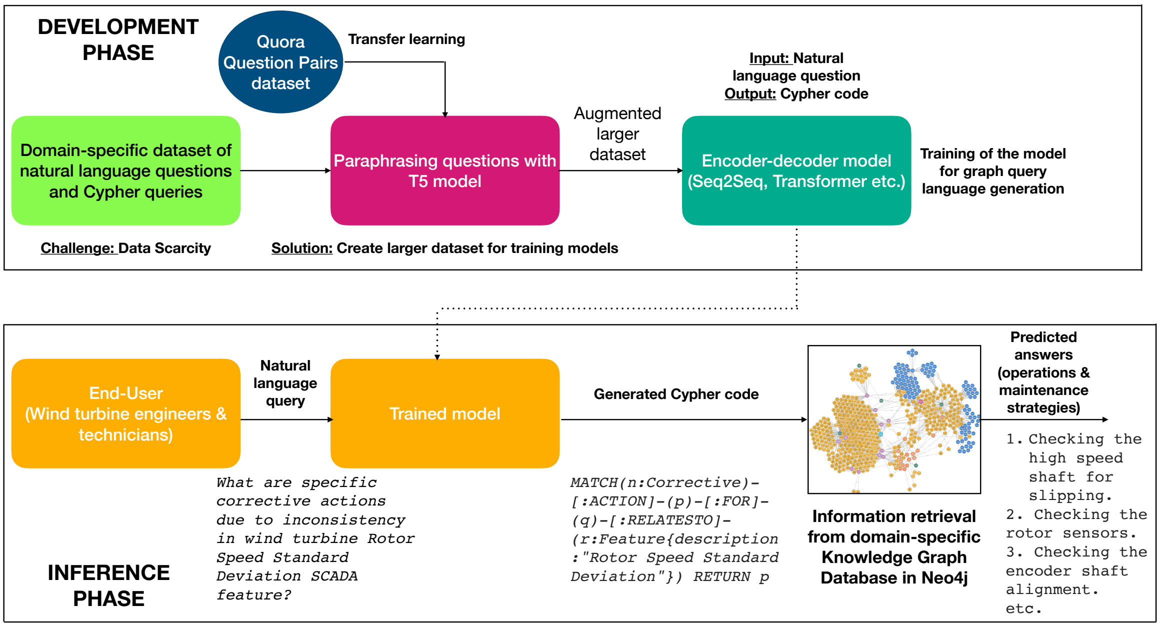
7.1 Introduction

As outlined in the previous chapters, despite demonstrating promising applications of AI in the wind industry, existing studies fail to provide an ambient interface for automated reasoning during CBM. As *decision making time* is a critical factor for O&M activities, an automated reasoning system can provide immediate answers to engineers on *why* a fault occurs for instance, and *how* to fix/avert the failure by planning for appropriate management of SCADA parameters, alarms and condition of sub-components etc., leading to significant savings in O&M costs and reduction of downtimes and operational inconsistencies. While the previous chapter has focused on leveraging NLG techniques for generating brief natural language phrases (alarm messages and maintenance action strategies), the approach cannot provide an ambient interface for interactive reasoning and generation of more informative and comprehensive O&M reports. To accomplish this objective in this chapter, we propose automated question-answering over Knowledge Graphs (KGs) for facilitating interactive decision support during O&M. Specifically, we aim to develop an interactive system wherein, natural language questions posed by the engineers e.g. "*What are the predictive activities for the power cabinet of wind turbines?*" can be effectively answered by appropriate retrieval of facts pertaining to the relevant entities – which in this case would describe the "*Predictive Activities*" property for the entity "*Power Cabinet*".

Question-Answering (QA) systems (Buck et al., 2018; Budiharto et al., 2020; Chakravarti et al., 2020; Dwivedi and Singh, 2013; Hien et al., 2020; Hong et al., 2019; Park et al., 2015; Srihari and Li, 2000; Voorhees, 2001) provide intelligence analysts and other users of information systems with the ability to pose questions to a system in natural language (*semantic queries*), and obtain the relevant answers (*assistance*) they may require to better perform their tasks (Liu et al., 2013; Small et al., 2003; Xie et al., 2015; Zafar et al., 2020). QA systems have historically been successfully utilised for information retrieval in domains like e-commerce (Yu et al., 2018), education (Elnozahy et al., 2019), tourism (Ou et al., 2008) and safety-critical applications like healthcare (Daniel et al., 2019; Goodwin and Harabagiu, 2016; Saibene et al., 2021) etc.

QA systems leverage either pre-structured databases or a collection of domain-specific natural language documents for information retrieval (Calijorne Soares and Parreiras, 2020). In the last decade, there has been a particularly growing interest in leveraging Knowledge Graphs (KGs) for QA (Berant et al., 2013), wherein, various real-world entities like concepts, events, objects etc. are represented as nodes in the graph and these are interlinked via graph edges that serve as a predicate (Vegupatti et al., 2020a). KGs can either be open-domain or domain-specific – open-domain KGs (e.g. Google KG) consist of a large collection of coarse-grained facts without being restricted to a specific domain, whereas, domain-specific KGs (e.g. in healthcare) consist of facts dedicated to specific application domains and are generally smaller in size (Vegupatti et al., 2020b). While there has been a significant emphasis on QA in open-domain tasks, there has been limited focus on leveraging domain-specific KGs in applications aimed at AI for social good, especially pertaining to tackling climate change (e.g. towards helping make wind energy sources more reliable), as enunciated before.

There has been a significant research focus on training AI models for QA over KG databases across multiple application areas, particularly for open-domain tasks (Bordes et al., 2015; He and Golub, 2016; Vegupatti et al., 2020b) by utilising natural language questions and answers that are either human-annotated or (semi)-automatically generated. Most existing studies have focused on directly generating answers in natural language itself, rather than the code which can be used to retrieve the answer from a KG. While this is an important research avenue in its own right, when applying QA to O&M of renewable energy sources in a safety-critical application (such as wind turbine maintenance), we have a need


DEVELOPMENT PHASE
Quora Question Pairs dataset
Transfer learning
Input: Natural language question
Output: Cypher code
Domain-specific dataset of natural language questions and Cypher queries
Paraphrasing questions with T5 model
Augmented larger dataset
Encoder-decoder model (Seq2Seq, Transformer etc.)
Training of the model for graph query language generation
Challenge: Data Scarcity
Solution: Create larger dataset for training models
End-User (Wind turbine engineers & technicians)
Natural language query
Trained model
Generated Cypher code
Predicted answers (operations & maintenance strategies)
Information retrieval from domain-specific Knowledge Graph Database in Neo4j
INFERENCE PHASE

What are specific corrective actions due to inconsistency in wind turbine Rotor Speed Standard Deviation SCADA feature?

```
MATCH(n:Corrective)-
[:ACTION]-(p)-[:FOR]-
(q)-[:RELATESTO]-
(r:Feature{description
:"Rotor Speed Standard
Deviation"}) RETURN p
```

```
1. Checking the
   high speed
   shaft for
   slipping.
2. Checking the
rotor sensors.
3. Checking the
encoder shaft
alignment.
etc.
```

Fig. 7.1 Proposed framework for graph query language generation in the wind industry, wherein, an encoder-decoder model is used to predict Cypher queries based on domain-specific natural language English questions. A pre-trained T5 Transformer model is utilised for data augmentation through paraphrasing.

to be exact as errors in response/answer generation can have serious consequences. We therefore opt for an approach that generates KG queries instead of direct natural language. This avoids accounting for the variability of natural language and reduces the required vocabulary (including the presence of special symbols, numbers and variables etc.). In this regard, our approach is related to work in code (Cummins et al., 2017; Feng et al., 2020b; Kusupati et al., 2018; Yin and Neubig, 2017) and formal language generation (Affolter et al., 2019; Liang et al., 2021; Singh et al., 2020). The retrieved answers in our study are related to look-up of information from specialised domain resources such as user manuals for wind turbine maintenance. The proposed approach also helps the QA system to incorporate multimodal information like images of turbine-components, SCADA features etc. in the responses, instead of learning to generate such entities themselves.

In this chapter, we propose a novel solution to multimodal decision support using a Transformer model (Vaswani et al., 2017a). We utilise a domain-specific KG (Chatterjee and Dethlefs, 2021a) that we developed from human-authored maintenance manuals in the wind industry to ensure high-quality outputs. We also employ paraphrase generation for data augmentation to account for the potential variability in natural language input queries. Figure 7.1 depicts a conceptual overview of our proposed framework. Experiments with an attention-based Sequence-to-Sequence (Seq2Seq) RNN model and a Transformer for graph query language generation for information retrieval from the KG show that the Transformer model predicts queries (and thereby the responses to natural language questions) with an accuracy of up to 89.75%.

In summary, we make the following key contributions in this chapter:-

1. A novel framework for graph query language generation is proposed in a real-world application of intelligent QA systems for interactive decision support in O&M of wind turbines.

2. We develop a domain-specific dataset of natural language questions and Cypher queries for QA over a publicly available domain-specific Neo4j KG database in the wind industry. All our data is made publicly available on GitHub[1].

3. We explore the role of pre-trained large language models for performing data augmentation of domain-specific information in the wind industry through paraphrase generation.

4. Our approach provides a complete QA system for engineers and technicians in the wind industry to automatically query domain-specific information in natural language, without requiring any specialised skills and understanding of KGs. This can potentially help reduce operational inconsistencies by assisting engineers to fix/avert failures in wind turbines in a timely manner, thereby helping make wind energy sources more reliable en-route to combat climate change.

7.2 Related Work

We will review related work in QA for open and closed domains and also in the area of formal language generation.

7.2.1 Question-Answer Generation

Domain-specific QA tasks have witnessed extensive research interest over the years, particularly in the last decade (Clarke et al., 2010; Kwiatkowksi et al., 2010; Mollá and Vicedo, 2007; Pavlić et al., 2015; Zelle and Mooney, 1996). Most existing studies in this area have utilised a static lexicon to map the surface forms of the relevant entities to their logical

[1] Datasets: https://github.com/joyjitchatterjee/WindTurbine-QAKG

forms (Aghaebrahimian and Jurčíček, 2016), which makes scaling up such lexicons (that often consist of thousands of entities) challenging and inefficient. To tackle these challenges, KGs have been utilised for QA through development of *Natural Language Interfaces to Knowledge Bases (NLIKB)* (Affolter et al., 2019; Habernal and Konopík, 2013; Han et al., 2016; Paredes-Valverde et al., 2015; Rohil et al., 2018). Existing studies performing QA over KGs have focused on utilising either general or neural network-based approaches (Tong et al., 2019).

The general QA approaches have mainly focused on leveraging information retrieval methods and semantic parsing. Information retrieval-based approaches (Bordes et al., 2015; Yao and Van Durme, 2014) analyse the dependency of words in questions to construct a candidate set consisting of possible answers retrieved from the KG, from which the most appropriate information is selected based on a quality or relevance evaluation (Tong et al., 2019). In contrast, the studies that leverage semantic parsing (Berant and Liang, 2014; Cai and Yates, 2013; Kwiatkowski et al., 2013; Liang et al., 2013; Reddy et al., 2014) map natural language questions to their logical forms based on either a combinatory grammar mechanism or dependency-based compositional semantics (Tong et al., 2019). These logical queries are then finally translated into structured queries for extracting relevant answers from the KGs (Mohammed et al., 2018). Note that such structured queries (e.g. SPARQL, Cypher etc.) are essentially domain-specific code, which are executable by computers for retrieval of relevant information from the KG databases. The task of generating these queries during data-driven semantic parsing thereby pertains to *Domain-Specific Language (DSL)*[2] generation, wherein, formalisms of the appropriate schema and syntax of code are to be learnt from the data. While the general QA approaches are simple to apply, they generally witness low evaluation scores and have mostly been outperformed by neural approaches in recent years, particularly when complex multi-hop reasoning over KGs is required.

There has been a rapidly growing interest in leveraging DL models for QA, particularly for open-domain tasks. This progress can be attributed to the release of large Knowledge Bases (KBs) that consist of consolidated knowledge extracted from various sources e.g. free-form text, tables etc. (such as *Freebase*) or annotated by humans (such as *SimpleQuestions*, *WebQuestions* etc.) (Bordes et al., 2015). In a notable study in this domain (Bordes et al., 2015), Memory Networks (MemNNs) have been utilised to achieve state of the art

[2] Domain-Specific Languages have code which is easy and intuitive for a specific application domain e.g. HTML for web development, SQL for querying relational databases etc. DSLs have high-level abstractions to reduce focus on overcoming low-level challenges in programming (Portugal et al., 2016).

performance in simple QA[3] on the WebQuestions database. MemNNs contain a memory component which can be read from or written to, along with a trainable neural network which can be used to query the memory (that can be incorporated with facts from KG databases) for appropriate information retrieval. The authors utilised the (*Freebase*) database as the underlying KG for the MemNN model. The paper also proposed a new *SimpleQuestions* structured KG database in this study, and demonstrated that the model can facilitate transfer learning – thereby making high-quality predictions in a new domain (with the Reverb database) without needing to be re-trained.

There have been some other popular studies in this domain that apply various types of neural network architectures for QA: a character-level encoder-decoder model with attention (He and Golub, 2016) has been used to learn higher level semantic concepts towards answering natural language questions from KG databases. The paper demonstrated that the model, which leverages attention-based Long Short-Term Memory Networks (LSTMs) for encoding and decoding, facilitates joint learning of question embeddings, entities and predicates which enables it to better generalise to unseen entities. The proposed model significantly improves the state of the art performance on the *SimpleQuestions* database, while utilising significantly less data for training in comparison to previous work. Some studies (Lukovnikov et al., 2017) have utilised Recurrent Neural Networks (RNNs) in an end-to-end manner for simple QA tasks. These studies have shown that RNNs are able to effectively model the sequential nature of the natural language questions and answers in KG databases by transforming the inputs into vectors using word or character level embeddings (Vegupatti et al., 2020b). However, the process of modelling large amounts of sequential data significantly increases the training time in such models, and additionally, these models are less efficient in learning long sequences in comparison to more recent neural architectures like Transformers.

Attention-based Convolutional Neural Networks (CNNs) have also been explored for a similar task in answering factual questions (Yin et al., 2016), wherein, the authors show that stacking an attentive max-pooling layer over the regular convolution layers in the CNN can help to model relationships between predicates and question patterns more effectively. More recently, Transformer-based models have been shown to outperform conventional end-to-end neural architectures in QA tasks (Kacupaj et al., 2021b; Lukovnikov et al., 2020; Phan and Do, 2020; Plepi et al., 2021). As Transformers eliminate recurrence prevalent in vanilla RNNs and utilise a self-attention mechanism instead, they are generally able to learn more

[3] In simple QA tasks, a specific fact from the KG database would answer the user's question without requiring multi-hop reasoning over multiple facts in the KG.

effectively over longer and more complex sequences of text (questions and answers) while also being computationally more efficient due to their parallelisation capabilities.

7.2.2 Automatic code and formal language generation

Recently, there has been a growing interest in applying DL to the automatic generation of high-level general purpose programming languages (such as Python, Java etc.) (Cummins et al., 2017; Feng et al., 2020b; Kusupati et al., 2018; Yin and Neubig, 2017) as a means to automatically find programs that satisfy certain criteria such as efficiency, optimality, correctness, hardware compatibility etc. This has led to a number of investigations that combine the traditionally separate fields of programming languages and NLP. While some studies aim to generate directly from language to code (or vice versa), others make use of a formal intermediary representation, such as an Abstract Syntax Tree (AST). Interestingly, the previous work on applying NLP techniques to programming code has mostly focused on the generation of small descriptions of code fragments, summaries and annotations, see e.g. the study by Haiduc et al. (2010) for an early approach to code summarisation. Another approach to the same task (Iyer et al., 2016) utilises an LSTM Seq2Seq model to learn questions that describe code segments from a corpus collected from StackOverflow. The authors attempt to learn a direct mapping from inputs to outputs without any intermediate representations and report low similarity with a human comparison. Hu et al. (2017) replicate Iyer et al. (2016)'s study and show that using an AST as input to their Seq2Seq learner can improve performance. Related to code summary generation, Allamanis et al. (2016) generate method names for code snippets by learning a mapping from long input sequences to short output sequences. The authors use an LSTM and extend it with a convolutional layer that acts as a domain-invariant attention mechanism.

In terms of generating code itself, Yin and Neubig (2017) trained a neural network to generate source code from natural language inputs by treating it as a semantic parsing problem. The authors demonstrate the benefit of modelling syntax explicitly and outperform previous work by 9-11%. In a related study, Ling et al. (2016) applied latent predictor networks to generate code for a computer card game. Another study in this domain (Dong and Lapata, 2018) utilises a hierarchical approach for neural semantic parsing across multiple tasks for generating Python source code and SQL queries. The authors generated intermediate logical forms by omitting low-level information in the code (e.g. variable names and arguments) and filling in the missing details by conditioning on the meaning representations of natural language questions. They utilised these for constructing the final logical forms using a simple encoder-decoder model.

More recent studies have focused on utilising Transformers for generating code snippets from natural language descriptions. Kusupati et al. (2018) have utilised Transformers for Python code snippets generation with the CoNaLa dataset, wherein, the authors utilised a modified form of back translation and cycle consistent losses to train the model in an end-to-end manner. They demonstrated that the self-attention based Transformer architecture outperforms LSTM based encoder-decoder models significantly. Large pre-trained language models have also been utilised for code generation with promising results. Feng et al. (2020b) proposed a Bidirectional Encoder Representations from Transformers (BERT) based model (Code-BERT) for generating code in six programming languages (like Python, Java etc.). The authors also explored zero-shot learning and demonstrated that the model achieves state of the art performance in generating the most semantically related code corresponding to natural language questions on the CodeSearchNet corpus, while also achieving promising performance in other downstream NLP tasks like code-documentation generation.

Overall, we observe that code generation is of growing interest to multiple communities. Current work in NLP focuses mostly on the analysis of code and generation of annotations rather than of code itself. Output sequences are mostly short and evaluation scores still low. The programming languages community has started to adopt neural nets but still relies mostly on engineered algorithms with a learnable component rather than fully learnt systems. In comparison to generating code in high-level programming languages, there has been limited research in generating graph query languages for querying KBs.

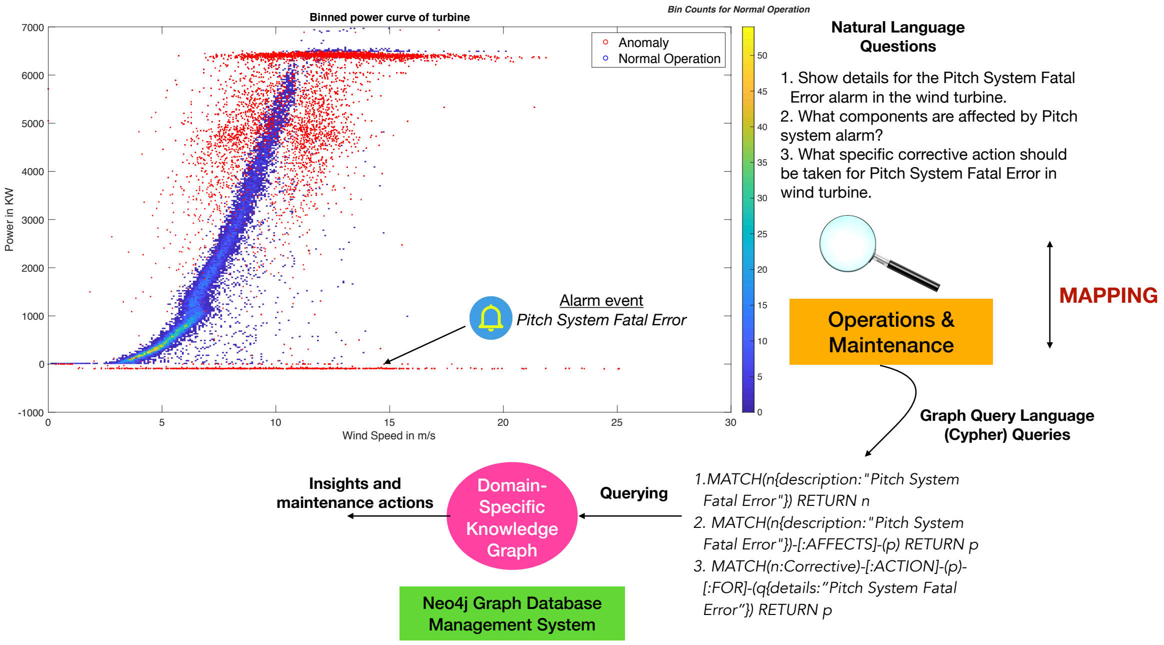
Most early approaches to graph query language generation (Affolter et al., 2019; Singh et al., 2020) have leveraged rule-based, pattern-based or grammar-based approaches to translate natural language questions to formal queries in DSLs like SQL etc. (Liang et al., 2021). In a notable study focusing on graph query language generation (Liang et al., 2021), an ensemble approach has been utilised wherein, a random forest model was used for phrase mapping to identify the question types and a tree-structured LSTM model was used for SPARQL query generation. The LSTM takes into account the syntatic and semantic structure of the input questions and the tree representation of all possible candidate queries for ranking the generated queries, thereby providing the queries which are most appropriate to the questions. The method significantly outperforms the state of the art QA systems on the 7th Question Answering over Linked Data Challenge (QALD-7) and the Large-Scale Complex Question Answering (LC-QuAD) databases. Some other recent studies in the area of formal language generation and KGs have explored ontology reasoning to train

Deep Neural Network models for effectively performing logical reasoning (Hohenecker and Lukasiewicz, 2020), generation of textual summaries from KG triples (Vougiouklis et al., 2020), learning language errors in real-word software programs (Chockler et al., 2020) etc.

While there has been particularly significant research in generating SQL code from natural language questions, there is very limited research in generating Cypher queries for information retrieval in Neo4j database systems. In possibly the only study in this area, a simple yet promising approach has been presented to transform English language questions to Cypher queries for an open-domain dataset used for a university project (Hains et al., 2019). The authors performed tokenization of the natural language string queries for assigning language tags, which were further used to perform systematic pattern matching in extracting relevant labels and variables towards generating Cypher queries. While the approach shows that a useful interface can be built to provide QA over KGs without utilising AI models by leveraging the expressive power of Cypher queries, it cannot generate more complex query patterns and understand the context of user's intentions, which are integral in real-world industrial applications.

A common challenge with most existing studies is the prevalence of small datasets for QA, making it difficult to train AI models. To tackle this problem, synthetic data generation models have been utilised (Alberti et al., 2019), wherein, the authors utilised a fine-tuned BERT model to generate synthetic question & answer pairs. Their study demonstrated that pre-training on synthetic data helps to significantly improve the performance of the QA system on the SQuAD2 and NQ datasets. Some studies (Dong et al., 2017; Hasan et al., 2016; Kacupaj et al., 2021a; Prakash et al., 2016) have utilised paraphrasing for data augmentation to generate synthetic data for QA. In a notable study in this domain (Kacupaj et al., 2021a), the authors experimented with different models (RNN, Transformer and CNN) for generating multiple paraphrase responses for the same questions in the DBpedia KG database. The study showed that paraphrase generation can significantly improve the performance of ML models in QA over KGs, providing a more expressive QA experience. There has been rather limited application of paraphrase generation in safety-critical applications beyond existing open-domain databases, with a notable exception being Hasan et al. (2016), wherein, the authors utilised an attention-based bidirectional RNN model for clinical paraphrase generation with promising results.

Clearly, most existing studies in generating graph query languages have focused on generating relatively simple queries for open-domain tasks, rather than for real-world safety-critical applications which generally require complex reasoning and often witness long sequences of queries with significant lexical and syntactic variations. This is particularly


Binned power curve of turbine
Bin Counts for Normal Operation
Anomaly
Normal Operation
Power in KW
7000
6000
5000
4000
3000
2000
1000
0
-1000
0
5
10
15
20
25
30
Wind Speed in m/s
50
45
40
35
30
25
20
15
10
5
0
Alarm event
Pitch System Fatal Error
Natural Language Questions
MAPPING
Operations & Maintenance
Graph Query Language (Cypher) Queries
Querying
Domain-Specific Knowledge Graph
Insights and maintenance actions
Neo4j Graph Database Management System

1. Show details for the Pitch System Fatal Error alarm in the wind turbine.
2. What components are affected by Pitch system alarm?
3. What specific corrective action should be taken for Pitch System Fatal Error in wind turbine.

```
1.MATCH(n{description:"Pitch System
  Fatal Error"}) RETURN n
2. MATCH(n{description:"Pitch System
  Fatal Error"})-[:AFFECTS]-(p) RETURN p
3. MATCH(n:Corrective)-[:ACTION]-(p)-
  [:FOR]-(q{details:"Pitch System Fatal
  Error"}) RETURN p
```

Fig. 7.2 Framework for information retrieval during anomalies in turbine operation. The Neo4j KG database is leveraged for mapping natural language questions to equivalent Cypher representations.

owed to the fact that the previous studies focusing on generating Cypher queries have not leveraged more recent advances in AI, particularly models like Transformers which have shown success in other domains of automated code generation in high-level programming languages as we have discussed before. Additionally, there is very limited research in leveraging data augmentation techniques such as paraphrasing in real-world applications, particularly for tasks wherein availability of large-scale domain-specific QA corpora is a major challenge, as in the wind industry.

7.3 Framework for information retrieval during QA

We utilise our domain-specific KG for facilitating automated reasoning by providing engineers and technicians with a natural language interface to query the KG in Neo4j[4] (which is the world's leading open source NoSQL graph database management system widely utilised in industry) through Cypher queries (Neo4j's native graph query language) (López and la Cruz, 2015). Given a natural language question, our goal is to generate the appropriate Cypher query to facilitate direct information retrieval of appropriate O&M strategies from

[4] Neo4j graph database management system: https://neo4j.com

the KG database in a fully automatic fashion. The answers in our QA system would be retrieved from the KG by mapping the Cypher queries to the corresponding details of the nodes and properties. While we specifically generate Cypher queries in this chapter, our approach is programming language-agnostic and can potentially be extended to other graph query languages such as SPARQL. Figure 7.2 shows the binned power curve (Chatterjee and Dethlefs, 2020e) reflecting the operational states for the LDT rated at 7MW. As can be seen, when an anomaly occurs in the turbine's sub-components, an alarm is raised (in this example, we have labelled a pitch system alarm which causes shutdown of the turbine operation). The engineers / technicians need to take instantaneous actions to fix the alarm in a timely manner to avoid continued downtimes due to no energy production. Here, the natural language questions posed by the engineers are converted into equivalent Cypher representations. These are then directly used for querying high-quality O&M actions and insights from the Neo4j Graph Database Server by leveraging the domain-specific KG that we utilise in this chapter.

7.4 Dataset Development and Pre-Processing

In this section, we describe the datasets utilised for developing the QA system. We utilise a domain-specific KG for the wind industry domain (Chatterjee and Dethlefs, 2021a)[5] that we have previously developed from scratch based on the LDT SCADA data and alarm logs described in Chapter 3 and the publicly available Skillwind Maintenance Actions Manual (Astiaso Garcia et al., 2020) [6] as the primary source of knowledge for information retrieval during QA. This domain-specific ontology contains 537 nodes and 1,059 relationships (of 9 distinct types), and includes various types of heterogeneous information such as descriptions of alarms, SCADA parameters, preventive/predictive/corrective O&M strategies, images of turbine sub-components etc. The complete details for all nodes in the KG alongside brief descriptions for their utility and the available properties are shown in Appendix B. Note that this KG, while being a valuable source of O&M information cannot be directly utilised for QA as it does not have any labelled templates or relationships for automated reasoning based on natural language questions. To facilitate QA, we develop a specialised domain-specific corpus of natural language questions and Cypher query pairs as described below.

[5] XAI4Wind Knowledge Graph: http://github.com/joyjitchatterjee/XAI4Wind

[6] Skillwind Maintenance Manual: https://skillwind.com/wp-content/uploads/2017/08/SKILWIND_Maintenance_1.0.pdf

7.4.1 Creation of natural language questions - Cypher query pairs of domain-specific templates

Initially, we manually created 93 pairs of domain-specific natural language questions in English and the corresponding Cypher queries required to extract the relevant answers from the Neo4j KG database. These templates are generic i.e. they do not represent the O&M actions etc. for any particular sub-component, fault type etc., but the same query can represent the relevant answers for different types of cases. As the data from wind turbines consist of several (often hundreds) of sub-components, SCADA features, alarms etc., it can be highly complex and time-consuming to manually create such natural language question-Cypher query (QC) pairs for each of the cases – thereby, we used wildcards containing specific tags (e.g. <*fevent-details*> for details of all fault types), which can later be automatically replaced with the corresponding names of the sub-components, details of the faults etc. present in the KG to develop unique QC pairs for each case. Note that in some cases, these tags were not created wherein, the QC pairs are unique and do not change across different node labels or their properties in the KG. For instance, for the question *"What are the main components of the system of the wind turbine?"*, there is only one unique Cypher query:-

```
MATCH(n:System)-[:CONTAINS]-(p) RETURN p
```

Here, the query signifies that all matched nodes are to be retrieved from the KG database which are related to the *SYSTEM* node with a *CONTAINS* relationship, thereby pointing to the various sub-components of the wind turbine (which is considered as a system in this chapter).

Table 7.1 Details of some tags used in developing QC templates, along with example QC pairs

Tag	**Description**	**Example Question (Q)**	**Example Cypher Query (C)**
<*subsys-name*>	Names of 13 different types of turbine sub-systems e.g. *Foundation & Concrete Section*, *Electric, Sensor & Control* etc.	Provide general predictive activities for <*subsys-name*> of wind turbine.	*MATCH(n{name:"<subsys-name>"}) RETURN n.PredictiveActivities*

<*scada description*>	Detailed descriptions for 102 different SCADA features e.g. *Pitch Angle Mean Value*, *Active Power Mean Value* etc.	Show specific corrective actions relating <*scadadescription*> wind turbine SCADA feature.	*MATCH(n:Corrective)-[:ACTION]-(p)-[:FOR]-(q)-[:RELATESTO]-(r:Feature{description:"<scada description >"}) RETURN p*

Table 7.1 describes some example tags which were created, alongside their descriptions. Example questions along with the relevant Cypher queries are also shown. More details on all other tags are provided in Appendix C.

7.4.2 Converting the generic templates to specific QC pairs

Next, the wildcard tags in the 93 generic templates were replaced with the corresponding node labels or their attribute details, both in the natural language questions as well as the Cypher queries. This was done automatically based on the available details in the KG database, wherein e.g. <*subsys-name*> in the QA pairs was replaced with the names of all subsystems in the turbine, <*scadadescription*> was replaced with the detailed description of all SCADA features etc. as previously discussed in Table 7.1. This led us to a total of 2,361 unique QC pairs, wherein, for each natural language question, there is an associated Cypher query which extracts the relevant answers from the KG to facilitate decision support. Table 7.2 describes some examples of the obtained QC pairs after the replacement of the wildcard tags, along with the extracted answers from the KG.

7.4.3 Data augmentation through paraphrasing

QA tasks are generally challenging due to the fact that there are many different ways in which natural language questions can express the same information need, given that there can be a large variety of surface forms pertaining to semantically equivalent expressions (Dong et al., 2017). For instance, it is integral for a QA system in the wind industry to recognise that the natural language questions "*What are some important details for the power cabinet subsystem of the wind turbine?*" and "*What are the fundamental features of power cabinet subsystem of wind turbine?*" are completely similar in meaning despite the subtle lexical

Table 7.2 Examples of natural language questions and Cypher queries, along with extracted answers from the KG

Natural language question (Q)	**Cypher query (C)**	**Extracted answer examples from the KG (A)**
What fault events affect the wind turbine Transformer subsystem?	*MATCH(n:FaultEvents)-[:TYPE]-(p)-[:AFFECTS]-(q{name:"Transformer"}) RETURN p*	1. UPS failure. 2. Communication error. 3. Transformer fan circuit breaker. The thermal fuse of the transformer fan is deactivated. 4. Temperature measurement module failure. Error in Temperature card (RTD).
What fault events are caused in the wind turbine by the Pitch Angle Mean Value SCADA feature?	*MATCH(n:Feature{description: "Pitch Angle Mean Value"})-[:RELATESTO]-(p) RETURN p*	1. Possible existence of ice on blades. 2. Blade Position Error. 3. Pitch Activation Error. If the pitch of any of the blades differs some degrees from the reference pitch for a short period. 4. Low pitch value in stop. The position of any of the 3 blades is less than 80º in STOP mode for a specific time.
What are the specific corrective actions for High temperature on the gearbox bearing fault event in the wind turbine?	*MATCH(n:Corrective)-[:ACTION]-(p)-[:FOR]-(q{details:"High temperature on the gearbox bearing"}) RETURN p*	1. Looking for bearing damages. 2. Making sure the bearing temperature is higher than the oil one (if not wiring could be swapped). 3. Checking the multiplier pump / cooling units. 4. Checking the wiring and cables for damages.

variations in these expressions. As using only a small dataset of 2,361 QC pairs described in Section 7.4.2 for developing the QA system would significantly hamper the generalisability (Gan and Ng, 2019) and the system's ability to consider wider contextual information in the questions (Dong et al., 2017), it is integral to perform data augmentation towards generating a larger dataset.

A popular technique for data augmentation in developing QA systems is paraphrase[7] generation (Egonmwan and Chali, 2019; Fu et al., 2019; Gan and Ng, 2019; Li et al., 2018; McKeown, 1983), wherein, texts with identical meanings are expressed in different ways for creating variations in queries (Qian et al., 2019). This helps to narrow down the gap prevailing between the natural language questions queried by the user and the system comprehension,

[7] Paraphrases are defined as sentences which convey the same meaning, but have different surface realization. (Fu et al., 2019)

increasing the likelihood of finding answers to the user's questions (Qian et al., 2019; Rinaldi et al., 2003).

Traditionally, paraphrase generation has mostly utilised rule based techniques (McKeown, 1983), which perform lexical substitutions of content words (Fu et al., 2019) by leveraging resources like WordNet (Fellbaum, 2012; Miller, 1995). More recently, such techniques have been significantly outperformed by neural models utilised for paraphrase generation like Sequence-to-Sequence (Seq2Seq) (Prakash et al., 2016) and Transformers, which have the ability to learn long-range dependencies in the input sequences (Egonmwan and Chali, 2019). In particular, the presence of individual attention heads within Transformers mimics the behaviour pertaining to the semantic and syntatic structure of sentences (Egonmwan and Chali, 2019; Vaswani et al., 2017a), which is a key factor for paraphrase generation. There have been some other recent approaches for paraphrase generation that combine existing neural models with Deep Reinforcement Learning techniques (Li et al., 2018), wherein, Seq2Seq models are used as generators for producing paraphrases for given sentences, and another DL model is used as evaluator for judging whether two sentences are appropriate paraphrases of each other, based on which rewards are assigned for fine-tuning the generator through Reinforcement Learning. Other promising approaches to paraphrase generation include, for instance hybrid Transformer-Seq2Seq models (Egonmwan and Chali, 2019), Seq2Seq models equipped with Latent Bag of Words (Fu et al., 2019) for performing differentiable content planning and surface realization etc. However, such models generally require large amounts of training data containing original sentences alongside multiple annotated paraphrased sentences, which make them challenging to utilise when such resources are either limited or not available at all, as is the case in the wind industry.

There has been a rising interest in leveraging transfer learning techniques for tackling this problem – wherein, specialised types of large-scale pre-trained Autoregressive Transformer models based on the architecture of traditional neural machine translation models are used for paraphrase generation (Niu et al., 2020). Some of the popular models which have shown success in paraphrase generation include Bidirectional and Auto-Regressive Transformer (BART), Generative Pre-trained Transformer 2 (GPT-2), XLNet, Text-To-Text Transfer Transformer (T5) etc. (Witteveen and Andrews, 2019). We chose to utilise the T5 model for our problem in generating paraphrases for our domain-specific dataset in the wind energy domain, inspired by its simplicity and the state of the art performance the model has achieved in various NLP tasks, including paraphrase generation (Raffel et al., 2020).

Description of the Text-To-Text Transfer Transformer (T5) Model

The Text-To-Text Transfer Transformer (T5) (Raffel et al., 2020) is a large-scale language model that adopts a unified approach in restricting the inputs and outputs to text, making it suitable for a variety of NLP tasks like document summarisation, QA, machine translation, sentiment classification etc., and has recently been adapted for paraphrase generation . The T5 model architecture is identical to the original Transformer architecture, which consists of an encoder-decoder block with self-attention. However, an exception is that the T5 model removes the Layer Norm bias prevalent in original Transformers, places layer normalisation outside the residual path and utilises a different positional embedding technique. The model is inspired by BERT's Masked Language Modelling (MLM) objective. However, the T5 replaces multiple consecutive tokens in input sequences with a single predicted mask token, unlike BERT which utilises specific predicted mask tokens for individual words in the sequences.

The T5 model has been pre-trained on the Colossal Clean Crawled Corpus (C4) dataset, which is a large corpus (approx. 750 GB) containing clean English text that was scraped by the authors from the web. The model trained on this dataset has achieved state of the art results in multiple benchmarks including text classification, summarization, QA etc. The key advantage of the model is that it treats every NLP problem as a text-to-text task, taking a text sequence as input and producing a modified text sequence as output. Note that while the original model has been shown to achieve promising results in open-domain applications, it is essential to fine-tune the model on domain-specific (or closely related) datasets depending on the downstream NLP tasks to optimally leverage transfer learning to enable the model to perform similarly (when fed with similar types of data) in these circumstances. This would eliminate the requirements of training the model from scratch, which is particularly integral in our domain of wind turbine O&M, wherein, the paraphrase generation task to be performed on our human-authored small domain-specific corpus of QC pairs significantly differs from the T5 model's original C4 corpus.

Given that the T5 model can be utilised for a variety of tasks, a prefix is incorporated in the original input sentences being fed to the model to specify the exact task which the model should perform (e.g. *translate:* $< OriginalSentence >$ can be used to specify the model should translate from one language to another etc.).

Utilising the T5 Model for Paraphrase Generation

As our problem focuses on developing a QA system, we utilised a specialised version of the T5 model[8] which has been fine-tuned on the Quora Question Pairs dataset[9] which was originally released by Quora as a part of a 2017 Kaggle competition on recognising semantic equivalence in questions. The Quora dataset contains 404k pairs of historically marked duplicate questions, which serves the goal of obtaining paraphrases for original questions. Some recent studies have also utilised this dataset for various tasks towards natural language understanding (Sharma et al., 2019), including as a part of the General Language Understanding Evaluation benchmark (GLUE) in the original BERT paper (Devlin et al., 2019). Note that we opted for this dataset after a careful analysis and consideration of all openly available datasets (with none presently existing in the wind industry), which showed that this was the closest match to our problem task (humans asking questions and receiving human-authored answers) towards generating diverse, human-like paraphrases.

This model has previously been successfully leveraged for data augmentation through paraphrase generation in a real-world application pertaining to the development of a human-robot interaction framework (Bird et al., 2020). We also explored an alternative T5 model fine-tuned on the Google PAWS dataset for this task, but it was not utilised as most paraphrases generated were completely duplicate repetitions of the original questions.

Below, we describe the process utilised in generating paraphrases for our data:-

1. For the paraphrase generation task, the model takes as input a natural language question $Q = (i_1, i_2, \ldots, i_n)$, and outputs a paraphrased version of the original question as $P_Q = (o_1, o_2, \ldots, o_k) \mid \exists y_m \notin Q$. Here, there would be a maximum of k words in the generated paraphrase P_Q, which together convey a similar meaning as the source question Q. Note that the Cypher queries corresponding to the paraphrases would be exactly the same as in the original questions – it is only the natural language questions for which the paraphrases are generated in our task.

2. In line with the T5 model's requirements, a *string* prefix "*paraphrase:*" is appended at the beginning of the input questions to the model, to indicate the paraphrase generation task to be performed. Besides, an end token $< /s >$ is appended after the input question. An example input to our model is thereby of the form – "*paraphrase: What are general*

[8] T5 model fine-tuned on the Quora Question Pairs dataset: https://github.com/ramsrigouthamg/Paraphrase-any-question-with-T5-Text-To-Text-Transfer-Transformer-

[9] 2017 Kaggle Competition – Quora Question Pairs dataset: https://www.kaggle.com/c/quora-question-pairs

corrective activities for the Electric, Sensor & Control subsystem of the wind turbine? $< /s >$". Note that each of the natural language questions in our original dataset contains the words "*wind turbine*" to ensure that the paraphrases generated by the model are specific to the wind industry.

3. We used the following model parameters – a maximum length for paraphrases generated (*max_length*) as 256 characters, a combination of top-p and top-k sampling with *top_k=120* and *top_p=0.98.* a maximum of 50 independently sampled outputs (signifying the maximum number of paraphrases generated per input question) and early stopping to ensure that the generation terminates when the end of sentence (EOS) token $< /s >$ is reached.

With this process, we obtained a large, augmented dataset consisting of 73,105 QC pairs. As some paraphrases can be completely identical (in situations wherein the model could not generate a unique paraphrased output), we eliminated repeated versions of paraphrases, finally obtaining 72,057 QC pairs which contain only unique paraphrases.[10] The average number of paraphrases generated by the model per input question was 30.518. Figure 7.3 describes some key statistical metrics and linguistic features[11] (such as the various parts-of-speech, composition of unique words and symbols etc.) in the datasets before and after paraphrasing. Clearly, it can be seen that the augmented dataset obtained after paraphrasing has more variation (larger number of unique words, verbs, adjectives etc.). It is also interesting to note that interjections were not present in the original data, but are incorporated through paraphrasing. These variations in linguistic features thereby help account for the diverse nature of human language (Fu et al., 2019), which we wanted to instil into our QA system for the wind industry.

Qualitative evaluation of generated paraphrases Besides the quantitative analysis above, we also performed a qualitative evaluation of the generated paraphrases by utilising Amazon Mechanical Turk (AMT)[12]. In this study, humans were shown the original questions (before paraphrasing) alongside the corresponding generated paraphrases and were asked to assign ratings on a 1-5 Likert scale for semantic similarity, wherein, 1 means *completely different meaning* and 5 means they *express the same meaning*. From the larger augmented dataset obtained after paraphrasing, we randomly selected 10% of the original dataset samples before

[10] Note that in some instances, the generated paraphrases can have varying surface realizations in the form of different character cases.

[11] Obtained through the Natural Language Toolkit (NLTK) (Bird et al., 2009)

[12] https://www.mturk.com

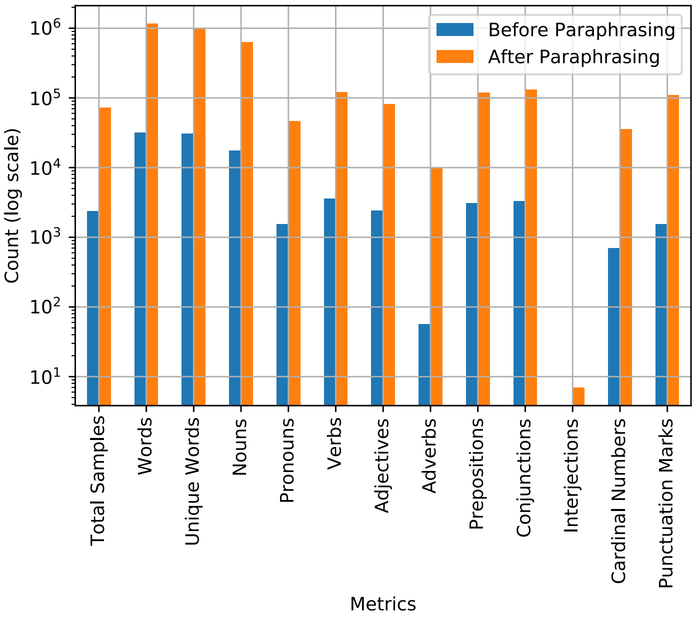


Fig. 7.3 Visualisation of some key statistical metrics and linguistic features in the datasets before and after paraphrasing – the significant rise in parameters for such metrics is clearly visible.

paraphrasing (236 natural language questions). For each of these 236 samples, 15 generated paraphrases were selected.

We asked 33 unique human judges to assign ratings for semantic similarity between the generated paraphrase and the original natural language question. It was also ensured that for each of the cases, two ratings are obtained from two different (unique) human judges, which led to a total of $236 * 15 * 2 = 7,080$ ratings overall. The average rating was obtained as 4.223, with a standard deviation of 1.015. Figure 7.4 summarises the total counts of ratings obtained across different categories on the 1-5 Likert scale. As can clearly be seen, the ratings 5 (*Exactly similar meaning*) and 4 together account for 80% of the total ratings. Also, only 2.6% of the ratings were for 1 (*completely different meaning*). These metrics clearly signify the high quality of the generated paraphrases according to human judgement.

Table 7.3 shows some examples of generated paraphrases obtained using the T5 model. As can clearly be inferred, most of the generated paraphrases are highly semantically similar (in terms of their meaning), grammatical and coherent. Additionally, clearly, there is natural variation in the generated paraphrases, reflected by the change in linguistic features and surface forms (which in some cases also includes variation in the text case). However, for some instances, the generated paraphrases contain additional information which is out of context e.g. in case (b) "*in the FPGA 5M Series wind turbine VVT configuration*" is not

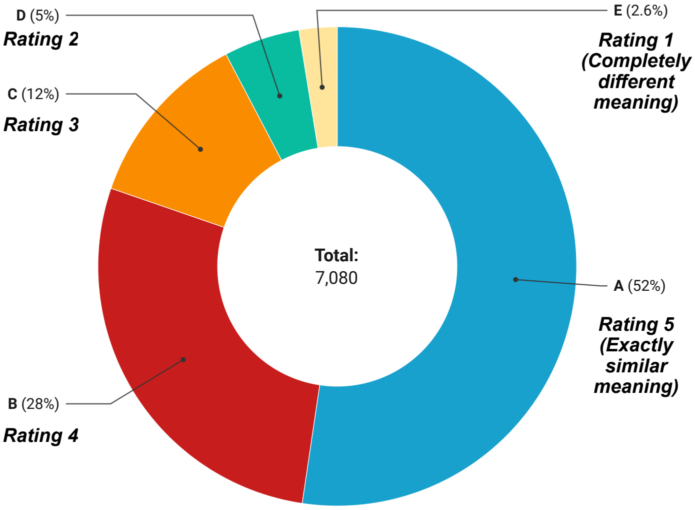


Fig. 7.4 Donut chart outlining the composition of assigned human ratings in AMT on the 1-5 Likert scale for semantic similarity, based on the original questions and their generated paraphrases. Mean rating is 4.223 with standard deviation of 1.015.

related to the turbine under consideration for our study (7 MW LDT), but pertains to a different wind turbine model. However, it clearly does not affect the meaning in terms of the domain-specific context for the question, as for any wind turbine, the context of retrieving corrective actions pertaining to an anomaly in its *pitch angle* would be similar. For case (d), there is unexpected information pertaining to "*Brake Circuit low pressure will take 4-6 hours*", which is redundant and does not make sense based on the question's context. However, it is interesting to note that this is likely an answer to the query which is outputted within the generated paraphrase itself, and given that the original question is also present within the complete paraphrase, it does not affect the meaning and context of retrieving corrective activities for low pressure in the brake circuit of the turbine. In another instance e.g. in case (e), the word "*power*" is unexpected, as the model should have ideally generated the word "*wind*" instead to signify *wind turbine*. However, again, the context of the question remains unchanged, as it clearly enunciates the query pertaining to fault states which occur due to SCADA feature 52.

Table 7.3 Examples of original natural language questions alongside the generated paraphrases – note the variations in the order of words, types of linguistic features and character cases. Any errors/inconsistencies which would not be expected based on our domain knowledge are highlighted.

Original question (before paraphrasing)	**Generated paraphrases**

(a) What are some important details for the Power Cabinet subsystem of the wind turbine?	1. What are some aspects of the Power Cabinet subsystem of the wind turbine? 2. What are some key details for the power cabinet of wind turbine? 3. What are the fundamental features of power cabinet subsystem of wind turbine?
(b) What are specific corrective actions due to inconsistency in Pitch Angle Maximum Value wind turbine SCADA feature?	1. What are the corrective actions due to inconsistency in Pitch Angle Maximum Value Wind Turbine SCADA feature? 2. What are the proper corrective actions due to inconsistency in pitch angle Maximum Value wind turbine SCADA? 3. What are the specific corrective actions due to inconsistency in pitch angle maximum value wind turbine SCADA feature in the FPGA 5M Series wind turbine VVT configuration?
(c) What specific preventive maintenance activities are required for the wind turbine Transformer?	1. What types of preventive maintenance activities are required for the wind turbine transformer? 2. What are a few preventive maintenance activities required for a wind turbine transformer? 3. Which preventive maintenance program is required to maintain a wind turbine transformer?
(d) Show specific corrective actions for Brake circuit low pressure failure in the wind turbine.	1.What should I do if I am still driving a wind turbine with low pressure but the Brake Circuit fails? 2. Explain specific corrective actions for Brake circuit low pressure failure in the wind turbine. 3. Show specific corrective actions for Brake circuit low pressure failure of a wind turbine. Brake Circuit low pressure will take 4-6 hours.
(e) What fault events are caused by wind turbine SCADA feature number 52?	1. How are wind turbine fault events dealt with by SCADA feature number 52? 2. What fault events are caused by wind turbine SCADA feature number 52 and how are they determined by their location? 3. What is the fault state caused by power turbine SCADA feature number 52?

Thereby, it is clear that while there are subtle inconsistencies and errors in the generated paraphrases, as would normally be expected from a large-scale language model leveraging transfer learning on a dataset from a different domain (the Quora database), the overall nature of the generated paraphrases is promising, as our key goal is to maintain the meaning of the questions within the context of the wind industry. Moreover, we observed that such major inconsistencies only occur in around 5% (3,603 cases) of the total dataset samples, which is a very minor proportion of our overall data with 72,057 samples. Additionally, some inconsistencies can actually prove worthwhile for our QA tasks, in cases wherein engineers and technicians may unknowingly input corrupted/incomplete details in the questions (e.g. incomplete alarm names, redundant information for fault events when querying corrective actions etc.). Thereby, we would utilise this augmented dataset for our further experiments in training learning models for the QA system in the forthcoming sections.

7.5 Learning models

In this section, we discuss the basic architecture of the Seq2Seq and Transformer learning models that are utilised in this chapter. First, the concept of sequence-to-sequence code generation in encoder-decoder models is introduced. Then, we describe the principle of extending our Seq2Seq(Att) model's sequential learning process and developing the Transformer model based on these descriptions. Note that while our primary learning model is a Transformer, we utilise the Seq2Seq(Att) model in this chapter as a baseline model for comparison of performance metrics.

7.5.1 Attention-based Sequence-to-Sequence model

We propose to develop our baseline model towards generating Cypher queries as an attention-based Sequence-to-Sequence encoder-decoder RNN (Mikolov et al., 2010; Sutskever et al., 2011), which would learn to condition a sequence of Cypher query fragments on the sequence of words present in a natural language question. Refer to Chapter 6 for more details on the architecture of the Seq2Seq(Att) model.

To generate Cypher queries, we let the input sequence $\mathbf{x}$ correspond to a natural language question posed by the engineers during O&M. We assume that the output sequence $\mathbf{y}$ corresponds to a sequence of code fragments that together form a valid Cypher query, which can retrieve the most appropriate O&M responses for $\mathbf{x}$ from our domain-specific KG. An alignment vector $\alpha = (\alpha_1, \dots, \alpha_N)$ focuses on the encoder's outputs and has the same length as the source sequence, based on which the decoder then predicts an output sequence representing a complete Cypher query that is conditioned on the context vector $\mathbf{s}$ and all previously predicted code fragments $\{y_1,\ldots,y_{t-1}\}$:

$$p(y_t|\{y_1,\ldots,y_{t-1}\},\mathbf{s}) = g(y_{t-1},\mathbf{h_t},\mathbf{s}), \tag{7.1}$$

where g is a nonlinear function.

Gated Recurrent Unit and Attention To address common problems of vanishing or exploding gradients (Bengio et al., 1994), we will use a Gated Recurrent Unit (GRU) (Cho et al., 2014a) for implementation of our model. A GRU computes $\mathbf{h}$ under consideration of two gates which play an integral role in controlling the model's loss and incorporation of information: the "update gate" $\mathbf{z_t}$ and "reset gate" $\mathbf{r_t}$, leading to an updated computation of $\mathbf{h}$ at time t as:

$$\mathbf{z_t} = \sigma\left(\mathbf{W_z}\cdot[\mathbf{h_{t-1}},\mathbf{x_t}]\right) \tag{7.2}$$

$$\mathbf{r_t} = \sigma\left(\mathbf{W_r}\cdot[\mathbf{h_{t-1}},\mathbf{x_t}]\right) \tag{7.3}$$

$$\tilde{\mathbf{h}}_\mathbf{t} = \tanh\left(\mathbf{W}\cdot[\mathbf{r_t} * \mathbf{h_{t-1}},\mathbf{x_t}]\right) \tag{7.4}$$

$$\mathbf{h_t} = (1-\mathbf{z_t}) * \mathbf{h_{t-1}} + \mathbf{z_t} * \mathbf{h_t} \tag{7.5}$$

Here, σ denotes the Logistic Sigmoid function.

We also integrate an attention mechanism into our model, which provides added transparency by identifying the specific words in the input and output sequences which are important for the model's prediction. Figure 7.5 shows the basic structure of the Seq2Seq(Att) model utilised for our study.

7.5.2 Transformer model

We utilise a Transformer (Vaswani et al., 2017a) as our primary learning model for graph query language generation. Refer to Chapter 6 for more details on the Transformer architecture.

In this chapter, our goal is to extend the original Transformer architecture towards the domain-specific application of QA in the wind industry. We aim to realise this by utilising the sequence of words in a natural language question $\mathbf{x} = (x_1,\ldots,x_N)$ as input to the model, and generating the appropriate Cypher query $\mathbf{y} = (y_1,\ldots,y_M)$ which is most appropriate to the

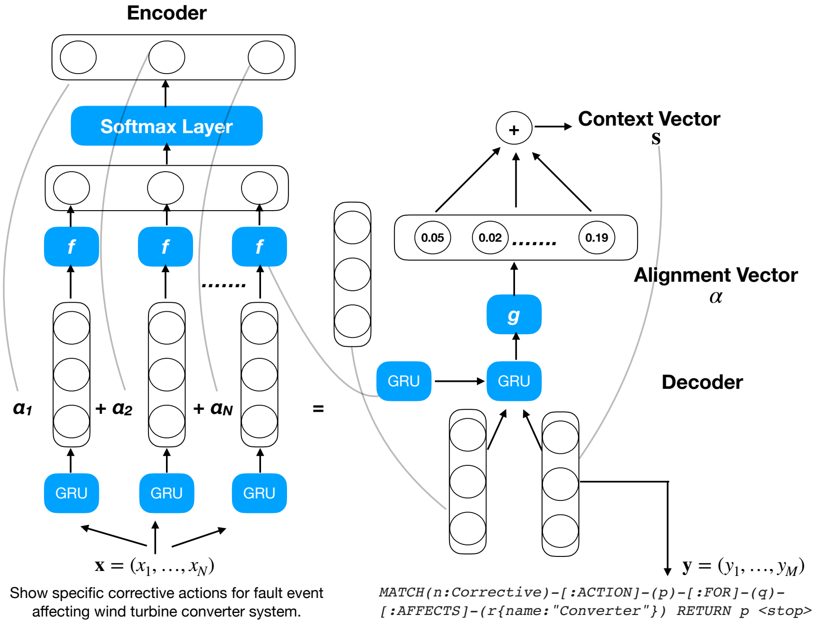

Encoder
Softmax Layer
f
f
f
....
a_1
$+ a_2$
$+ a_N$
=
GRU
GRU
GRU
$\mathbf{x} = (x_1, \ldots, x_N)$
Show specific corrective actions for fault event affecting wind turbine converter system.
+
Context Vector
s
0.05
0.02
....
0.19
Alignment Vector
α
g
GRU
GRU
Decoder
$\mathbf{y} = (y_1, \ldots, y_M)$
MATCH(n:Corrective)-[:ACTION]-(p)-[:FOR]-(q)-[:AFFECTS]-(r{name:"Converter"}) RETURN p <stop>

Fig. 7.5 Basic structure of the Seq2Seq(Att) model utilised for graph query language generation – given a input sequence of words in a natural language question, the model would generate the appropriate Cypher queries for information retrieval from our domain-specific KG database.

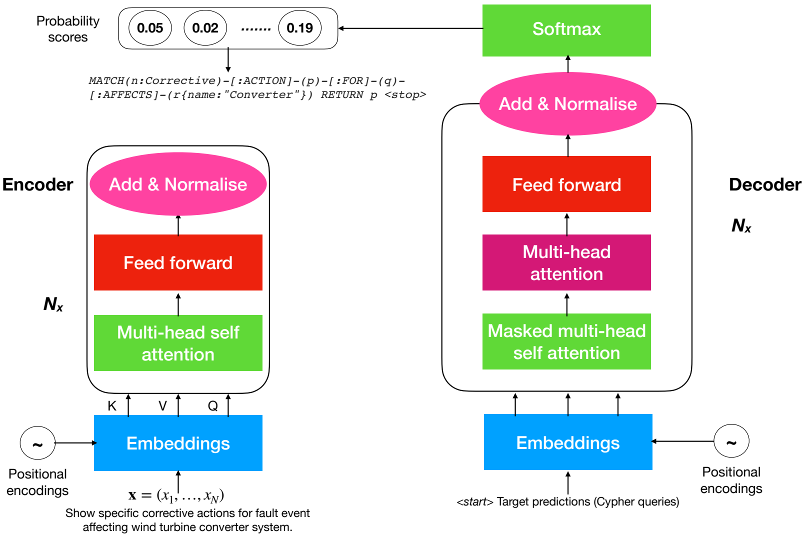

Probability scores
0.05
0.02
....
0.19
MATCH(n:Corrective)-[:ACTION]-(p)-[:FOR]-(q)-[:AFFECTS]-(r{name:"Converter"}) RETURN p <stop>
Softmax
Add & Normalise
Encoder
Add & Normalise
Feed forward
N_x
Multi-head self attention
K
V
Q
Embeddings
~
Positional encodings
$\mathbf{x} = (x_1, \ldots, x_N)$
Show specific corrective actions for fault event affecting wind turbine converter system.
Feed forward
Multi-head attention
Masked multi-head self attention
Decoder
N_x
Embeddings
~
Positional encodings
<start> Target predictions (Cypher queries)

Fig. 7.6 Architecture of the Transformer model (adapted from (Vaswani et al., 2017a)) utilised for graph query language generation in our QA system – the self-attention mechanism facilitates learning in the absence of recurrence prevalent in the conventional Seq2Seq architecture.

context of domain-specific information requested by the turbine engineers and technicians. Additionally, similar to the Seq2Seq(Att) model, we aim to appropriately leverage the attention weights of the Transformer model to gain insights into the model's decisions besides making accurate predictions of Cypher queries. Figure 7.6 provides an overview of the Transformer architecture that we utilise in our QA system.

7.6 Experiments

For our experiments, we utilise the data of 72,057 QC pairs obtained after paraphrasing, as described in Section 7.4 for training the learning models. Initially, we performed word-level tokenization of the QC pairs, filtered on whitespace characters. We did not use lower-casing during tokenization as Cypher queries are case-sensitive. All numbers, special symbols and punctuations were retained in the tokenized data to ensure the generation of valid Cypher queries. Additionally, an *[UNK]* token was incorporated to account for words that fall out of vocabulary. A $< start >$ and $< stop >$ token were also added to the original samples to help the models clearly understand when to start and stop predicting.

We utilised a train-test split ratio of 70-30%, ensuring that the dataset is split into 50,439 training and 21,618 test instances. A batch size of 128, 256-dimensional word embeddings, input vocabulary size of 26,034 words and target vocabulary size of 1,181 words were used for training all models for 50 epochs. All models were trained in TensorFlow (Python) (Abadi et al., 2015) with Adam optimisation.

- **Seq2Seq(Att):** The Seq2Seq model with GRU and Bahdanu attention, see Section 7.5.1 for details. We experimented with 512/1,024 hidden units, 2 hidden layers and a learning rate of 0.01.

- **Transformer:** The Transformer model with multi-head attention mechanism, refer to Section 7.5.2 for details. We experimented with model sizes of 128/256, 2/4 total layers (multi-head attention + feed-forward layers) and 2/4/8 attention heads. The model's learning rate was decayed with the *WarmupThenDecaySchedule* class in TensorFlow with 5,000 warmup steps.

To retrieve the relevant responses/answers from our domain-specific KG, we utilised *Py2neo*[13], a specialised Python library which facilitates interfacing of the Neo4j KG database server with Python applications. Given a natural language question posed by the user, the predicted Cypher queries are automatically executed in Neo4j and O&M actions retrieved, which ultimately provides an environment for automated reasoning in our QA system.

[13] Py2neo Information Handbook: https://py2neo.org/2020.1/

7.7 Results

In this section, we discuss the experimental results obtained in generating Cypher queries with the Seq2Seq(Att) and Transformer models. We also provide a qualitative evaluation of the generated queries and perform an error & output analysis for each model. Some example cases of retrieved responses from the KG based on the Cypher queries predicted by the models corresponding to natural language questions are also discussed.

7.7.1 Objective evaluation

Tables 7.4 and 7.5 show the performance metrics for objective evaluation of the Seq2Seq(Att) and Transformer models respectively in terms of the percentage of Cypher queries that are correctly predicted and computation time. We can clearly see that the Transformer outperforms the Seq2Seq(Att) model in terms of percentage of Cypher queries correctly predicted, achieving an accuracy of up to 89.75%. The Seq2Seq(Att) model attains the highest accuracy of 88.99%, which is 0.76% worse than the Transformer model. This is, although a very minor improvement in the model's performance.

Table 7.4 Performance metrics for Cypher queries predicted by the **Seq2Seq(Att)** model – average computation time per epoch is shown in brackets. The best performing model is outlined in bold face.

Hidden units	**Total samples correctly predicted**	**Accuracy**	**Computation time**
512	19,106/21,618	88.38%	**1 hr 29 min 14 sec (1.784 min)**
1,024	19,238/21,618	**88.99%**	3 hr 14 min 38 sec (3.892 min)

Another important metric we would like to reflect on is the computation time. We note that the Transformer model is the fastest, achieving the shortest computation time (20 min 34 sec). On this front, the best-performing Seq2Seq(Att) model takes 3 hr 14 min 38 sec (946.35% more than the best-performing Transformer). Note that these computation times were obtained with the NVIDIA Tesla K80 GPU based on Google's Compute Engine. We also observe that scaling up the Transformer model (including the model size and number of attention heads) degrades the model's performance while leading to increased computation time. It is likely that more than two attention heads are not integral for learning good representations of the Cypher queries in our dataset. Given the mostly common syntax and

Table 7.5 Performance metrics for Cypher queries predicted by the **Transformer** model – average computation time per epoch is shown in brackets. The best performing model is outlined in bold face.

Model size	**Attention heads**	**No. of layers**	**Total samples correctly predicted**	**Accuracy**	**Computation time**
128	4	4	17,655/21,618	81.66%	38 min 32 sec (46.24 sec)
	2		18,461/21,618	85.396%	33 min 7 sec (39.74 sec)
		2	**19,403/21,618**	**89.75%**	**20 min 34 sec (24.68 sec)**
256	8	4	17,859/21,618	73.36%	1 hr 29 min 7 sec (1.782 min)

structure of Cypher queries (e.g. all queries use common words like *MATCH, WHERE, RETURN* etc.), besides learning to generalise to intricacies of the graph query language, the models only need to learn the unique domain-specific information (e.g. alarm types, SCADA feature names etc.), rather than a large-scale vocabulary and long sequences generally prevalent in machine translation tasks. This is likely the reason why the Transformer only has an improvement of 0.76% over the Seq2Seq(Att), clearly indicating that the Bahdanu attention in our Seq2Seq(Att) model suffices for the learning task if we ignore the marginal gain in accuracy.

Considering the environmental impact of our learning models As we have discussed above, the computation time for the best-performing Seq2Seq(Att) model (with 1,024 hidden units) is more than nine times that of the Transformer and for the smaller Seq2Seq(Att) model (with 512 hidden units), the computation time is more than 4.5 times as the Transformer. More notably, all our Transformer model configurations achieve lower computation time than the Seq2Seq(Att) model, which is in line with the parallelisation capabilities of Transformers on GPUs. While our models are not exponentially large given the limited availablility of data at present in the wind industry, we believe training AI models with larger datasets in the wind energy sector could make the computation time scale exponentially, particularly as new datasets focusing on O&M continue to become available, newer turbines are deployed and older turbines in operation record more data every day from their sensors.

Some recent studies (Bender et al., 2021; Henderson et al., 2020; Schwartz et al., 2020) have highlighted the environmental impact of DL systems, wherein, scaling up the model size can have serious negative environmental consequences of its resource consumption. Moreover, these studies highlight the importance of prioritising energy efficiency for reducing negative environmental impact and inequitable access to resources (Bender et al., 2021). As our key goal in this chapter is to leverage AI in helping make wind energy sources more reliable towards tackling climate change, we echo the importance of considering dangers of rising carbon emissions over marginal improvements in performance of large language models. Converse to our results, even if the Seq2Seq(Att) model had achieved a marginal improvement over the Transformer, we believe it would be the most rational to discard the marginally better Seq2Seq(Att) model in favour of the Transformer's computational efficiency. As the Transformer model meets both these expectations (best performance as well as lowest computation time), we believe they are highly promising for utilisation in the wind industry for real-time decision support when considering the *AI and Society* perspective.

7.7.2 Error and output analysis

Table 7.6 shows some examples of Cypher queries predicted by each of the models alongside the expected queries corresponding to natural language questions. Note that our QA system is case-insensitive, and any capitalisation of component names (e.g. Yaw), SCADA feature labels (e.g. *ReactivePower_kVAr_Max*) etc. in the natural language question examples shown only reflect the test data sample variations (including in semantic structure, linguistic features and word-cases) which were introduced during paraphrasing – the model would work the same way and generate the same Cypher queries if different case words are passed during inference.

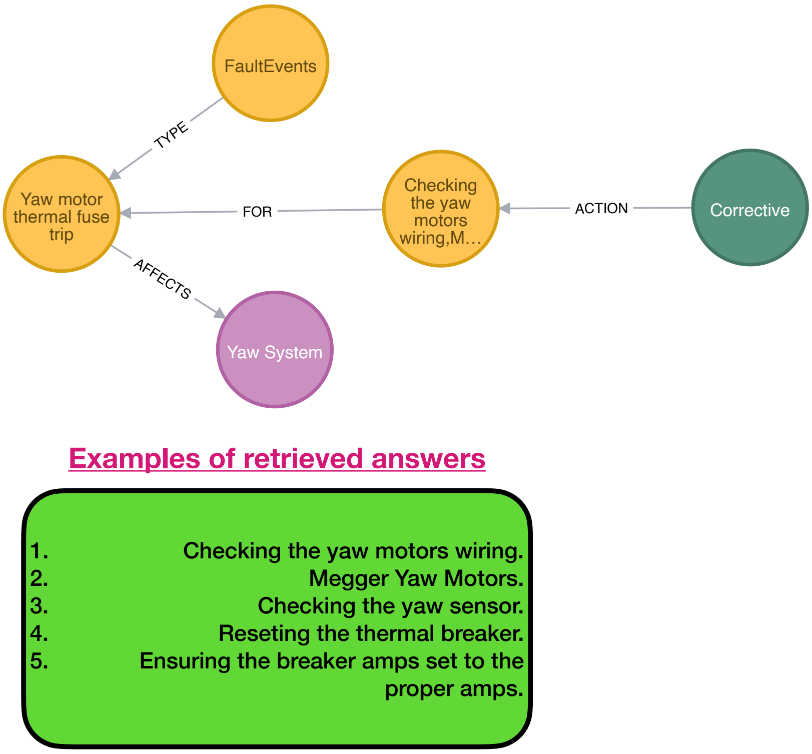
Table 7.6 Examples of Cypher queries generated by the Seq2Seq(Att) and Transformer models, with remarks about their viability – $ denotes missing words, symbols or numbers in the predicted Cypher query.

Natural language question	**Expected Cypher query**	**Predicted Cypher query with Seq2Seq(Att)**	**Predicted Cypher query with Transformer**
(i) What are the corrective actions for Yaw motor thermal fuse trip fault event?	*MATCH(n:Corrective)-[:ACTION]-(p)-[:FOR]-(q{details:"Yaw motor thermal fuse trip"}) RETURN p*	As expected	As expected

(ii) Which SCADA features contribute to high temperature on the gearbox oil fault event in the wind turbine?	MATCH(n:FaultEvents)-[:TYPE]-(p{details:"High temperature on the gearbox oil"})-[:RELATESTO]-(q:Feature) RETURN q	As expected	As expected
(iii) What is the effect of an alarm on the wind turbine yaw system?	MATCH(n)-[:AFFECTS]-(p{name:"Yaw System"}) RETURN n	MATCH(n${name: "Yaw System"}) RETURN n	As expected
(iv) What are details of ReactivePower_kVAr_Max wind turbine SCADA feature?	MATCH(n:Feature{ name:"ReactivePower _kVAr_Max"}) RETURN n	As expected	As expected
(v) What are fault events related to alarm 923 in the wind turbine?	MATCH(n{alarm_no: "923"})-[:RELATESTO]-(p) RETURN p	MATCH(n{alarm_no: "922"})-[:RELATESTO]-(p) RETURN p	As expected
(vi) What causes the fault events of the Absolute Wind Direction maximum value SCADA feature?	MATCH(n:Feature{ description:"Absolute Wind Direction Maximum Value"})-[:RELATESTO]-(p) RETURN p	As expected	As expected
(vii) What are some examples of SCADA causes of blade positioning error in wind turbine?	MATCH(n:FaultEvents)-[:TYPE]-(p{details:"Blade Position Error"})-[:RELATESTO]-(q:Feature) RETURN q	MATCH(n: FaultEvents)-[:TYPE]-(p{details:"Generator-rotor speed discrepancy"})-[:RELATESTO]-(q:Feature) RETURN q	As expected
(viii) What are the predictive activities for the Power Cabinet of Wind Turbines?	MATCH(n{name:"Power Cabinet"}) RETURN n.PredictiveActivities	MATCH(n{ name:"Generator"}) RETURN n.PredictiveActivities	MATCH(n:Predictive)-[:ACTION]-(p)-[:FOR]-(q{name:"Power Cabinet"}) RETURN p

We discuss some notable cases below:-

- In case *(i)* of Table 7.6, the natural language question pertains to retrieving corrective O&M activities for a fault event in the yaw motor. The Seq2Seq(Att) and Transformer models both predict the Cypher queries correctly in this case. Figure 7.7 shows the


FaultEvents
TYPE
Yaw motor thermal fuse trip
FOR
Checking the yaw motors wiring,M…
ACTION
Corrective
AFFECTS
Yaw System

Examples of retrieved answers

1. Checking the yaw motors wiring.
2. Megger Yaw Motors.
3. Checking the yaw sensor.
4. Reseting the thermal breaker.
5. Ensuring the breaker amps set to the proper amps.

Fig. 7.7 Output on executing the predicted Cypher query by the Seq2Seq(Att)/Transformer models in case *(i)* of Table 7.6 – examples of retrieved O&M actions for yaw motor fault event based on identified nodes and properties in our KG database are also shown.

output of executing the predicted Cypher query in this case. As can be seen, given that the Cypher query is correctly predicted corresponding to the domain-specific natural language question, the information retrieved from the KG is also accurate. Additionally, a visual depiction of the relevant nodes and properties in the KG relevant to the question is also obtained, facilitating intuitive decision making for turbine engineers and technicians.

- In case *(iii)* of Table 7.6, the natural language question focuses on determining the effect of an alarm event on the turbine's yaw system. While the Transformer model correctly predicts the Cypher query in this situation, the Seq2Seq(Att) model only predicts the query partially, missing out on the *AFFECTS* relationship. This a critical error as the incorrect Cypher query would provide details of the yaw system of the turbine itself, rather than the operational inconsistencies/errors which take precedence due to an alarm in the yaw system. Figure 7.8 outlines the retrieved O&M actions from the KG in this case with the Transformer model.

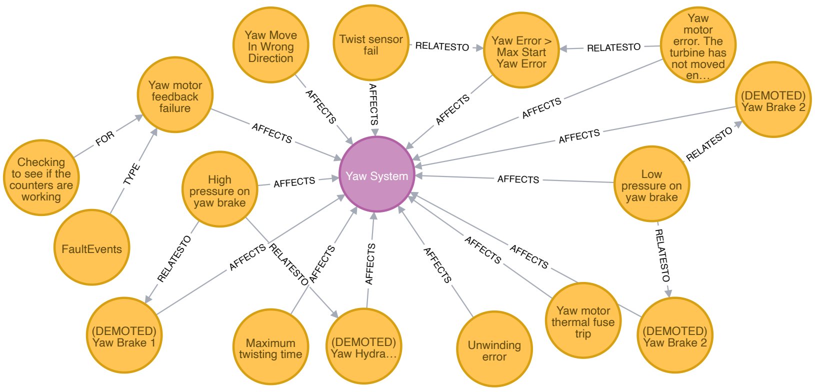


Fig. 7.8 Output on executing the Cypher query predicted by the Transformer model in case *(iii)* of Table 7.6 – the nodes shown denote the inconsistencies/errors caused due to yaw system alarm

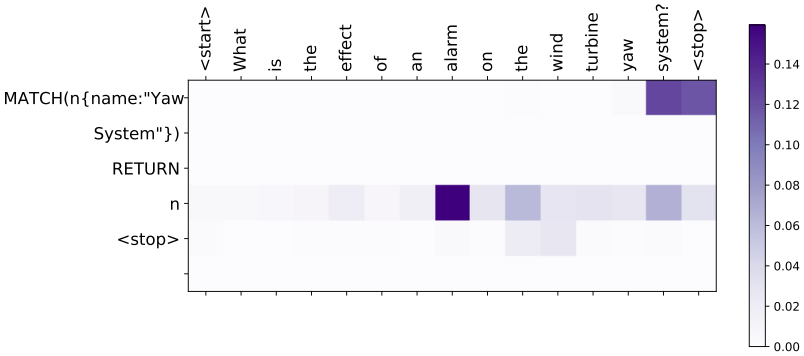


Fig. 7.9 Heatmap visualisation of attention weights for prediction of Cypher query in case *(iii)* of Table 7.6 by the Seq2Seq(Att) model – note the *AFFECTS* relationship which the model misses by failing to focus on the question's essential keyword *alarm* when predicting the initial *MATCH* fragment.

To inspect and analyse the working of each of the models during the prediction making process and the likely cause(s) of any errors, we visualise the attention weights of the models. Figures 7.9 and 7.10 show the attention weights for the Cypher queries predicted by the Seq2Seq(Att) and Transformer models respectively. As can clearly be seen, the highest weights (activations) of the Seq2Seq(Att) model are attributed to the word *system* for predicting the *MATCH* fragment of the node. However, the keyword of *alarm* (which is the essence of the question) is focused on the node's label *n* rather than the *Yaw* itself wherein the alarm occurs. On the other hand, the Transformer has a significant attention weight on the word *alarm* when it is predicting the *MATCH* segment of the relationship correctly that includes the *AFFECTS* relationship, which is essential as the query corresponds to retrieving the inconsistencies/errors which

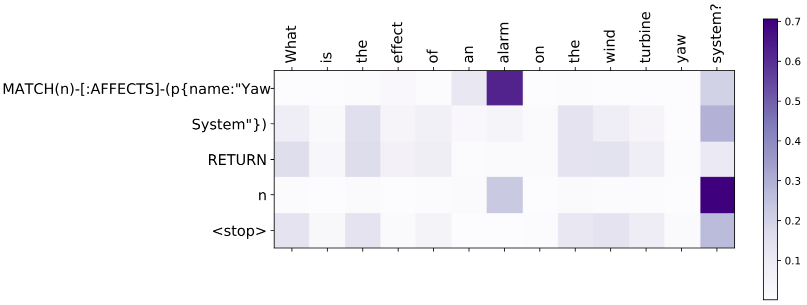


Fig. 7.10 Heatmap visualisation of attention weights for prediction of Cypher query in case *(iii)* of Table 7.6 by the Transformer model – note that the model focuses on the keyword of the question *alarm* when predicting the *MATCH* fragment and *system* when predicting the appropriate node label *n*, thereby generating the complete Cypher query correctly.

arise out of the yaw system alarm. Besides, note that when the end of the question is reached, the Transformer places a significant emphasis on the node's label *n* whereas, the Seq2Seq(Att) model only focuses on the initial *MATCH* code fragment. This visualisation further suggests that the Seq2Seq(Att) model is lacking in its ability to focus on the keywords in natural language questions (e.g. *alarm* in this case) when predicting the code fragments, whereas, the Transformer performs suitably in identifying the keywords, thus leading to the correctly generated Cypher query.

- In case *(vii)* outlined in Table 7.6, the Seq2Seq(Att) model fails to predict the complete Cypher query pertaining to identifying SCADA features which cause a blade positioning error in the turbine. The Seq2Seq(Att) model incorrectly references the blade positioning error due to a discrepancy in the generator rotor speed, which has no relation whatsoever with the turbine's blade position. The Transformer is able to predict the complete valid Cypher query in this case. Figure 7.13 depicts the output of execution of the Cypher query, summarising the details of the SCADA features which contribute to the blade positioning error fault event, ultimately affecting the turbine's pitch system.

  To analyse the reasons for the Transformer's valid prediction, it would be useful to visualise the self-attention weights of the model in this situation. Note that not all attention visualisations are easily comprehensible by human engineers as they represent the focus elements of the model during its prediction making process – thereby, we try to provide a perspective on the model's weights based on our domain knowledge.

  Figures 7.11 and 7.12 outline the self-attention weights for the Transformer model's encoder and decoder respectively, describing the model's focus during the prediction

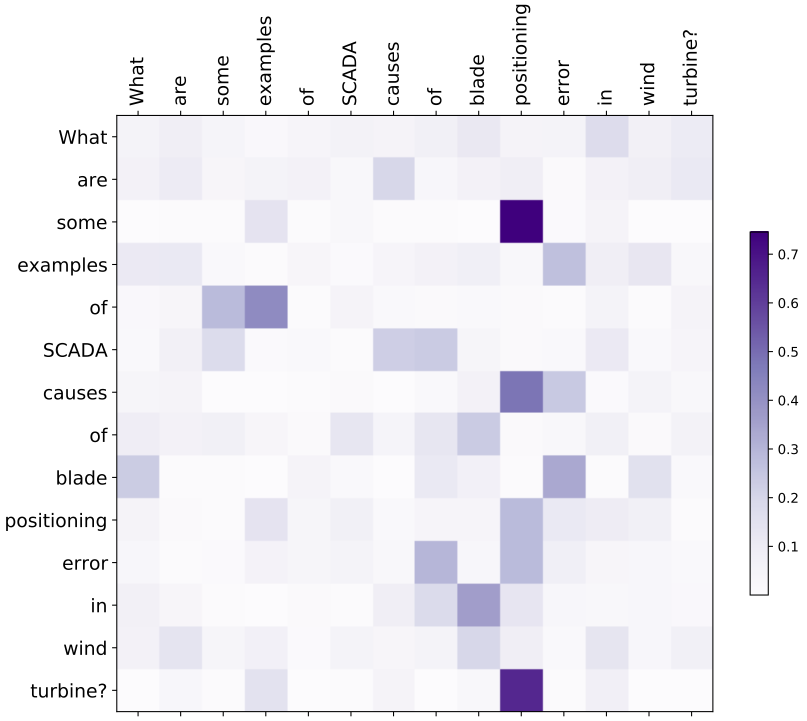


Fig. 7.11 Heatmap visualisation of self-attention weights for the Transformer's encoder during prediction of Cypher query in case *(vii)* of Table 7.6

when the source and target sequences attend to themselves. As can clearly be seen, the encoder places significant emphasis on the word *positioning* corresponding to *some*, *causes* and *turbine* etc. – which is reasonable as the question pertains to a fault event caused by a blade positioning error in the turbine. Additionally, note the other relevant attention weights such as *error* corresponding to *blade*, *blade* corresponding to *in*, *of* corresponding to *SCADA*, *examples* corresponding to *of* etc., which clearly signifies that examples of contributing SCADA features are to be retrieved for a fault which occurs in the turbine's blades. Besides, the self-attention weights at the decoder end of the model also show effective focus of the Transformer on the *position* keyword corresponding to the *MATCH* fragment – which references the blade's anomaly type being in its position. Note the high attention weights for the *RETURN* keyword corresponding to multiple other Cypher query fragments such as *RELATESTO* and the node label *q* – these are highly reasonable as the model aims to generate a complete valid Cypher query that can return multiple SCADA features from the KG which are relevant to a specific type of fault event (blade position error) and these thus relate to

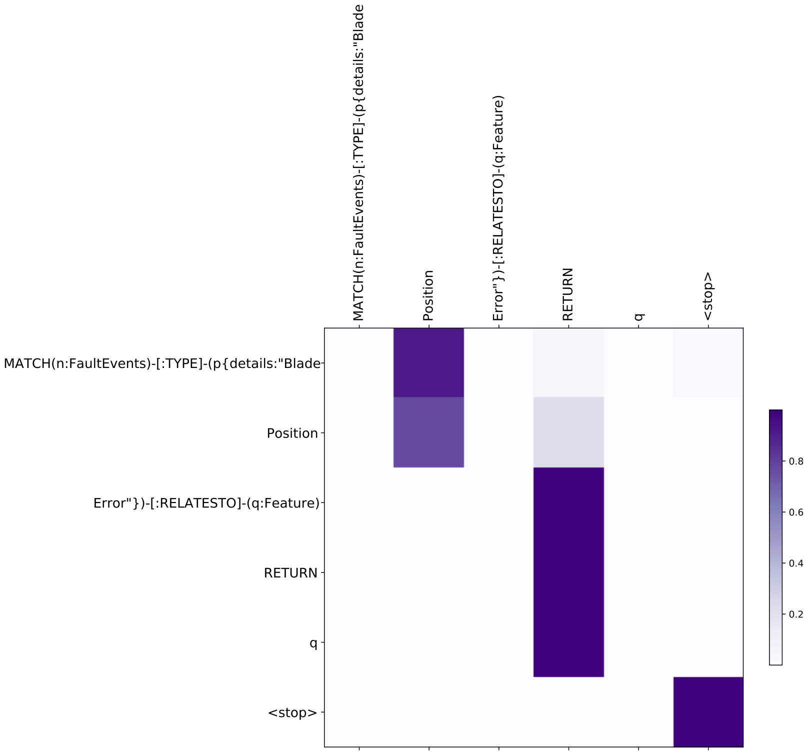
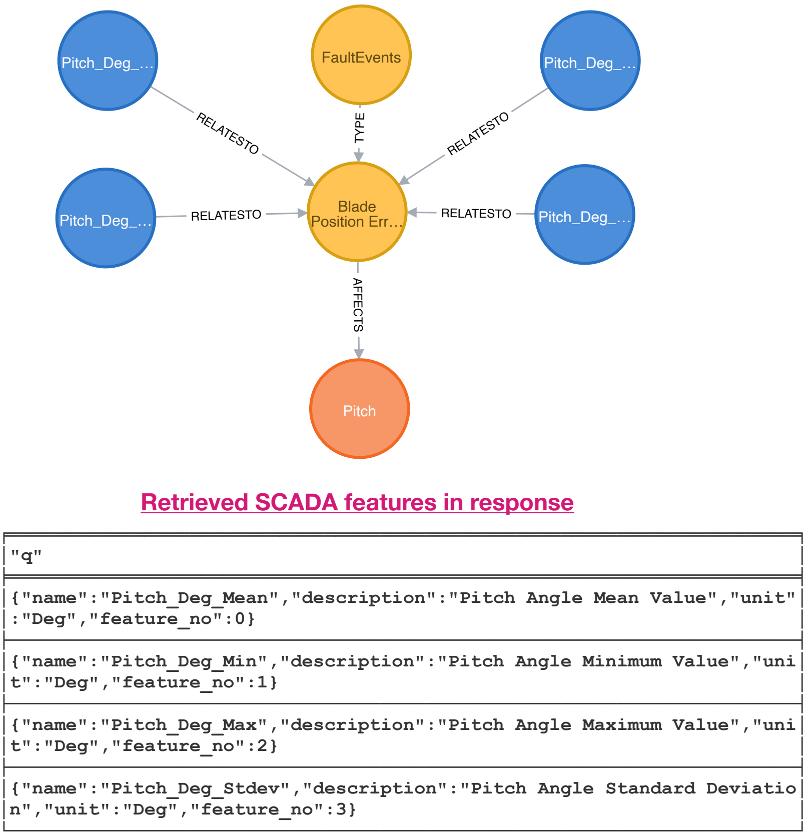


Fig. 7.12 Heatmap visualisation of self-attention weights for the Transformer's decoder during prediction of Cypher query in case *(vii)* of Table 7.6

the SCADA features with the *RELATESTO* relationship and are referenced by the *n* node label, which the model rightly focuses on.

- For case *(viii)* of Table 7.6, both models fail to accurately predict the Cypher query for retrieving predictive actions required for the turbine's power cabinet. While the Seq2Seq(Att) model makes a completely incorrect prediction of the node's name – predicting maintenance actions for the *Generator* instead of the *Power Cabinet*, the Transformer model still accurately infers the node's name in this scenario. However, the Transformer predicts the Cypher query towards retrieving associated nodes based on the *ACTION* relationship rather than the *PredictiveActivities* property of the *Power Cabinet* node itself. This is still a relatively less fatal error, as the incorrect query generated by the Seq2Seq(Att) would predict activities for the *Generator*, which is a specific internal sub-component of the *Power Cabinet*, while the *Transformer* would return specific predictive activities for the associated events (e.g. SCADA features and alarms) that affect the *Power Cabinet* rather than the generic system itself (reflected by the system's individual property). Thus, the Cypher query generated by the Transformer (despite not being completely accurate) is still realistic to avert faults



Retrieved SCADA features in response

```
"q"
{"name":"Pitch_Deg_Mean","description":"Pitch Angle Mean Value","unit"
:"Deg","feature_no":0}
{"name":"Pitch_Deg_Min","description":"Pitch Angle Minimum Value","uni
t":"Deg","feature_no":1}
{"name":"Pitch_Deg_Max","description":"Pitch Angle Maximum Value","uni
t":"Deg","feature_no":2}
{"name":"Pitch_Deg_Stdev","description":"Pitch Angle Standard Deviatio
n","unit":"Deg","feature_no":3}
```

Fig. 7.13 Output on executing the Cypher query predicted by the Transformer model in case *(vii)* of Table 7.6 – the nodes shown represent the SCADA features which contribute to the blade positioning error fault event

by rectifying SCADA features and alarms (which directly affect the *Power Cabinet*) in this scenario.

7.8 Discussion

By mapping natural language questions posed by engineers and technicians to the appropriate Cypher queries for retrieving domain-specific information, the proposed approach provides a completely automated QA system for O&M of wind turbines. The approach is also effective as whenever a Cypher query representation is correctly predicted for a natural language question, the information retrieved would always be factually correct as it is retrieved from a domain-specific KG in the wind industry. However, there are some significant limitations of this chapter which are caused due the unavailability of large amounts of domain-specific O&M information openly for research & development. While we have demonstrated that techniques such as paraphrasing can support data augmentation and provide a highly effective solution for QA, there are still some key challenges which our study faces due to utilisation of the Quora Question Pairs database for fine-tuning the T5 Transformer model towards data augmentation of our original domain-specific corpus in the absence of any other source of viable information. Moreover, our error and output analysis has shown that in some (rare) situations, neither the Transformer nor the Seq2Seq(Att) model are able to correctly predict Cypher queries for information retrieval, which would mean that the O&M operators would still need to rely on human feedback and existing documentation (e.g. maintenance manuals).

We do not claim that our QA system can be directly utilised in the wind industry in its present state – the system would still require human engineers to analyse the responses provided and account for the incorrect decisions which the AI-based QA system can make, particularly in situations wherein the questions posed by engineers are significantly different from the training data, which, despite having significant amount of variations and diversity introduced by paraphrasing may still suffer from rarer alarm events and faults which our models have not witnessed previously. We believe that the QA system can be improved further in the future by utilising human inputs for continuously optimising the trained models and potential avenues for development of larger amounts of domain-specific O&M information by the wind farm operators – which continues to become closer to our vision with data-driven decision support becoming increasingly popular in the wind industry. In future, we plan to develop such datasets and make them publicly available to the expert and intelligent systems community, which can hopefully encourage further research in this direction to help AI-based interactive decision support systems transition from academic labs to real-world operational use-cases in the industry – particularly within the wider context of AI for Social Good towards tackling climate change.

7.9 Conclusion

In this chapter, we have presented a novel application of AI-based intelligent QA systems for facilitating interactive decision support in the wind industry to support O&M. By leveraging a domain-specific KG for information retrieval of O&M strategies and developing a specialised corpus of natural language questions-graph query language pairs, we have demonstrated the promising role of training AI models for automated reasoning during QA in a domain-specific application for the wind energy sector. Our proposed approach has utilised Transformers for automatically generating the code for querying the KG during O&M based on natural language questions posed by engineers and technicians. Experiments with a Seq2Seq(Att) baseline model and the Transformer have shown that while the Transformer only marginally outperforms the Seq2Seq(Att) model in terms of accuracy, it takes one-ninth the time to train in comparison. On considering the environmental impact of our learning models and the overall goal of tackling climate change with AI, we have shown that Transformers are highly promising for automated QA in the wind industry, achieving optimal performance at the lowest computational cost. We also include a brief video demonstration of the proposed QA system[14]. We envisage that our QA system can support O&M by helping provide appropriate O&M strategies to engineers for fixing/averting faults and operational inconsistencies and reducing the decision making time. This can potentially pave the way to encouraging further research in developing AI-based QA systems for real-world applications in the wind industry and beyond towards tackling climate change by helping make present-day energy systems smarter and more reliable. In the next chapter (Chapter 8), we would conclude the thesis and discuss its key strengths and limitations, along with possible directions for future research in XAI for the wind industry.

[14] Demonstration of the QA system : https://www.youtube.com/watch?v=Mx9SIW_7FsE

Chapter 8

Conclusion

> There is no finish line. When you reach one goal, find a new one.
>
> Chuck Norris

This thesis has explored the potential of XAI techniques in providing explainable decision support during O&M of wind turbines. By revolving around three key strands of XAI – DL, NLG and KGs, the thesis has demonstrated that there are a plethora of possibilities in leveraging such models for achieving transparent decisions in the wind industry. Our experiments conducted with operational data from real-world wind turbines have shown that integrating DL models with conventional ML techniques towards developing specialised hybrid algorithms can provide learning models which are highly accurate as well as transparent. In particular, we have demonstrated the applications of such models in explainable anomaly prediction during CBM even under circumstances with lack of historical turbine data in new domains. Additionally, we have shown that Deep Reinforcement Learning techniques can assist turbine engineers and technicians in offshore vessel transfer planning based on the blessings of efficient performance and trustworthiness that XAI can instil in the traditionally black-box neural networks.

To identify novel insights during data-driven decision making, we have explored causal inference techniques towards determining causal relationships between SCADA features that can provide an indication of hidden confounders which DL models focus on during anomaly prediction. The thesis has then applied data-to-text generation models for NLG, towards generating human-intelligible alarm messages and maintenance action strategies during O&M, which can provide for further fine-grained decisions. Finally, we have developed a domain-specific multimodal KG for O&M by utilising SCADA data, alarm logs and turbine maintenance manuals, which was then utilised for automated reasoning by providing a natural language interface to engineers and technicians for querying the KG through an interactive QA system. Note that the thesis has evolved over three stages – from explainable anomaly prediction in turbine sub-components to novel insights during O&M and finally automated reasoning through QA. All these stages have focused on the central theme of XAI, which has been instilled into the various chapters.

The key strengths of this thesis are outlined below:-

- Multiple strands of XAI (DL, NLG and causal inference) have been explored and their promising avenues demonstrated by experiments conducted with real-world operational turbine data.
- To the best of our knowledge, there is no existing study in the wind industry which focuses on achieving human-intelligible decisions during CBM through generation of natural language messages and O&M reports – which this thesis has tried to address.
- The thesis has finally culminated in a full-fledged QA system that leverages domain-specific KG for interactive decision support, which can potentially serve as a software application for deployment in the wind industry to support automated reasoning.
- Various resources (including the domain-specific KG, natural language templates etc.) that were developed by the author over the course of this thesis have been made open-source, which can hopefully encourage future research and development in the domain of XAI for the wind industry.

Despite the several strengths of this thesis, there are some key limitations which we would like to mention:-

- A significant challenge which this thesis faces is the development of DL models under constraints of limited data – which, despite having demonstrated promising results are not sufficient for real-world deployment in the wind industry in a fully automated manner. As we are presently in the age of big data having seen immense success in domains like computer vision and NLP, the data from wind turbines is still comparatively significantly smaller to build trustworthy decision support systems that can be used to train AI models in safety-critical applications, as in the wind industry.
- To develop AI models on the face of limited data, this thesis has explored oversampling techniques such as SMOTE, achieving highly promising results during anomaly

prediction. However, we believe that oversampling cannot still be the best substitute for extrapolating turbine operational data, which can, at times suffer from sudden fluctuations/rarer fault events that the oversampled data may not be able to reflect.

- Some aspects of the QA system which has been developed in this thesis have focused on leveraging open-domain data to extrapolate domain-specific information in the wind industry through paraphrase generation for data augmentation. This has demonstrated promising results, but there is still a possibility for the models to make inaccurate judgements in situations when rarer fault events or operational inconsistencies occur in the turbine that are not present in the training data.
- The transparent decisions provided by the XAI models are not always accurate, and there are also some specific relationships (e.g. during causal inference) which are difficult for humans to interpret as they are identified by the model during the learning process. As human rationale and judgements can at times differ from the rationale which the model utilises to arrive at its decisions, we believe that human engineers would still need to make the final decisions on whether or not to consider the model's decisions during specific fault events. However, this isn't a major drawback in comparison to the other challenges mentioned above – as identifying novel insights and getting a better understanding of an AI model's learning process can potentially help engineers and technicians to better tackle the pitfalls in using AI during O&M.

We believe that these challenges can potentially be overcome in the near future with further developments in data-driven decision making in the wind industry. As new data sources (e.g. SCADA data with historical alarms) and various types of information (e.g. domain-specific KGs) continue to emerge from multiple wind farms and from turbines which have reached the end of their useful life, the AI community will have greater resources to train ML models, providing better performance as well as transparent decisions without the requirement of oversampling and data augmentation techniques. Additionally, the NLG models and the QA system can potentially be refined further by human optimisation through a feedback process, and the learning models can continuously be updated to account for rarer fault events and alarms, inconsistencies etc. during O&M. As wind turbine O&M is a safety-critical application, we believe that further research on developing new datasets and resources, exploring other avenues of XAI and leveraging transfer learning techniques for new domains can significantly help AI to transition from academic labs to the wind industry.

Bibliography

Martín Abadi and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

I. Abdallah, V. Dertimanis, H. Mylonas, K. Tatsis, E. Chatzi, N. Dervilis, K. Worden, and E. Maguire. Fault diagnosis of wind turbine structures using decision tree learning algorithms with big data. In *Safety and Reliability – Safe Societies in a Changing World, Proceedings of the European Safety and Reliability Conference*, pages 3053–3061, Trondheim, Norway, June 2018a.

Imad Abdallah, Vasilis Dertimanis, and Eleni Chatzi. An autonomous real-time decision tree framework for monitoring & diagnostics on wind turbines. In *2nd International Conference on Wind Energy Harvesting (WINERCOST)*, pages 149–152, Catanzaro, Italy, 03 2018b.

Katrin Affolter, Kurt Stockinger, and A. Bernstein. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal*, 28:793 – 819, 2019.

Ahmad Aghaebrahimian and Filip Jurčíček. Open-domain factoid question answering via knowledge graph search. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 22–28, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-0104. URL https://www.aclweb.org/anthology/W16-0104.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1620. URL https://www.aclweb.org/anthology/P19-1620.

Lina Alhmoud and Hussein Al-Zoubi. Iot applications in wind energy conversion systems. *Open Engineering*, 9(1):490–499, 2019. doi: https://doi.org/10.1515/eng-2019-0061. URL https://www.degruyter.com/view/journals/eng/9/1/article-p490.xml.

Anwaar Ali, Junaid Qadir, Raihan ur Rasool, Arjuna Sathiaseelan, Andrej Zwitter, and Jon Crowcroft. Big data for development: applications and techniques. *Big Data Analytics*, 1 (2), July 2016. doi: 10.1186/s41044-016-0002-4.

Miltiadis Allamanis, Hao Peng, and Charles A. Sutton. A convolutional attention network for extreme summarization of source code. In *Proceedings of the 33nd International*

Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, pages 2091–2100, 2016.

Maxime Allard. What is a transformer? - inside machine learning, July 2019. URL https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04.

Md. Zahangir Alom, Tarek Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Nasrin, Mahmudul Hasan, Brian Essen, Abdul Awwal, and Vijayan Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8:292, 03 2019. doi: 10.3390/electronics8030292.

Kamal Aly, Vaishali Sohoni, S. C. Gupta, and R. K. Nema. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. *Journal of Energy*, 2016:8519785, 2016. doi: 10.1155/2016/8519785. URL https://doi.org/10.1155/2016/8519785.

Plamen Angelov and Eduardo Soares. Towards explainable deep neural networks (xdnn). *Neural Networks*, 130:185–194, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.07.010. URL http://www.sciencedirect.com/science/article/pii/S0893608020302513.

Rohan Anil, Vineet Gupta, Tomer Koren, and Yoram Singer. Memory efficient adaptive optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9749–9758. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9168-memory-efficient-adaptive-optimization.pdf.

Massimo Aria and Corrado Cuccurullo. bibliometrix: An r-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4):959–975, 2017. ISSN 1751-1577. doi: https://doi.org/10.1016/j.joi.2017.08.007. URL http://www.sciencedirect.com/science/article/pii/S1751157717300500.

Muhammad Arshad and Brendan O'Kelly. Global status of wind power generation: theory, practice, and challenges. *International Journal of Green Energy*, 16(14):1073–1090, 2019. doi: 10.1080/15435075.2019.1597369. URL https://doi.org/10.1080/15435075.2019.1597369.

Ashish Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017.

Davide Astiaso Garcia, Daniele Groppi, and Siamak Tavakoli. Developing and testing a new tool to foster wind energy sector industrial skills. *Journal of Cleaner Production*, page 124549, 2020. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2020.124549. URL http://www.sciencedirect.com/science/article/pii/S0959652620345935.

Martin Bach-Andersen, Bo Romer-Odgaard, and Ole Winther. Scalable systems for early fault detection in wind turbines : A data driven approach. In *European Wind Energy Association Annual Conference and Exhibition*, pages 382–390, Paris, France, November 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

Ramakrishna Balijepalli, V.P. Chandramohan, and K. Kirankumar. Wind turbine control methods, 03 2018. ISSN 2213-1388. URL https://www.ni.com/en-gb/innovations/white-papers/08/wind-turbine-control-methods.html.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2019.12.012. URL http://www.sciencedirect.com/science/article/pii/S1566253519308103.

Jens Baum, Christoph Laroque, Benjamin Oeser, Anders Skoogh, and Mukund Subramaniyan. Applications of big data analytics and related technologies in maintenance—literature-based research. *Machines*, 6(4):54, 2018. doi: 10.3390/machines6040054.

G. Bebis and M. Georgiopoulos. Feed-forward neural networks. *IEEE Potentials*, 13(4): 27–31, 1994.

Eric Bechhoefer, Brandon Hecke, and David He. Processing for improved spectral analysis. In *Annual Conference of Prognostics and Health Management Society*, volume 4, pages 1–6, October 2013.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Y. Bengio, P. Frasconi, and P. Simard. The problem of learning long-term dependencies in recurrent networks. *IEEE International Conference on Neural Networks*, 1993a. doi: 10.1109/icnn.1993.298725.

Y. Bengio, Paolo Frasconi, and Patrice Simard. Problem of learning long-term dependencies in recurrent networks. pages 1183–1188 vol.3, 02 1993b. doi: 10.1109/ICNN.1993.298725.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science Neural Networks: Tricks of the Trade*, page 437–478, 2012. doi: 10.1007/978-3-642-35289-8_26.

Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1133. URL https://www.aclweb.org/anthology/P14-1133.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1160.

Roheet Bhatnagar. Machine learning and big data processing: A technological perspective and review. In Aboul Ella Hassanien, Mohamed F. Tolba, Mohamed Elhoseny, and Mohamed Mostafa, editors, *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 468–478, Cham, 2018. Springer International Publishing. ISBN 978-3-319-74690-6.

Jordan J. Bird, Anikó Ekárt, and Diego R. Faria. Chatbot interaction with artificial intelligence: Human data augmentation with t5 and language transformer ensemble for text classification, 2020.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

Linda Dale. Bloomberg and Marie Volpe. *Completing your qualitative dissertation a roadmap from beginning to end*. SAGE, 2016.

Anders Bech Borchersen, Jesper Abildgaard Larsen, and Jakob Stoustrup. Fault analysis of wind turbines based on error messages and work orders. *I F A C Workshop Series*, 47(3): 4316–4321, November 2012. ISSN 1474-6670. 10th European Workshop on Advanced Control and Diagnosis 2012 ; Conference date: 08-11-2012 Through 09-11-2012.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks, 2015.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven Scott. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9:247–274, 03 2015. doi: 10.1214/14-AOAS788.

J. Browell, C. Gilbert, and D. McMillan. Use of turbine-level data for improved wind power forecasting. In *2017 IEEE Manchester PowerTech*, pages 1–6, 2017.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. Ask the right questions: Active question reformulation with reinforcement learning. In *Sixth International Conference on Learning Representations (ICLR)*, Vancouver, Canada, May 2018. URL https://openreview.net/forum?id=S1CChZ-CZ.

Widodo Budiharto, Vincent Andreas, and Alexander Gunawan. Deep learning-based question answering system for intelligent humanoid robot. *Journal of Big Data*, 7:77, 09 2020. doi: 10.1186/s40537-020-00341-6.

Qingqing Cai and Alexander Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 423–433, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-1042.

Marco Antonio Calijorne Soares and Fernando Silva Parreiras. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University - Computer and Information Sciences*, 32(6):635–646, 2020. ISSN 1319-1578. doi: https://doi.org/10.1016/j.jksuci.2018.08.005. URL https://www.sciencedirect.com/science/article/pii/S131915781830082X.

M. Canizo, E. Onieva, A. Conde, S. Charramendieta, and S. Trujillo. Real-time predictive maintenance for wind turbines using big data frameworks. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 70–77, 2017. doi: 10.1109/ICPHM.2017.7998308.

James Carroll, Alasdair McDonald, Iain Dinwoodie, David McMillan, Matthew Revie, and Iraklis Lazakis. Availability, operation and maintenance costs of offshore wind turbines with different drive train configurations. *Wind Energy*, 20:361–378, 2016.

James Carroll, Sofia Koukoura, Alasdair Mcdonald, Anastasis Charalambous, Stephan Weiss, and Stephen McArthur. Wind turbine gearbox failure and remaining useful life prediction using machine learning techniques. *Wind Energy*, 22, 11 2018. doi: 10.1002/we.2290.

Rishav Chakravarti, Anthony Ferritto, Bhavani Iyer, Lin Pan, Radu Florian, Salim Roukos, and Avi Sil. Towards building a robust industry-scale question answering system. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 90–101, Online, December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-industry.9. URL https://www.aclweb.org/anthology/2020.coling-industry.9.

Yassine Charabi and Sabah Abdul-wahab. Wind turbine performance analysis for energy cost minimization. *Renewables: Wind, Water, and Solar*, 7, 08 2020. doi: 10.1186/s40807-020-00062-7.

Joyjit Chatterjee. Explainable ai for intelligent decision support in operations maintenance of wind turbines. In *Proceedings of the Doctoral Consortium, European Conference on Artificial Intelligence (ECAI)*, Santiego, Spain, August 2020.

Joyjit Chatterjee and Nina Dethlefs. Natural language generation for operations and maintenance in wind turbines. In *NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning*, Vancouver, Canada, December 2019a.

Joyjit Chatterjee and Nina Dethlefs. Transparent deep learning and transductive transfer learning; a new dimension for wind energy research. In *WindEurope Offshore*, Copenhagen, Denmark, November 2019b.

Joyjit Chatterjee and Nina Dethlefs. A deep learning approach towards prediction of faults in wind turbines. In *Northern Lights Deep Learning Workshop*, Tromsø, Norway, January 2019c.

Joyjit Chatterjee and Nina Dethlefs. The promise of causal reasoning in reliable decision support for wind turbines. In *Fragile Earth: Data Science for a Sustainable Planet Workshop (FEED) at 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, San Diego, California, USA, August 2020a.

Joyjit Chatterjee and Nina Dethlefs. Deep reinforcement learning for maintenance planning of offshore vessel transfer. In *Proceedings of the 4th International Conference on Renewable Energies Offshore (RENEW 2020)*, pages 435–443, Lisbon, Portugal, October 2020b.

Joyjit Chatterjee and Nina Dethlefs. Temporal causal inference in wind turbine scada data using deep learning for explainable AI. *Journal of Physics: Conf Series*, September 2020c.

Joyjit Chatterjee and Nina Dethlefs. A dual transformer model for intelligent decision support for maintenance of wind turbines. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, Glasgow (UK), July 2020d.

Joyjit Chatterjee and Nina Dethlefs. Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines. *Wind Energy*, pages 1–18, April 2020e.

Joyjit Chatterjee and Nina Dethlefs. Xai4wind: A multimodal knowledge graph database for explainable decision support in operations & maintenance of wind turbines, 2021a.

Joyjit Chatterjee and Nina Dethlefs. Scientometric review of artificial intelligence for operations maintenance of wind turbines: The past, present and future. *Renewable and Sustainable Energy Reviews*, 144:111051, 2021b. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2021.111051. URL https://www.sciencedirect.com/science/article/pii/S1364032121003403.

Joyjit Chatterjee and Nina Dethlefs. Automated question-answering for interactive decision support in operations & maintenance of wind turbines. *n.a.*, n.d.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. doi: 10.1613/jair.953.

Chaomei Chen. Citespace ii: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377, 2006. doi: 10.1002/asi.20317. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20317.

Chiung-Yu Chen, Weiming Yao, Zishun Liu, Zhaoyu Zhu, Yan-Gang Zhao, and Tianhe Chi. Measuring impacts of urban environmental elements on housing prices based on multisource data—a case study of shanghai, china. *ISPRS International Journal of Geo-Information*, 9:106, 02 2020a. doi: 10.3390/ijgi9020106.

David Chen, Joohyun Kim, and Raymond Mooney. Training a Multilingual Sportscaster: Using Perceptual Context to Learn Language. *Journal of Artificial Intelligence Research*, 37:397–435, 2010.

Hugh Chen, Scott Lundberg, and Su-In Lee. Hybrid Gradient Boosting Trees and Neural Networks for Forecasting Operating Room Data. *arXiv e-prints*, art. arXiv:1801.07384, January 2018.

J. Chen and X. Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019. doi: 10.1109/JPROC.2019.2921977.

Jianjun Chen, Weihao Hu, Di Cao, Bin Zhang, Qi Huang, Zhe Chen, and Frede Blaabjerg. An imbalance fault detection algorithm for variable-speed wind turbines: A deep learning approach. *Energies*, 12(14), 2019a. ISSN 1996-1073. doi: 10.3390/en12142764. URL https://www.mdpi.com/1996-1073/12/14/2764.

Niya Chen, Zheng Qian, Ian T. Nabney, and Xiaofeng Meng. Wind power forecasts using gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems*, 29(2):656–665, March 2014. doi: 10.1109/tpwrs.2013.2282366. URL https://doi.org/10.1109%2Ftpwrs.2013.2282366.

S. Chen, H. He, and E. A. Garcia. Ramoboost: Ranked minority oversampling in boosting. *IEEE Transactions on Neural Networks*, 21(10):1624–1642, 2010. doi: 10.1109/TNN.2010.2066988.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016a. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16*, 2016b. doi: 10.1145/2939672.2939785.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, 2015.

Yiran Chen, Yuan Xie, Linghao Song, Fan Chen, and Tianqi Tang. A survey of accelerator architectures for deep neural networks. *Engineering*, 6(3):264–274, 2020b. ISSN 2095-8099. doi: https://doi.org/10.1016/j.eng.2020.01.007. URL http://www.sciencedirect.com/science/article/pii/S2095809919306356.

Zhiyu Chen, Harini Eavani, Yinyin Liu, and William Yang Wang. Few-shot nlg with pre-trained language model. *CoRR*, arXiv:1904.09521, 2019b.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online, July 2020c. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.18. URL https://www.aclweb.org/anthology/2020.acl-main.18.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014a. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014b.

Hana Chockler, Pascal Kesseli, Daniel Kroening, and Ofer Strichman. Learning the language of software errors. *Journal of Artificial Intelligence Research*, 67:881–903, apr 2020. doi: 10.1613/jair.1.11798. URL https://doi.org/10.1613%2Fjair.1.11798.

François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

G. Clark, M. Doran, and W. Glisson. A malicious attack on the machine learning policy of a robotic system. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 516–521, 2018. doi: 10.1109/TrustCom/BigDataSE.2018.00079.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. Driving semantic parsing from the world's response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 18–27, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-2903.

A Clifton, L Kilcher, J K Lundquist, and P Fleming. Using machine learning to predict wind turbine power output. *Environmental Research Letters*, 8(2):024009, April 2013a. doi: 10.1088/1748-9326/8/2/024009. URL https://doi.org/10.1088%2F1748-9326%2F8%2F2%2F024009.

A. Clifton, L. Kilcher, JK. Lundquist, and P. Fleming. Using machine learning to predict wind turbine power output. In *Environmental Research Letters, Volume 8, Number 2*, pages 1–8. IOP Publishing Ltd., 2013b.

A. Clifton, M.H. Daniels, and M. Lehning. Effect of winds in a mountain pass on turbine performance. *Wind Energy*, 17:1543–1562, 2014.

David Cox and Thomas Dean. Neural networks and neuroscience-inspired computer vision. *Current biology : CB*, 24:R921–R929, 09 2014. doi: 10.1016/j.cub.2014.08.026.

Chris Cummins, Pavlos Petoumenos, Zheng Wang, and Hugh Leather. Synthesizing benchmarks for predictive modeling. In *Proceedings of the 2017 International Symposium on Code Generation and Optimization*, CGO '17, pages 86–99, Austin, USA, 2017. IEEE Press. ISBN 978-1-5090-4931-8.

Yalcin Dalgic, Iraklis Lazakis, and Osman Turan. Investigation of optimum crew transfer vessel fleet for offshore wind farm maintenance operations. *Wind Engineering*, 39:31–52, 02 2015. doi: 10.1260/0309-524X.39.1.31.

Jeanne E. Daniel, Willie Brink, Ryan Eloff, and Charles Copley. Towards automating healthcare question answering in a noisy multilingual low-resource setting. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 948–953, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1090. URL https://www.aclweb.org/anthology/P19-1090.

Rafael Dawid, David Mcmillan, and Matthew Revie. Decision support tool for offshore wind farm vessel routing under uncertainty. *Energies*, 11:2190, 08 2018. doi: 10.3390/en11092190.

Wim De Mulder, Steven Bethard, and Marie-Francine Moens. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech Language*, 30(1):61–98, 2015. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2014.09.005. URL http://www.sciencedirect.com/science/article/pii/S088523081400093X.

Gustavo de Novaes Pires Leite, Alex Maurício Araújo, and Pedro André Carvalho Rosas. Prognostic techniques applied to maintenance of wind turbines: a concise and specific review. *Renewable and Sustainable Energy Reviews*, 81:1917–1925, 2018. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2017.06.002. URL http://www.sciencedirect.com/science/article/pii/S1364032117309383.

Dina Demner-Fushman, Wendy W. Chapman, and Clement J. McDonald. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42(5):760–772, 2009. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2009.08.007. URL http://www.sciencedirect.com/science/article/pii/S1532046409001087. Biomedical Natural Language Processing.

N. Dethlefs. Domain Transfer for Deep Natural Language Generation from Abstract Meaning Representations. *IEEE Computational Intelligence Magazine: Special Issue on Natural Language Generation with Computational Intelligence*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Amir Dezfouli, Kristi Griffiths, Fabio Ramos, Peter Dayan, and Bernard W. Balleine. Models that learn how humans learn: The case of decision-making and its disorders. *PLOS Computational Biology*, 15(6):1–33, 06 2019. doi: 10.1371/journal.pcbi.1006903. URL https://doi.org/10.1371/journal.pcbi.1006903.

Li Dong and Mirella Lapata. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1068. URL https://www.aclweb.org/anthology/P18-1068.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1091. URL https://www.aclweb.org/anthology/D17-1091.

M. Du, S. Ma, and Q. He. A scada data based anomaly detection method for wind turbines. In *China International Conference on Electricity Distribution*, pages 1–6, Xi'an, China, August 2016. IEEE.

Qianjin Du, Weixi Gu, Lin Zhang, and Shao-Lun Huang. Attention-based lstm-cnns for time-series classification. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, SenSys '18, page 410–411, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359528. doi: 10.1145/3274783.3275208. URL https://doi.org/10.1145/3274783.3275208.

Neha Dubey, Sutanu Chakraborti, and Deepak Khemani. Content selection for time series summarization using case-based reasoning. In Keith Brawner and Vasile Rus, editors, *Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference, FLAIRS 2018, Melbourne, Florida, USA. May 21-23 2018*, pages 395–398. AAAI Press, 2018. URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS18/paper/view/17673.

Aman Dureja and Payal Pahwa. Analysis of nonlinear activation functions for classification tasks using convolutional neural networks. In Sukumar Mishra, Yog Raj Sood, and Anuradha Tomar, editors, *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, pages 1179–1190, Singapore, 2019. Springer Singapore. ISBN 978-981-13-6772-4.

Ondrej Dusek and Filip Jurcicek. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, 2016.

S. Dutta and T. J. Overbye. A clustering based wind farm collector system cable layout design. In *IEEE Power and Energy Conference at Illinois*, pages 1–6, Champaign, IL, USA, February 2011.

Sanjay K. Dwivedi and Vaishali Singh. Research and reviews in question answering system. *Procedia Technology*, 10:417–424, 2013. ISSN 2212-0173. doi: https://doi.org/10.1016/j.protcy.2013.12.378. URL https://www.sciencedirect.com/science/article/pii/S2212017313005409. First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013.

E. Bechhoefer. Wind Turbine High-Speed Bearing Prognosis Dataset via MathWorks. https://uk.mathworks.com/help/predmaint/ug/wind-turbine-high-speed-bearing-prognosis.html, October 2013. Accessed: 2021-01-09.

E. Bechhoefer. Wind Turbine High-Speed Bearing Prognosis Dataset via GitHub. https://github.com/mathworks/WindTurbineHighSpeedBearingPrognosis-Data, 2018. Accessed: 2021-01-09.

Nees Jan Van Eck and Ludo Waltman. Software survey: Vosviewer, a computer program for bibliometric mapping. *Scientometrics*, 84(2):523–538, 2009. doi: 10.1007/s11192-009-0146-3.

EDP Renewables. EDPR Wind Farm Data. https://opendata.edp.com/pages/windfarms/, 2018a. Accessed: 2021-01-10.

EDP Renewables. Hack The Wind - Wind Turbine Failures Detection Competition. https://opendata.edp.com/pages/hackthewind/description#description, 2018b. Accessed: 2021-01-10.

Elozino Egonmwan and Yllias Chali. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 249–255, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5627. URL https://www.aclweb.org/anthology/D19-5627.

Evangelos Eleftheriou, Manuel Gallo, Nandakumar S.R., Christophe Piveteau, Irem Boybat, Vinay Joshi, Riduan Khaddam-Aljameh, Martino Dazzi, Iason Giannopoulos, Geethan Karunaratne, Benedikt Kersting, Milos Stanisavijevic, Vara Jonnalagadda, Nikolas Ioannou, Kornilios Kourtis, Pier Francese, and Abu Sebastian. Deep learning acceleration based on in-memory computing. *IBM Journal of Research and Development*, PP:1–1, 10 2019. doi: 10.1147/JRD.2019.2947008.

Walaa A. Elnozahy, Ghada A. El Khayat, Lilia Cheniti-Belcadhi, and Bilal Said. Question answering system to support university students' orientation, recruitment and retention. *Procedia Computer Science*, 164:56–63, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.12.154. URL https://www.sciencedirect.com/science/article/pii/S1877050919321933. CENTERIS 2019 - International Conference on ENTERprise Information Systems / ProjMAN 2019 - International Conference on Project MANagement / HCist 2019 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2019.

Unai Elosegui, Igor Egaña, Alain Ulazia, and Gabriel Berastegi. Pitch angle misalignment correction based on benchmarking and laser scanner measurement in wind farms. *Energies*, 11:3357, 12 2018. doi: 10.3390/en11123357.

Engie Renewables. La Haute Borne Data. https://opendata-renewables.engie.com/. Accessed: 2020-07-26.

Dr Mohd Faizal Fauzan, R K Chelvan, and A Amirah. Energy, economic and environmental impact of wind power in malaysia. *International Journal of Advanced Scientific Research and Management*, 2:81–87, 07 2017.

Telmo Felgueira, Silvio Rodrigues, Christian S. Perone, and Rui Castro. The Impact of Feature Causality on Normal Behaviour Models for SCADA-based Wind Turbine Fault Detection. *Climate Change: How Can AI Help? ICML, California, USA*, abs/1906.12329, 2019. URL http://arxiv.org/abs/1906.12329.

Christiane Fellbaum. *WordNet*. American Cancer Society, 2012. ISBN 9781405198431. doi: https://doi.org/10.1002/9781405198431.wbeal1285. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781405198431.wbeal1285.

Jinyue Feng, Chantal Shaib, and Frank Rudzicz. Explainable clinical decision support from text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1478–1489, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.115. URL https://www.aclweb.org/anthology/2020.emnlp-main.115.

Yanhui Feng, Yingning Qiu, Christopher Crabtree, Hui Long, and P.J. Tavner. Monitoring wind turbine gearboxes. *Wind Energy*, 16, 07 2013. doi: 10.1002/we.1521.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.139. URL https://www.aclweb.org/anthology/2020.findings-emnlp.139.

Zhipeng Feng, Ming Liang, Yi Zhang, and Shumin Hou. Fault diagnosis for wind turbine planetary gearboxes via demodulation analysis based on ensemble empirical mode decomposition and energy separation. *Renewable Energy*, 47(C):112–126, 2012. doi: 10.1016/j.renene.2012.04. URL https://ideas.repec.org/a/eee/renene/v47y2012icp112-126.html.

Borja Fernandez-Gauna, Unai Fernandez-Gamiz, and Manuel Graña. Variable speed wind turbine controller adaptation by reinforcement learning. *Integrated Computer-Aided Engineering*, 24:1–13, 08 2016. doi: 10.3233/ICA-160531.

A. M. Foley, P. G. Leahy, and E. J. McKeogh. Wind power forecasting prediction methods. In *2010 9th International Conference on Environment and Electrical Engineering*, pages 61–64, 2010.

Laura Folguera, Jure Zupan, Daniel Cicerone, and Jorge F. Magallanes. Self-organizing maps for imputation of missing data in incomplete data matrices. *Chemometrics and Intelligent Laboratory Systems*, 143:146–151, 2015. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2015.03.002. URL http://www.sciencedirect.com/science/article/pii/S016974391500060X.

Ruben Foresti, Stefano Rossi, Matteo Magnani, Corrado Guarino Lo Bianco, and Nicola Delmonte. Smart society and artificial intelligence: Big data scheduling and the global standard method applied to smart maintenance. *Engineering*, 2020. ISSN 2095-8099. doi: https://doi.org/10.1016/j.eng.2019.11.014. URL http://www.sciencedirect.com/science/article/pii/S2095809920300266.

Markus Freitag and Scott Roy. Unsupervised natural language generation with denoising autoencoders. *CoRR*, abs/1804.07899, 2018.

X. Fu, F. Gao, J. Wu, X. Wei, and F. Duan. Spatiotemporal attention networks for wind power forecasting. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 149–154, 2019.

Yao Fu, Yansong Feng, and John P Cunningham. Paraphrase generation with latent bag of words. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5e2b66750529d8ae895ad2591118466f-Paper.pdf.

Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1610. URL https://www.aclweb.org/anthology/P19-1610.

Fausto Pedro García Márquez, Andrew Mark Tobias, Jesús María Pinar Pérez, and Mayorkinos Papaelias. Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178, 2012. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2012.03.003. URL http://www.sciencedirect.com/science/article/pii/S0960148112001899.

Claire Gardent, Anastasia Shimorina, Shashi Narayana, and Laura Perez-Beltrachini. The WebNLG Challenge: Generating Text from RDF Data. In *Proceedings of the International Natural Language Generation Conference (INLG)*, page 124–133, Santiago de Compostella, Spain, 2017.

Konstantina Garoufi and Alexander Koller. Automated planning for situated natural language generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1573–1582, 2010.

Y. Ge, D. Yue, and L. Chen. Prediction of wind turbine blades icing based on mbk-smote and random forest in imbalanced data set. In *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pages 1–6, 2017. doi: 10.1109/EI2.2017.8245530.

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands, 2018.

Guillaume Genthial. Seq2seq with attention and beam search, November 2017. URL https://guillaumegenthial.github.io/sequence-to-sequence.html.

F.A. Gers. Learning to forget: continual prediction with lstm. *IET Conference Proceedings*, pages 850–855(5), January 1999. URL https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218.

Indraneel Ghosh. Recurrent neural networks: A man-made copy of the human brain, August 2019. URL http://www.freelunch.co.in/recurrent-neural-networks-a-man-made-copy-of-the-human-brain/.

Gregor Giebel and C. Hasager. *An Overview of Offshore Wind Farm Design*, pages 337–346. MARE-WINT. Springer, Cham, 08 2016. ISBN 978-3-319-39094-9. doi: 10.1007/978-3-319-39095-6_19.

Jamie Godwin and Peter Matthews. Classification and detection of wind turbine pitch faults through scada data analysis. *International Journal of Prognostics and Health Management*, 4, 01 2013.

Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *Plos One*, 11(4), 2016. doi: 10.1371/journal.pone.0152173.

Li Gong, Josep Crego, and Jean Senellart. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong, November 2019. Association for Computational Linguistics.

E Gonzalez, M Reder, and J J Melero. SCADA alarms processing for wind turbine component failure detection. *Journal of Physics: Conference Series*, 753:072019, September 2016. doi: 10.1088/1742-6596/753/7/072019. URL https://doi.org/10.1088/1742-6596/753/7/072019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. *The MIT Press, Cambridge, Mass*, 2017.

Travis R. Goodwin and Sanda M. Harabagiu. Medical question answering for clinical decision support. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 297–306, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340731. doi: 10.1145/2983323.2983819. URL https://doi.org/10.1145/2983323.2983819.

A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T. Yang, and E. Choi. Morphnet: Fast simple resource-constrained structure learning of deep networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1586–1595, 2018. doi: 10.1109/CVPR.2018.00171.

A. Goudarzi, I. E. Davidson, A. Ahmadi, and G. K. Venayagamoorthy. Intelligent analysis of wind turbine power curve models. In *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*, pages 1–7, 2014.

A. Graves, N. Jaitly, and A. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278, 2013. doi: 10.1109/ASRU.2013.6707742.

Alex Graves. Generating Sequences With Recurrent Neural Networks. *CoRR*, abs/1308.0850, 2013. URL http://arxiv.org/abs/1308.0850.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL https://www.sciencedirect.com/science/article/pii/S0893608005001206. IJCNN 2005.

Enzo Grossi and Massimo Buscema. Introduction to artificial neural networks. *European journal of gastroenterology hepatology*, 19:1046–54, 01 2008. doi: 10.1097/MEG.0b013e3282f198a0.

Shivani Gupta and Atul Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.11.146. URL http://www.sciencedirect.com/science/article/pii/S1877050919318575. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.

Ivan Habernal and Miloslav Konopík. Swsnl: Semantic web search using natural language. *Expert Systems with Applications*, 40(9):3649–3664, 2013. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2012.12.070. URL https://www.sciencedirect.com/science/article/pii/S0957417412013115.

Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008. doi: 10.1109/IJCNN.2008.4633969.

Sonia Haiduc, Jairo Aponte, Laura Moreno, and Andrian Marcus. On the use of automated text summarization techniques for summarizing source code. In *Proceedings of the 2010 17th Working Conference on Reverse Engineering*, WCRE '10, pages 35–44, Washington, DC, USA, 2010. IEEE Computer Society.

G. J. D. R. Hains, Y. Khmelevsky, and T. Tachon. From natural language to graph queries. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–4, 2019. doi: 10.1109/CCECE.2019.8861892.

Elin Halvorsen-Weare, Christian Gundegjerde, Ina Halvorsen, Lars Magnus Hvattum, and Lars Nonås. Vessel fleet analysis for maintenance operations at offshore wind farms. *Energy Procedia*, 35:167–176, 08 2013. doi: 10.1016/j.egypro.2013.07.170.

Yong-Jin Han, Seong-Bae Park, and Se-Young Park. A natural language interface concordant with a knowledge base. *Intell. Neuroscience*, 2016, January 2016. ISSN 1687-5265. doi: 10.1155/2016/9174683. URL https://doi.org/10.1155/2016/9174683.

Ryoichi Hara. Chapter 9 - offshore wind. *Wind Power Generation*, pages 75–84, 2016. doi: https://doi.org/10.1016/B978-0-12-804038-6.00009-8. URL https://doi.org/10.1214/aoms/1177729586.

Sadid A. Hasan, Bo Liu, Joey Liu, Ashequl Qadir, Kathy Lee, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. Neural clinical paraphrase generation with attention. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 42–53, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/W16-4207.

Hossein Hassani and Emmanuel Silva. A kolmogorov-smirnov based test for comparing the predictive accuracy of two sets of forecasts. *Econometrics*, 3(3):590–609, 2015. doi: 10.3390/econometrics3030590.

Helen Hastie, Heriberto Cuayahuitl, Nina Dethlefs, Simon Keizer, and Xingkun Liu. Evaluation of NLG in an end-to-end Spoken dialogue system-is it worth it? In *Proc. of 7th International Workshop on Spoken Dialogue Systems*, Saariselka, Finland, 2016.

Chad Hazlett. Estimating causal effects of new treatments despite self-selection: The case of experimental medical treatments. *Journal of Causal Inference*, 7(1):1–11, December 2018. doi: 10.1515/jci-2018-0019.

D. He, Z. Wang, and J. Liu. A survey to predict the trend of ai-able server evolution in the cloud. *IEEE Access*, 6:10591–10602, 2018. doi: 10.1109/ACCESS.2018.2801293.

Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, Boston, MA, USA, 06 2015. doi: 10.1109/CVPR.2015.7299173.

Xiaodong He and David Golub. Character-level question answering with attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1598–1607, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1166. URL https://www.aclweb.org/anthology/D16-1166.

Y. He and A. Kusiak. Performance assessment of wind turbines: Data-derived quantitative metrics. *IEEE Transactions on Sustainable Energy*, 9(1):65–73, 2018. doi: 10.1109/TSTE.2017.2715061.

Georg Helbing and Matthias Ritter. Deep Learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98(C):189–198, 2018a. doi: 10.1016/j.rser.2018.09.01. URL https://ideas.repec.org/a/eee/rensus/v98y2018icp189-198.html.

Georg Helbing and Matthias Ritter. Deep learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98:189–198, 12 2018b. doi: 10.1016/j.rser.2018.09.012.

Georg Helbing and Matthias Ritter. Deep learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98:189–198, 2018c.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020. URL http://jmlr.org/papers/v21/20-312.html.

L. T. Hien, L. Tran Thi Ly, C. Pham-Nguyen, T. Le Dinh, H. Tiet Gia, and L. N. Hoai Nam. Towards chatbot-based interactive what- and how-question answering systems: the adobot approach. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–3, 2020. doi: 10.1109/RIVF48685.2020.9140742.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Overview of mini-batch gradient descent, n.d. URL https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, 1998. doi: 10.1142/s0218488598000094.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997a. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Sepp Hochreiter and JÃŒrgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997b. doi: 10.1162/neco.1997.9.8.1735.

Patrick Hohenecker and Thomas Lukasiewicz. Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research*, 68, jul 2020. doi: 10.1613/jair.1.11661. URL https://doi.org/10.1613%2Fjair.1.11661.

Yining Hong, Jialu Wang, Yuting Jia, Weinan Zhang, and Xinbing Wang. Academic reader: An interactive question answering system on academic literatures. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9855–9856, Jul. 2019. doi: 10.1609/aaai.v33i01.33019855. URL https://ojs.aaai.org/index.php/AAAI/article/view/5065.

Zhao Hongshan, Liu Huihai, Hu Wenjing, and Yan Xihui. Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renewable Energy*, 127, 05 2018. doi: 10.1016/j.renene.2018.05.024.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. doi: 10.1073/pnas.79.8.2554.

Qinghua Hu, Rujia Zhang, and Yucan Zhou. Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy*, 85(C):83–95, 2016. doi: 10.1016/j.renene.2015.06. URL https://ideas.repec.org/a/eee/renene/v85y2016icp83-95.html.

Xing Hu, Yuhan Wei, Ge Li, and Zhi Jin. Codesum: Translate program language to natural language. *CoRR*, abs/1708.01837, 2017.

Yuxiao Huang and Samantha Kleinberg. Fast and accurate causal inference from time series data, 2015. URL https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS15/paper/view/10434.

Mahbub Hussain, Jordan J. Bird, and Diego R. Faria. A study on cnn transfer learning for image classification. In Ahmad Lotfi, Hamid Bouchachia, Alexander Gegov, Caroline Langensiepen, and Martin McGinnity, editors, *Advances in Computational Intelligence Systems*, pages 191–202, Cham, 2019. Springer International Publishing. ISBN 978-3-319-97982-3.

Raed K Ibrahim, Jannis Tautz-Weinert, and Simon J Watson. Neural networks for wind turbine fault detection via current signature analysis. In *WindEurope Summit*, Hamburg, Germany, September 2016.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2073–2083, Berlin, Germany, 2016. Association for Computational Linguistics.

Paras Jain, Ajay Jain, Aniruddha Nrusimha, Amir Gholami, Pieter Abbeel, Joseph Gonzalez, Kurt Keutzer, and Ion Stoica. Checkmate: Breaking the memory wall with optimal tensor rematerialization. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 497–511, 2020. URL https://proceedings.mlsys.org/paper/2020/file/084b6fbb10729ed4da8c3d3f5a3ae7c9-Paper.pdf.

Valentina Janev, Dea Pujić, Marko Jelić, and Maria-Esther Vidal. *Chapter 9 Survey on Big Data Applications*, pages 149–164. Springer International Publishing, Cham, 2020. ISBN 978-3-030-53199-7. doi: 10.1007/978-3-030-53199-7_9. URL https://doi.org/10.1007/978-3-030-53199-7_9.

B. Jang, M. Kim, G. Harerimana, and J. W. Kim. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, 7:133653–133667, 2019.

Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, and Jong Kim. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, PP:1–1, 09 2019. doi: 10.1109/ACCESS.2019.2941229.

Jaejin Jang, Im Jung, and Jong Park. An effective handling of secure data stream in iot. *Applied Soft Computing*, 68, 05 2017. doi: 10.1016/j.asoc.2017.05.020.

José M. Jerez, Ignacio Molina, Pedro J. García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*, 50 (2):105–115, 2010. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2010.05.002. URL http://www.sciencedirect.com/science/article/pii/S0933365710000679.

Yangfeng Ji, Antoine Bosselut, Thomas Wolf, and Asli Celikyilmaz. The amazing world of neural language generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 37–42, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-tutorials.7. URL https://www.aclweb.org/anthology/2020.emnlp-tutorials.7.

Feng Jia, Yaguo Lei, Hongkai Shan, and Jing Lin. Early fault diagnosis of bearings using an improved spectral kurtosis by maximum correlated kurtosis deconvolution. *Sensors*, 15 (11):29363–29377, 2015. doi: 10.3390/s151129363.

Yuening Jia. Attention mechanism in machine translation. *Journal of Physics: Conference Series*, 1314:012186, 10 2019. doi: 10.1088/1742-6596/1314/1/012186.

G. Jiang, P. Xie, H. He, and J. Yan. Wind turbine fault detection using a denoising autoencoder with temporal information. *IEEE/ASME Transactions on Mechatronics*, 23(1):89–100, 2018. doi: 10.1109/TMECH.2017.2759301.

Hai Jin, Bo Liu, Wenbin Jiang, Yang Ma, Xuanhua Shi, Bingsheng He, and Shaofeng Zhao. Layer-centric memory reuse and data migration for extreme-scale deep learning on many-core architectures. *ACM Trans. Archit. Code Optim.*, 15(3), September 2018. ISSN 1544-3566. doi: 10.1145/3243904. URL https://doi.org/10.1145/3243904.

X. Jin, F. Cheng, Y. Peng, W. Qiao, and L. Qu. Drivetrain gearbox fault diagnosis: Vibration- and current-based approaches. *IEEE Industry Applications Magazine*, 24(6):56–66, 2018.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1014.

Mohsen Kaboli. A Review of Transfer Learning Algorithms. Research report, Technische Universität München, August 2017. URL https://hal.archives-ouvertes.fr/hal-01575126. Transfer Learning Algorithms.

Endri Kacupaj, Barshana Banerjee, Kuldeep Singh, and Jens Lehmann. Paraqa: A question answering dataset with paraphrase responses for single-turn conversation, 2021a.

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. Conversational question answering over knowledge graphs with transformer and graph attention networks, 2021b.

John Kaldellis and Dimitrios Zafirakis. The wind energy (r)evolution: A short review of a long history. *Renewable Energy*, 36:1887–1901, 07 2011. doi: 10.1016/j.renene.2011.01.002.

D. Kalyanraj, S. L. Prakash, and S. Sabareswar. Wind turbine monitoring and control systems using internet of things. In *2016 21st Century Energy Needs - Materials, Systems and Applications (ICTFCEN)*, pages 1–4, 2016. doi: 10.1109/ICTFCEN.2016.8052714.

Davood Karimi, Haoran Dou, Simon K. Warfield, and Ali Gholipour. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis*, 65:101759, 2020. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101759. URL http://www.sciencedirect.com/science/article/pii/S1361841520301237.

M. F. Kasim. Playing the game of congklak with reinforcement learning. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–5, 2016.

Paul Kassebaum. circulargraph. (https://www.github.com/paul-kassebaum-mathworks/circularGraph), 2020.

Kay H. Brodersen et al. Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, 9:247–274, 2015.

William Keller, Camille Hamon, Björn Palm, and Gaël Millet. Wind farm noise impact in france: A proposition of acoustic model improvements for predicting energy production. *Master Thesis: KTH Royal Institute of Technology*, pages 1–58, 2014.

M-G Kim, P H Dalhoff, and P Gust. Yawing characteristics during slippage of the nacelle of a multi mw wind turbine. *Journal of Physics: Conference Series*, 753:062010, 09 2016. doi: 10.1088/1742-6596/753/6/062010.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

KK Wind Solutions A/S). Rotor Imbalance Cancellation Whitepaper (kk wind solutions a/s). https://www.kkwindsolutions.com/Files/Files/WhitePapers/20160311_Rotor-Imbalance-Cancellation.pdf, 2019. Accessed: 2019-05-08.

Samantha Kleinberg. Causality, Probability, and Time. *Cambridge University Press: Cambridge, UK*, 2013.

Peter Kokol, Helena Blazun Vosner, and Jernej Zavrsnik. Application of bibliometrics in medicine: a historical bibliometrics analysis. *Health Information & Libraries Journal*, 01 2020. doi: 10.1111/hir.12295.

Alexios Koltsidopoulos Papatzimos. *Data-driven operations and maintenance for offshore wind farms: tools and methodologies*. PhD thesis, The University of Edinburgh, 01 2020.

Ziqian Kong, Baoping Tang, Lei Deng, Wenyi Liu, and Yan Han. Condition monitoring of wind turbines based on spatio-temporal fusion of scada data by convolutional neural networks and gated recurrent units. *Renewable Energy*, 146:760–768, 2020. doi: 10.1016/j.renene.2019.07.033.

Muthu Krishnan and Muthu Krishnan. Mathematics behind k-mean clustering algorithm, July 2018. URL https://muthu.co/mathematics-behind-k-mean-clustering-algorithm/.

Pravin A Kulkarni, Ashwinkumar S Dhoble, and Pramod M Padole. Deep neural network-based wind speed forecasting and fatigue analysis of a large composite wind turbine blade. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 233(8):2794–2812, 2019. doi: 10.1177/0954406218797972. URL https://doi.org/10.1177/0954406218797972.

Ibrahim Sina Kuseyri. *Condition monitoring of wind turbines: Challenges and opportunities*, pages 116–126. 2015.

Andrew Kusiak. Renewables: Share data on wind energy. *Nature*, 529(7584):19–21, 2016. doi: 10.1038/529019a.

Uday Kusupati, Venkata Ravi, and Teja Ailavarapu. Natural language to code using transformers. 2018.

Tom Kwiatkowksi, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, MA, October 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D10-1119.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1556, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1161.

Thi-Thu-Huong Le, Jihyun Kim, and Howon Kim. Analyzing effective of activation functions on recurrent neural networks for intrusion detection. *Journal of Multimedia Information System*, 3(3):91–96, 2016. doi: 10.9717/JMIS.2016.3.3.91. URL https://doi.org/10.9717/JMIS.2016.3.3.91.

Tung D. Le, Haruki Imai, Yasushi Negishi, and Kiyokuni Kawachiya. Automatic gpu memory management for large neural models in tensorflow. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on Memory Management*, ISMM 2019, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367226. doi: 10.1145/3315573.3329984. URL https://doi.org/10.1145/3315573.3329984.

Kevin Leahy, R. Lily Hu, Ioannis C. Konstantakopoulos, Costas J. Spanos, and Alice M. Agogino. Diagnosing wind turbine faults using machine learning techniques applied to operational data. In *International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8, Ottawa, ON, Canada, June 2016. IEEE.

Kevin Leahy, Colm Gallagher, Ken Bruton, Peter O'Donovan, and Dominic T.J. O'Sullivan. Automatically identifying and predicting unplanned wind turbine stoppages using SCADA and alarms system data: Case study and results. *Journal of Physics: Conference Series*, 926:012011, November 2017. doi: 10.1088/1742-6596/926/1/012011. URL https://doi.org/10.1088/1742-6596/926/1/012011.

Kevin Leahy, Colm Gallagher, Peter O'Donovan, and Dominic T. J. O'Sullivan. Issues with data quality for wind turbine condition monitoring and reliability analyses. *Energies*, 12 (2), 2019. ISSN 1996-1073. doi: 10.3390/en12020201. URL https://www.mdpi.com/1996-1073/12/2/201.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015. doi: 10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

J. C. Y. Lee, P. Stuart, A. Clifton, M. J. Fields, J. Perr-Sauer, L. Williams, L. Cameron, T. Geer, and P. Housley. The power curve working group's assessment of wind turbine power performance prediction methods. *Wind Energy Science*, 5(1):199–223, 2020. doi: 10.5194/wes-5-199-2020. URL https://wes.copernicus.org/articles/5/199/2020/.

Sil C. van de Leemput, Jonas Teuwen, Bram van Ginneken, and Rashindra Manniesing. Memcnn: A python/pytorch package for creating memory-efficient invertible neural networks. *Journal of Open Source Software*, 4(39):1576, 7 2019. ISSN 2475-9066. doi: 10.21105/joss.01576. URL http://dx.doi.org/10.21105/joss.01576.

Jinhao Lei, Chao Liu, and Dongxiang Jiang. Fault diagnosis of wind turbine based on long short-term memory networks. *Renewable Energy*, 133, 10 2018. doi: 10.1016/j.renene.2018.10.031.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

Dimitris Leventis and Dimitris Leventis. Xgboost mathematics explained, November 2018. URL https://towardsdatascience.com/xgboost-mathematics-explained-58262530904a.

Michael Lewis and Alexis Kuerbis. An overview of causal directed acyclic graphs for substance abuse researchers. *Journal of Drug and Alcohol Research*, 5:1–8, 01 2016. doi: 10.4303/jdar/235992.

D. Li, S.C.M. Ho, G. Song, L. Ren, and H. Li. A review of damage detection methods for wind turbine blades. *Smart Materials and Structures*, 24:1–24, 2015.

Hui Li, Wensong Zhou, and Jinlong Xu. Structural health monitoring of wind turbine blades. *Wind Turbine Control and Monitoring-Part of the Advances in Industrial Control book series (AIC)*, pages 231–265, 2014.

Meng Li and Shuangxin Wang. Dynamic fault monitoring of pitch system in wind turbines using selective ensemble small-world neural networks. *Energies*, 12:3256, 08 2019. doi: 10.3390/en12173256.

Mingqi Li, Xiaoyang Fu, and Dongdong Li. Diabetes prediction based on xgboost algorithm. *IOP Conference Series: Materials Science and Engineering*, 768:072093, 03 2020. doi: 10.1088/1757-899X/768/7/072093.

Y. Li. A discussion on using empirical mode decomposition for incipient fault detection and diagnosis of the wind turbine gearbox. In *2010 World Non-Grid-Connected Wind Power and Energy Conference*, pages 1–5, 2010.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1421. URL https://www.aclweb.org/anthology/D18-1421.

Percy Liang, Michael I. Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446, 2013. doi: 10.1162/COLI_a_00127. URL https://www.aclweb.org/anthology/J13-2005.

S. Liang, Kurt Stockinger, T. M. Farias, Maria Anisimova, and M. Gil. Querying knowledge graphs in natural language. *Journal of Big Data*, 8, 2021.

Cheng-Jian Lin, Jyun-Yu Jhang, Hsueh-Yi Lin, Chin-Ling Lee, and Kuu-Young Young. Using a reinforcement q-learning-based deep neural network for playing video games. *Electronics*, 8:1128, 10 2019. doi: 10.3390/electronics8101128.

Zi Lin and Xiaolei Liu. Wind power forecasting of an offshore wind turbine based on high-frequency scada data and deep learning neural network. *Energy*, 201:117693, 2020. ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2020.117693. URL http://www.sciencedirect.com/science/article/pii/S0360544220308008.

Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. Latent predictor networks for code generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 599–609, Berlin, Germany, 2016. Association for Computational Linguistics.

Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015. URL http://arxiv.org/abs/1506.00019.

Ruishan Liu and James Zou. The effects of memory replay in reinforcement learning. In *56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 478–485, 10 2018. doi: 10.1109/ALLERTON.2018.8636075.

Song Liu, Yi-Xin Zhong, and Fu-Ji Ren. Interactive question answering based on faq. In Maosong Sun, Min Zhang, Dekang Lin, and Haifeng Wang, editors, *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 73–84, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-41491-6.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI-18, pages 4881–488, 2018.

W.Y. Liu. The vibration analysis of wind turbine blade–cabin–tower coupling system. *Engineering Structures*, 56:954–957, 2013. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2013.06.008. URL http://www.sciencedirect.com/science/article/pii/S0141029613002915.

Yao Liu, Lin Guan, Chen Hou, Hua Han, Zhangjie Liu, Yao Sun, and Minghui Zheng. Wind power short-term prediction based on lstm and discrete wavelet transform. *Applied Sciences*, 9(6):1108, 2019. doi: 10.3390/app9061108.

Andrés Llombart, Carlos Pueyo, JM Fandos, and Josechu Guerrero. Robust data filtering in wind power systems. *European wind energy conference EWEC, Athens*, 2:149–54, 01 2006.

Félix Melchor Santos López and Eulogio Santos De la Cruz. Literature review about neo4j graph database as a feasible alternative for replacing rdbms. *Industrial Data*, 18:135–139, 2015.

M Lorenz, G Aisch, and D Kokkelink. Create charts and maps [software], 2012. URL https://github.com/datawrapper/datawrapper.

Yang Lu, Liping Sun, Xinyue Zhang, Feng Fenga, Jichuan Kang, and Guoqiang Fu. Condition based maintenance optimization for offshore wind turbine considering opportunities based on neural network approach. *Applied Ocean Research*, 74:69–79, 2018.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and S. Auer. Neural network-based question answering over knowledge graphs on word and character level. *Proceedings of the 26th International Conference on World Wide Web*, 2017.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. Pretrained transformers for simple question answering over knowledge graphs. *CoRR*, abs/2001.11985, 2020. URL https://arxiv.org/abs/2001.11985.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017a. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://www.aclweb.org/anthology/D15-1166.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, September 2015b. doi: 10.18653/v1/d15-1166. URL https://www.aclweb.org/anthology/D15-1166.

Sangkug Lym, Esha Choukse, Siavash Zangeneh, Wei Wen, Sujay Sanghavi, and Mattan Erez. Prunetrain: Fast neural network training by dynamic sparse model reconfiguration. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362290. doi: 10.1145/3295500.3356156. URL https://doi.org/10.1145/3295500.3356156.

F. Mairesse, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, and S. Young. Phrase-Based Statistical Language Generation Using Graphical Models and Active Learning. In *Proc. of the 48th Annual Meeting of the Association of Computational Linguistics (ACL)*, Uppsala, Sweden, 2010.

Jorge Maldonado, S. Martin-Martinez, Estefania Artigao, and Emilio Gomez-Lazaro. Using scada data for wind turbine condition monitoring: A systematic literature review. *Energies*, 13:3132, 06 2020. doi: 10.3390/en13123132.

S. Malgaonkar, S. Soral, S. Sumeet, and T. Parekhji. Study on big data analytics research domains. In *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 200–206, 2016. doi: 10.1109/ICRITO.2016.7784952.

Gian Luca Marcialis and Fabio Roli. Fusion of multiple fingerprint matchers by single-layer perceptron with class-separation loss function. *Pattern Recognition Letters*, 26(12):1830–1839, 2005. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2005.03.004. URL http://www.sciencedirect.com/science/article/pii/S0167865505000590. Artificial Neural Networks in Pattern Recognition.

Tobias Mayer, Elena Cabrio, and S. Villata. Transformer-based argument mining for healthcare applications. In *ECAI*, 2020.

Kathleen R. McKeown. Paraphrasing questions using given and new information. *Comput. Linguist.*, 9(1):1–10, January 1983. ISSN 0891-2017.

Julio Melero, Josechu Guerrero, Julian Beltran, and Carlos Pueyo. Efficient data filtering for wind energy assessment. *Renewable Power Generation, IET*, 6:446–454, 11 2012. doi: 10.1049/iet-rpg.2011.0288.

Diogo Menezes, Mateus Mendes, Jorge Alexandre Almeida, and Torres Farinha. Wind farm and resource datasets: A comprehensive survey and overview. *Energies*, 13(18), 2020. ISSN 1996-1073. doi: 10.3390/en13184702. URL https://www.mdpi.com/1996-1073/13/18/4702.

Zhaorui Meng and Xianze Xu. A hybrid short-term load forecasting framework with an attention-based encoder–decoder network based on seasonal and trend adjustment. *Energies*, 12(24), 2019. ISSN 1996-1073. doi: 10.3390/en12244612. URL https://www.mdpi.com/1996-1073/12/24/4612.

Z. Menghua, C. Zhe, and B. Freede. Generation ratio availability assessment of electrical systems for offshore wind farms. *IEEE Transactions on Energy Conversion*, 22:755–763, 2007.

Jose M. Merigo and Jian-Bo Yang. Bibliometric analysis in financial research. In *IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 223–230, 03 2014. doi: 10.1109/CIFEr.2014.6924077.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *arXiv:1609.07843*. CoRR, 2016.

Yuri Merizalde, Luis Hernández-Callejo, Oscar Duque-Perez, and Víctor Alonso-Gómez. Maintenance models applied to wind turbines. a comprehensive overview. *Energies*, 12 (2), 2019. ISSN 1996-1073. doi: 10.3390/en12020225. URL https://www.mdpi.com/1996-1073/12/2/225.

Jakob W. Messner, Pierre Pinson, Jethro Browell, Mathias B. Bjerregård, and Irene Schicker. Evaluation of wind power forecasts—an up-to-date view. *Wind Energy*, 23(6):1461–1481, 2020. doi: 10.1002/we.2497. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2497.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11): 39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

C. Minutti, S. Gomez, and G. Ramos. A machine-learning approach for noise reduction in parameter estimation inverse problems, applied to characterization of oil reservoirs. *Journal of Physics: Conference Series*, 1047:012010, June 2018. doi: 10.1088/1742-6596/1047/1/012010. URL https://doi.org/10.1088/1742-6596/1047/1/012010.

N. Mittelmeier and M. Kühn. Determination of optimal wind turbine alignment into the wind and detection of alignment changes with scada data. *Wind Energy Science*, 3(1):395–408, 2018. doi: 10.5194/wes-3-395-2018. URL https://www.wind-energ-sci.net/3/395/2018/.

N Mittelmeier, T Blodau, G Steinfeld, A Rott, and M Kühn. An analysis of offshore wind farm SCADA measurements to identify key parameters influencing the magnitude of wake effects. *Journal of Physics: Conference Series*, 753:032052, September 2016. doi: 10.1088/1742-6596/753/3/032052. URL https://doi.org/10.1088%2F1742-6596%2F753%2F3%2F032052.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2047. URL https://www.aclweb.org/anthology/N18-2047.

Diego Mollá and José Luis Vicedo. Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61, 2007. doi: 10.1162/coli.2007.33.1.41. URL https://www.aclweb.org/anthology/J07-1004.

Sahir Moreno, Miguel Pena, Alexia Toledo, Ricardo Trevino, and Hiram Ponce. A new vision-based method using deep learning for damage inspection in wind turbine blades. In *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–5, Mexico City, Mexico, September 2018.

M. Morshedizadeh. Condition monitoring of wind turbines using intelligent machine learning techniques. In *Electronic Theses and Dissertations. 6002, https://scholar.uwindsor.ca/etd/6002*, pages 1–96, University of Windsor, Ontario, Canada, 2017. University of Windsor.

Sajad Mousavi, Michael Schukat, and Enda Howley. Deep reinforcement learning: An overview. In *Proceedings of SAI Intelligent Systems Conference*, pages 426–440, 06 2018. ISBN 978-3-319-56990-1. doi: 10.1007/978-3-319-56991-8_32.

Noah J. Myrent, Joshua F. Kusnick, Douglas Adams, and Daniel T. Griffith. *Pitch Error and Shear Web Disbond Detection on Wind Turbine Blades for Offshore Structural Health and Prognostics Management*. doi: 10.2514/6.2013-1695. URL https://arc.aiaa.org/doi/abs/10.2514/6.2013-1695.

Elaheh Gholamzadeh Nabati and Klaus-Dieter Thoben. Data driven decision making in planning the maintenance activities of off-shore wind energy. *Procedia CIRP*, 59:160–165, 2017. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2016.09.026. URL http://www.sciencedirect.com/science/article/pii/S221282711630960X. Proceedings of the 5th International Conference in Through-life Engineering Services Cranfield University, 1st and 2nd November 2016.

Maryam Najafabadi, Flavio Villanustre, Taghi Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2, 12 2015. doi: 10.1186/s40537-014-0007-7.

National Renewable Energy Laboratory. Eastern Wind Dataset. https://www.nrel.gov/grid/eastern-wind-data.html, a. Accessed: 2020-07-26.

National Renewable Energy Laboratory. Western Wind Dataset. https://www.nrel.gov/grid/western-wind-data.html, b. Accessed: 2020-07-26.

National Renewable Energy Laboratory- National Wind Technology Center. Wind Turbine Gearbox Condition Monitoring Vibration Analysis Benchmarking Datasets. https://openei.org/datasets/dataset/wind-turbine-gearbox-condition-monitoring-vibration-analysis-benchmarking-datasets, March 2014. Accessed: 2021-01-09.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal Discovery with Attention-Based Convolutional Neural Networks. *Machine Learning and Knowledge Extraction*, 1(1): 312–340, July 2019. doi: 10.3390/make1010019.

Aloys Nghiem, Daniel Fraile, Ariola Mbistrova, and Tom Remy. Wind energy in europe: Outlook to 2020. *WindEurope*, September 2017.

Giang Nguyen, Stefan Dlugolinsky, Martin Bobak, Viet Tran, Alvaro Lopez Garcia, Ignacio Heredia, Peter Malík, and Ladislav Hluchý. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, 52:77–124, 06 2019. doi: 10.1007/s10462-018-09679-z.

Feng Nie, Jinpeng Wang, Jin-Ge Yao, Rong Pan, and Chin-Yew Lin. Operations guided neural networks for high fidelity data-to-text generation. *CoRR*, abs/1809.02735, 2018. URL http://arxiv.org/abs/1809.02735.

Tong Niu, Semih Yavuz, Yingbo Zhou, Huan Wang, Nitish Shirish Keskar, and Caiming Xiong. Unsupervised paraphrase generation via dynamic blocking, 2020.

Shota Ohnishi, Eiji Uchibe, Yotaro Yamaguchi, Kosuke Nakanishi, Yuji Yasui, and Shin Ishii. Constrained deep q-learning gradually approaching ordinary q-learning. *Frontiers in Neurorobotics*, 13:103, 12 2019. doi: 10.3389/fnbot.2019.00103.

ORE Catapult. Platform for Operational Data: Equinor Hywind Scotland Windfarm . https://pod.ore.catapult.org.uk/source/equinor-hywind-scotland-windfarm, a. Accessed: 2020-07-26.

ORE Catapult. Platform for Operational Data: Levenmouth Demonstration Turbine. https://pod.ore.catapult.org.uk/source/levenmouth-demonstration-turbine, b. Accessed: 2020-07-26.

ORE Catapult. System performance, availability and reliability trend analysis. *ORE Catapult*, 1(1):1–26, 2019.

Shiyan Ou, Viktor Pekar, Constantin Orasan, Christian Spurk, and Matteo Negri. Development and alignment of a domain-specific ontology for question answering. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/561_paper.pdf.

Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Big data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, 30(4):431–448, 2018. ISSN 1319-1578. doi: https://doi.org/10.1016/j.jksuci.2017.06.001. URL https://www.sciencedirect.com/science/article/pii/S1319157817300034.

Ruilin Pan, Tingsheng Yang, Jianhua Cao, Ke Lu, and Zhanchao Zhang. Missing data imputation by k nearest neighbours based on grey relational structure and mutual information. *Applied Intelligence*, 43(3):614–632, October 2015. ISSN 0924-669X. doi: 10.1007/s10489-015-0666-x. URL https://doi.org/10.1007/s10489-015-0666-x.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/tkde.2009.191.

Yubin Pan, Rongjing Hong, Jie Chen, Jianshe Feng, and Weiwei Wu. Performance degradation assessment of wind turbine gearbox based on maximum mean discrepancy and multi-sensor transfer learning. *Structural Health Monitoring*, page 1475921720919073, 2021. doi: 10.1177/1475921720919073. URL https://doi.org/10.1177/1475921720919073.

Yanhua Pang, Qun He, Guoqian Jiang, and Ping Xie. Spatio-temporal fusion neural network for multi-class fault diagnosis of wind turbines based on scada data. *Renewable Energy*, 2020a. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2020.06.154. URL http://www.sciencedirect.com/science/article/pii/S0960148120310764.

Yanhua Pang, Qun He, Guoqian Jiang, and Ping Xie. Spatio-temporal fusion neural network for multi-class fault diagnosis of wind turbines based on scada data. *Renewable Energy*, 161:510–524, 2020b. doi: 10.1016/j.renene.2020.06.154.

N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 399–414, 2018. doi: 10.1109/EuroSP.2018.00035.

Mario Andrés Paredes-Valverde, José Ángel Noguera-Arnaldos, Cristian Aarón Rodríguez-Enríquez, Rafael Valencia-García, and Giner Alor-Hernández. A natural language interface to ontology-based knowledge bases. In Sigeru Omatu, Qutaibah M. Malluhi, Sara Rodríguez Gonzalez, Grzegorz Bocewicz, Edgardo Bucciarelli, Gianfranco Giulioni, and Farkhund Iqba, editors, *Distributed Computing and Artificial Intelligence, 12th International Conference*, pages 3–10, Cham, 2015. Springer International Publishing. ISBN 978-3-319-19638-1.

J. Park, K. Smarsly, K. H. Law, , and D. Hartmann. Multivariate analysis and prediction of wind turbine response to varying wind field characteristics based on machine learning. In *ASCE International Workshop on Computing in Civil Engineering*, pages 113–120, Los Angeles, California, June 2013.

Seonyeong Park, Soonchoul Kwon, Byungsoo Kim, Sangdo Han, Hyosup Shim, and Gary Geunbae Lee. Question answering system using multiple information source and open type answer merge. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 111–115, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-3023. URL https://www.aclweb.org/anthology/N15-3023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Mile Pavlić, Zdravko Dovedan Han, and Alen Jakupović. Question answering with a conceptual framework for knowledge-based system development "node of knowledge". *Expert Syst. Appl.*, 42(12):5264–5286, July 2015. ISSN 0957-4174. doi: 10.1016/j.eswa.2015.02.024. URL https://doi.org/10.1016/j.eswa.2015.02.024.

Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM*, 62(3):54–60, February 2019. ISSN 0001-0782. doi: 10.1145/3241036. URL https://doi.org/10.1145/3241036.

Domagoj Peharda, Igor Ivankovic, and Nikola Jaman. Using data from scada for centralized transformer monitoring applications. *Procedia Engineering*, 202:65–75, 12 2017. doi: 10.1016/j.proeng.2017.09.695.

Joanne Peng, Kuk Lee, and Gary Ingersoll. An introduction to logistic regression analysis and reporting. *Journal of Educational Research - J EDUC RES*, 96:3–14, 09 2002. doi: 10.1080/00220670209598786.

Truong V. Phan and P. Do. Bert+vnkg: Using deep learning and knowledge graph to improve vietnamese question answering system. *International Journal of Advanced Computer Science and Applications*, 11, 2020.

PHM Society. 2011 PHM society conference data challenge. https://www.phmsociety.org/competition/phm/11, 2011. Accessed: 2021-01-09.

Joan Plepi, Endri Kacupaj, Kuldeep Singh, Harsh Thakkar, and Jens Lehmann. Context transformer with stacked pointer networks for conversational question answering over knowledge graphs, 2021.

I. Portugal, P. Alencar, and D. Cowan. A preliminary survey on domain-specific languages for machine learning in big data. In *2016 IEEE International Conference on Software Science, Technology and Engineering (SWSTE)*, pages 108–110, 2016. doi: 10.1109/SWSTE.2016.23.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2923–2934, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1275.

Sonja Pravilovic, Annalisa Appice, Antonietta Lanza, and Donato Malerba. Wind power forecasting using time series cluster analysis. In *Discovery Science*, pages 276–287. Springer International Publishing, 2014. doi: 10.1007/978-3-319-11812-3_24. URL https://doi.org/10.1007%2F978-3-319-11812-3_24.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. Exploring diverse expressions for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3173–3182, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1313. URL https://www.aclweb.org/anthology/D19-1313.

Peng Qian, Xiandong Ma, and Yifei Wang. Condition monitoring of wind turbines based on extreme learning machine. In *Proceedings of the 21st International Conference on Automation and Computing*, pages 1–6, Glasgow, UK, September 2015.

Wei Qiao and Dingguo Lu. A survey on wind turbine condition monitoring and fault diagnosis part ii: Signals and signal processing methods. *IEEE Transactions on Industrial Electronics*, 62(10):6546–6557, 2015. doi: 10.1109/tie.2015.2422394.

Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016, 05 2016. doi: 10.1186/s13634-016-0355-x.

Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9:909, 03 2019. doi: 10.3390/app9050909.

Yingning Qiu, Yanhui Feng, P.J. Tavner, Paul Richardson, Gábor Erdos, and Bindi Chen. Wind turbine scada alarm analysis for improving reliability. *Wind Energy*, 15:951–966, 11 2012. doi: 10.1002/we.513.

Aqsa Saeed Qureshi, Asifullah Khan, Aneela Zameer, and Anila Usman. Wind power prediction using deep neural network based meta regression and transfer learning. *Applied Soft Computing*, 58:742–755, 2017. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.05.031. URL http://www.sciencedirect.com/science/article/pii/S1568494617302946.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

S. Reddy, Kishore Prathipati, and Young Lho. Transient stability improvement of a system connected with wind energy generators. *International Journal of Emerging Electric Power Systems*, 18, 01 2017. doi: 10.1515/ijeeps-2017-0063.

Siva Reddy, Mirella Lapata, and Mark Steedman. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2: 377–392, 2014. doi: 10.1162/tacl_a_00190. URL https://www.aclweb.org/anthology/Q14-1030.

Abdolrahim Rezaeiha, Ivo Kalkman, and Bert Blocken. Effect of pitch angle on power performance and aerodynamics of a vertical axis wind turbine. *Applied Energy*, 197: 132–150, 2017. ISSN 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2017.03.128. URL http://www.sciencedirect.com/science/article/pii/S0306261917303744.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Matīss Rikters. Impact of corpora quality on neural machine translation. *Frontiers in Artificial Intelligence and Applications: Human Language Technologies – The Baltic Perspective*, 307:126–133, October 2018. doi: 10.3233/978-1-61499-912-6-126.

Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. Exploiting paraphrases in a question answering system. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 25–32, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1118984.1118988. URL https://www.aclweb.org/anthology/W03-1604.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

Christine Röckmann, Sander Lagerveld, and John Stavenuiter. *Operation and Maintenance Costs of Offshore Wind Farms and Potential Multi-use Platforms in the Dutch North Sea*, chapter 1, pages 97–113. Springer International Publishing, Cham, 2017. ISBN 978-3-319-51159-7. doi: 10.1007/978-3-319-51159-7_4. URL https://doi.org/10.1007/978-3-319-51159-7_4.

Y. Roh, G. Heo, and S. E. Whang. A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2019. doi: 10.1109/TKDE.2019.2946162.

M. K. Rohil, R. K. Rohil, D. Rohil, and A. Runthala. Natural language interfaces to domain specific knowledge bases: An illustration for querying elements of the periodic table. In *2018 IEEE 17th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 517–523, 2018. doi: 10.1109/ICCI-CC.2018.8482023.

Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-5004. URL https://www.aclweb.org/anthology/N19-5004.

Aitor Saenz-Aguirre, Ekaitz Zulueta, Unai Fernandez-Gamiz, Alain Ulazia, and Daniel Teso-Fz-Betono. Performance enhancement of the artificial neural network–based reinforcement learning for wind turbine yaw control. *Wind Energy*, 23(3):676–690, 2020. doi: 10.1002/we.2451. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/we.2451.

Aurora Saibene, Michela Assale, and Marta Giltri. Expert systems: Definitions, advantages and issues in medical field applications. *Expert Systems with Applications*, 177:114900, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.114900. URL https://www.sciencedirect.com/science/article/pii/S0957417421003419.

L. Saidi, E. Bechhoefer, J. Ben Ali, and M. Benbouzid. Wind turbine high-speed shaft bearing degradation analysis for run-to-failure testing using spectral kurtosis. In *2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 267–272, 2015.

Esther Sainz, Andrés Llombart, and Josechu Guerrero. Robust filtering for the characterization of wind turbines: Improving its operation and maintenance. *Energy Conversion and Management*, pages 2136–2147, 09 2009. doi: 10.1016/j.enconman.2009.04.036.

Pedro Santos, Luisa Villa Montoya, Anibal Renones, Andres Bustillo, and Jesus Maudes. An svm-based solution for fault detection in wind turbines. *Sensors (Basel, Switzerland)*, 15:5627–48, 03 2015. doi: 10.3390/s150305627.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12):54–63, November 2020. ISSN 0001-0782. doi: 10.1145/3381831. URL https://doi.org/10.1145/3381831.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1066.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019, 2019.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. Natural language understanding with the quora question pairs dataset, 2019.

Carla Shearer. Machine learning in the wind industry, December 2019. URL http://machinelearning.shoutwiki.com/wiki/Machine_Learning_in_the_Wind_Industry.

X. Shen, X. Fu, and C. Zhou. A combined algorithm for cleaning abnormal data of wind turbine power curve based on change point grouping algorithm and quartile algorithm. *IEEE Transactions on Sustainable Energy*, 10(1):46–54, 2019. doi: 10.1109/TSTE.2018.2822682.

Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018. URL http://arxiv.org/abs/1808.03314.

Kumar Shivam, Jong-Chyuan Tzou, and Shang-Chen Wu. Multi-step short-term wind speed prediction using a residual dilated causal convolutional network with nonlinear attention. *Energies*, 13(7), 2020. ISSN 1996-1073. doi: 10.3390/en13071772. URL https://www.mdpi.com/1996-1073/13/7/1772.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020.

Yulin Si, Liyang Qian, Baijin Mao, and Dahai Zhang. A data-driven approach for fault detection of offshore wind turbines using random forests. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 3149–3154, Beijing, China, October 2017.

Siemens. Wpd offshore goes for service vessels, 2018. URL https://www.siemensgamesa.com/explore/customer-references/smooth-wind-rough-seas.

Kuldeep Singh, Ioanna Lytra, Arun Sethupat Radhakrishna, Saeedeh Shekarpour, Maria-Esther Vidal, and Jens Lehmann. No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph. *Journal of Web Semantics*, 65:100594, 2020. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2020.100594. URL https://www.sciencedirect.com/science/article/pii/S1570826820300342.

Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017. ISSN 0148-2963. doi: https://doi.org/10.1016/j.jbusres.2016.08.001. URL https://www.sciencedirect.com/science/article/pii/S014829631630488X.

Sharon Small, Ting Liu, Nobuyuki Shimizu, and Tomek Strzalkowski. HITIQA: An interactive question answering system: A preliminary report. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 46–53, Sapporo, Japan, July 2003. Association for Computational Linguistics. doi: 10.3115/1119312.1119318. URL https://www.aclweb.org/anthology/W03-1206.

Vaishali Sohoni, S. Gupta, and Rajesh Nema. A critical review on wind turbine power curve modelling techniques and their applications in wind based energy systems. *Journal of Energy*, 2016:1–18, 01 2016. doi: 10.1155/2016/8519785.

Soodabeh Soleymani, Sirus Mohammadi, Hamid-Reza Rezayi, and Rohalla Moghimai. A new hybrid method to forecast wind turbine output power in power systems. *Journal of Intelligent & Fuzzy Systems*, 28(4):1503–1508, 2015. ISSN 1064-1246. doi: 10.3233/IFS-141433. URL http://doi.org/10.3233/IFS-141433.

Ana Sovic and Damir Sersic. Signal decomposition methods for reducing drawbacks of the dwt. *Engineering Review*, 32(2):70–77, 01 2012.

Pranay Kumar Venkata Sowdaboina, Sutanu Chakraborti, and Somayajulu Sripada. Learning to summarize time series data. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 515–528, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-642-54906-9.

Rohini Srihari and Wei Li. A question answering system supported by information extraction. In *Sixth Applied Natural Language Processing Conference*, pages 166–172, Seattle, Washington, USA, April 2000. Association for Computational Linguistics. doi: 10.3115/974147.974170. URL https://www.aclweb.org/anthology/A00-1023.

Somayajulu G. Sripada, Ehud Reiter, Ian Davy, and Kristian Nilssen. Lessons from deploying nlg technology for marine weather forecast text generation. In *Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, pages 760–764, 2004. ISBN 978-1-58603-452-8.

D. Steinkraus, I. Buck, and P. Y. Simard. Using gpus for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1115–1120 Vol. 2, 2005. doi: 10.1109/ICDAR.2005.251.

Adrian Stetco, Fateme Dinmohammadi, Xingyu Zhao, Valentin Robu, David Flynn, Mike Barnes, John Keane, and Goran Nenadic. Machine learning methods for wind turbine condition monitoring: A review. *Renewable Energy*, 133:620–635, 2019. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2018.10.047. URL http://www.sciencedirect.com/science/article/pii/S096014811831231X.

M. Sundermeyer, I. Oparin, J. . Gauvain, B. Freiberg, R. Schlüter, and H. Ney. Comparison of feedforward and recurrent neural network language models. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8430–8434, 2013. doi: 10.1109/ICASSP.2013.6639310.

Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1017–1024. ACM, June 2011.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969173.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Johan Suykens, Joos Vandewalle, and Bart De Moor. *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Springer, 01 1996. ISBN 978-0-7923-9678-9. doi: 10.1007/978-1-4757-2493-6.

José A. Sáez, Bartosz Krawczyk, and Michał Woźniak. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178, 2016. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2016.03.012. URL http://www.sciencedirect.com/science/article/pii/S0031320316001072.

Chee Fai Tan, L.S. Wahidin, Siti Nurhaida Khalil, Noreffendy Tamaldin, Jun Hu, and Matthias Rauterberg. The application of expert system: A review of research and applications. *ARPN Journal of Engineering and Applied Sciences*, 11:2448–2453, 01 2016.

Hong Hui Tan and King Hann Lim. Vanishing gradient mitigation with deep learning neural network optimization. *2019 7th International Conference on Smart Computing Communications (ICSCC)*, 2019. doi: 10.1109/icscc.2019.8843652.

Y. Tang, J. Xu, K. Matsumoto, and C. Ono. Sequence-to-sequence model with attention for time series classification. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 503–510, Los Alamitos, CA, USA, December 2016. IEEE Computer Society. doi: 10.1109/ICDMW.2016.0078. URL https://doi.ieeecomputersociety.org/10.1109/ICDMW.2016.0078.

Barnabas K. Tannahill and Mo Jamshidi. Big data analytic paradigms -from pca to deep learning. In *The Intersection of Robust Intelligence and Trust in Autonomous Systems: Papers from the AAAI Spring Symposium*, pages 84–90, Stanford University, California, USA, March 2014.

Liang Tao, Qian Siqi, Yingjuan Zhang, and Huan Shi. Abnormal detection of wind turbine based on scada data mining. *Mathematical Problems in Engineering*, 2019:1–10, 08 2019. doi: 10.1155/2019/5976843.

Jannis Tautz-Weinert and Simon J. Watson. Using scada data for wind turbine condition monitoring – a review. *IET Renewable Power Generation*, 11:382–394, 2017.

Aditya Thiruvengadam. Transformer architecture: Attention is all you need, March 2019. URL https://medium.com/@adityathiruvengadam/transformer-architecture-attention-is-all-you-need-aeccd9f50d09.

Antu Annam Thomas and M. Wilscy. A comparative study of feedforward neural network and simplified fuzzy artmap in the context of face recognition. In Natarajan Meghanathan, Brajesh Kumar Kaushik, and Dhinaharan Nagamalai, editors, *Advances in Networks and Communications*, pages 277–289, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-17878-8.

Xue Tian, Wu Dinghai, and Wang Huaiguang. Research on application of transfer learning in equipment fault diagnosis. *Journal of Physics: Conference Series*, 1986(1):012099, aug 2021. doi: 10.1088/1742-6596/1986/1/012099. URL https://doi.org/10.1088/1742-6596/1986/1/012099.

N. Tomin, V. Kurbatsky, and H. Guliyev. Intelligent control of a wind turbine based on reinforcement learning. In *2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA)*, pages 1–6, 2019.

Peihao Tong, Qifan Zhang, and Junjie Yao. Leveraging domain context for question answering over knowledge graph. *Data Science and Engineering*, 4, 12 2019. doi: 10.1007/s41019-019-00109-w.

Eva Topham and David McMillan. Sustainable decommissioning of an offshore wind farm. *Renewable Energy*, 102:470–480, 2017. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2016.10.066. URL http://www.sciencedirect.com/science/article/pii/S0960148116309430.

F. Trivellato, L. Battisti, and G. Miori. The ideal power curve of small wind turbines from field data. *Journal of Wind Engineering and Industrial Aerodynamics*, 107-108:263–273, 2012a. ISSN 0167-6105. doi: https://doi.org/10.1016/j.jweia.2012.04.026. URL http://www.sciencedirect.com/science/article/pii/S0167610512001286.

F. Trivellato, L. Battisti, and G. Miori. The ideal power curve of small wind turbines from field data. *Journal of Wind Engineering and Industrial Aerodynamics*, 107-108:263–273, 2012b. doi: 10.1016/j.jweia.2012.04.026.

Brian Tubay and Marta R. Costa-Jussà. Neural machine translation with the transformer and multi-source romance languages for the biomedical WMT 2018 task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 667–670, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6449.

Alan Turnbull, James Carroll, and Alasdair McDonald. Combining SCADA and vibration data into a single anomaly detection model to predict wind turbine component failure. *Wind Energy*, oct 2020. doi: 10.1002/we.2567. URL https://doi.org/10.1002%2Fwe.2567.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

G. A. M. van Kuik et al. Long-term research challenges in wind energy – a research agenda by the european academy of wind energy. *WindEnergy Science*, 1:1–39, 2016. doi: 10.5194/wes-1-1-2016.

Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of expected sarsa. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184, 05 2009. doi: 10.1109/ADPRL.2009.4927542.

Vincent Vanhoucke, Andrew Senior, and Mark Z. Mao. Improving the speed of neural networks on cpus. In *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, 2011.

Nadya Vasilyeva, Thomas Blanchard, and Tania Lombrozo. Stable causal relationships are better causal relationships. *Cognitive Science*, 42(4):1265–1296, 2018. doi: 10.1111/cogs.12605. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Mani Vegupatti, Matthias Nickles, and Bharathi Raja Chakravarthi. Simple question answering over a domain-specific knowledge graph using BERT by transfer learning. In Luca Longo, Lucas Rizzo, Elizabeth Hunter, and Arjun Pakrashi, editors, *Proceedings of The 28th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Republic of Ireland, December 7-8, 2020*, volume 2771 of *CEUR Workshop Proceedings*, pages 289–300. CEUR-WS.org, 2020a. URL http://ceur-ws.org/Vol-2771/AICS2020_paper_42.pdf.

Mani Vegupatti, Matthias Nickles, and Bharathi Raja Chakravarthi. Simple question answering over a domain-specific knowledge graph using bert by transfer learning. In *AICS*, 2020b.

Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019. URL https://arxiv.org/abs/1906.05714.

Ellen M. Voorhees. The trec question answering track. *Natural Language Engineering*, 7(4): 361–378, 2001. doi: 10.1017/S1351324901002789.

Pavlos Vougiouklis, Hady Elsahar, Lucie-Aimée Kaffee, Christoph Gravier, Frederique Laforest, Jonathon Hare, and Elena. Neural wikipedian: Generating textual summaries from knowledge base triples. *Journal of Web Semantics*, in press, 2018.

Pavlos Vougiouklis, Eddy Maddalena, Jonathon Hare, and Elena Simperl. Point at the triple: Generation of text summaries from knowledge base triples. *Journal of Artificial Intelligence Research*, 69:1–31, sep 2020. doi: 10.1613/jair.1.11694. URL https://doi.org/10.1613%2Fjair.1.11694.

Shuting Wan, Lifeng Cheng, and Xiaoling Sheng. Effects of yaw error on wind turbine running characteristics based on the equivalent wind speed model. *Energies*, 8(7):6286–6301, 2015a. doi: 10.3390/en8076286.

Shuting Wan, Lifeng Cheng, and Xiaoling Sheng. Effects of yaw error on wind turbine running characteristics based on the equivalent wind speed model. *Energies*, 8:6286–6301, 07 2015b. doi: 10.3390/en8076286.

X. Wang, Z. Li, J. Zhang, H. Liu, C. Qiu, and X. Cai. An lstm-attention wind power prediction method considering multiple factors. In *8th Renewable Power Generation Conference (RPG 2019)*, pages 1–7, 2019.

Yixin Wang and David M. Blei. The Blessings of Multiple Causes. *Journal of the American Statistical Association*, Volume 0:1–71, 2019. doi: 10.1080/01621459.2019.1686987. URL https://doi.org/10.1080/01621459.2019.1686987.

Yu Emma Wang, Gu-Yeon Wei, and David Brooks. Benchmarking tpu, gpu, and cpu platforms for deep learning, 2019.

Christopher Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8: 279–292, 05 1992. doi: 10.1007/BF00992698.

Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016. doi: 10.1186/s40537-016-0043-6. URL https://doi.org/10.1186/s40537-016-0043-6.

Feng Wen, Xingqiao Wang, and Xiaowei Xu. Hierarchical sarsa learning based route guidance algorithm. *Journal of Advanced Transportation*, 2019:1–12, 06 2019. doi: 10.1155/2019/1019078.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Joyce Jiyoung Whang, Inderjit S. Dhillon, and David F. Gleich. Non-exhaustive, overlapping k-means. *Proceedings of the 2015 SIAM International Conference on Data Mining*, 2015. doi: 10.1137/1.9781611974010.105.

Sam Witteveen and Martin Andrews. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5623. URL https://www.aclweb.org/anthology/D19-5623.

JiaHui Wu, ZhenGuo Shao, and ShaoHua Yang. A combined algorithm for data cleaning of wind power scatter diagram considering actual engineering characteristics. *Journal of Physics: Conference Series*, 1639:012044, October 2020a. doi: 10.1088/1742-6596/1639/1/012044. URL https://doi.org/10.1088/1742-6596/1639/1/012044.

Junjie Wu. The uniform effect of k-means clustering. *Advances in K-means Clustering Springer Theses*, pages 17–35, 2012.

Yuan-Kang Wu, Ching-Yin Lee, Chao-Rong Chen, Kun-Wei Hsu, and Huang-Tien Tseng. Optimization of the wind turbine layout and transmission system planning for a large-scale offshore wind farm by ai technology. *IEEE Transactions on Industry Applications*, 50: 2071–2080, 2014.

Zidong Wu, Xiaoli Wang, and Baochen Jiang. Fault diagnosis for wind turbines based on relieff and extreme gradient boosting. *Applied Sciences*, 10(9), 2020b. ISSN 2076-3417. doi: 10.3390/app10093258. URL https://www.mdpi.com/2076-3417/10/9/3258.

Cheng Xiao, Zuojun Liu, Tieling Zhang, and Lei Zhang. On fault prediction for wind turbine pitch system using radar chart and support vector machine approach. *Energies*, 12 (14), 2019. ISSN 1996-1073. doi: 10.3390/en12142693. URL https://www.mdpi.com/1996-1073/12/14/2693.

X. Xie, W. Song, L. Liu, C. Du, and H. Wang. Research and implementation of automatic question answering system based on ontology. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pages 1366–1370, 2015. doi: 10.1109/CCDC.2015.7162131.

Y. Xie and T. Zhang. Imbalanced learning for fault diagnosis problem of rotating machinery based on generative adversarial networks. In *2018 37th Chinese Control Conference (CCC)*, pages 6017–6022, 2018. doi: 10.23919/ChiCC.2018.8483334.

Yan Xu, Yanming Sun, Xiaolong Liu, and Yonghua Zheng. A digital-twin-assisted fault diagnosis using deep transfer learning. *IEEE Access*, 7:19990–19999, 2019. doi: 10.1109/ACCESS.2018.2890566.

J. Yan, C. Liu, and M. Govindarasu. Cyber intrusion of wind farm scada system and its impact analysis. In *2011 IEEE/PES Power Systems Conference and Exposition*, pages 1–6, 2011. doi: 10.1109/PSCE.2011.5772593.

Chunzhen Yang, Jingquan Liu, Yuyun Zeng, and Guangyao Xie. Real-time condition monitoring and fault detection of components based on machine-learning reconstruction model. *Renewable Energy*, 133:433–441, 2019.

Lei Yang, Miao He, Junshan Zhang, and Vijay Vittal. Support-vector-machine-enhanced markov model for short-term wind power forecast. *IEEE Transactions on Sustainable Energy*, 6(3):791–799, July 2015. doi: 10.1109/tste.2015.2406814. URL https://doi.org/10.1109%2Ftste.2015.2406814.

Qingyu Yang and Dou An. Emd and wavelet transform based fault diagnosis for wind turbine gear box. *Advances in Mechanical Engineering*, 5:212836, 2013. doi: 10.1155/2013/212836. URL https://doi.org/10.1155/2013/212836.

Wenxian Yang and Jiesheng Jiang. Wind turbine condition monitoring and reliability analysis by scada information. *2011 Second International Conference on Mechanic Automation and Control Engineering*, 2011. doi: 10.1109/mace.2011.5987329.

Wenxian Yang, Richard Court, Peter J. Tavner, and Christopher J. Crabtree. Bivariate empirical mode decomposition and its contribution to wind turbine condition monitoring. *Journal of Sound Vibration*, 330(15):3766–3782, July 2011. doi: 10.1016/j.jsv.2011.02.027.

Wenxian Yang, Peter J. Tavner, Christopher J. Crabtree, Y. Feng, and Y. Qiu. Wind turbine condition monitoring: technical and commercial challenges. *Wind Energy*, 17:673–693, 2014.

Yanping Guo, Wenjun Yan, and Zhejing Bao. Gear fault diagnosis of wind turbine based on discrete wavelet transform. In *2010 8th World Congress on Intelligent Control and Automation*, pages 5804–5808, 2010.

Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with Freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956–966, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1090. URL https://www.aclweb.org/anthology/P14-1090.

H. Yi, Q. Jiang, X. Yan, and B. Wang. Imbalanced classification based on minority clustering smote with wind turbine fault detection application. *IEEE Transactions on Industrial Informatics*, pages 1–1, 2020. doi: 10.1109/TII.2020.3046566.

Pengcheng Yin and Graham Neubig. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1041. URL https://www.aclweb.org/anthology/P17-1041.

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1164.

Qian Yu, Wai Lam, and Zihao Wang. Responding E-commerce product questions via exploiting QA collections and reviews. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2192–2203, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1186.

Yajie Yu, Hui Cao, Shang Liu, Shuo Yang, and Ruixian Bai. Image-based damage recognition of wind turbine blades. In *2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 161–166, Hefei and Tai'an, China, August 2017.

Tongke Yuan, Zhifeng Sun, and Shihao Ma. Gearbox Fault Prediction of Wind Turbines Based on a Stacking Model and Change-Point Detection. *Energies*, 12(22):1–20, November 2019. URL https://ideas.repec.org/a/gam/jeners/v12y2019i22p4224-d283963.html.

A. Zabetian-Hosseini, A. Mehrizi-Sani, and C. Liu. Cyberattack to cyber-physical model of wind farm scada. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 4929–4934, 2018. doi: 10.1109/IECON.2018.8591200.

Hamid Zafar, Mohnish Dubey, Jens Lehmann, and Elena Demidova. Iqa: Interactive query construction in semantic question answering systems. *Journal of Web Semantics*, 64: 100586, 2020. ISSN 1570-8268. doi: https://doi.org/10.1016/j.websem.2020.100586. URL https://www.sciencedirect.com/science/article/pii/S1570826820300305.

A. Zaher, S.D.J. McArthur, and D.G. Infield. Online wind turbine fault detection through automated scada data analysis. *Wind Energy*, 12:574–593, 2009.

John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press, 1996. ISBN 026251091X.

C. Zhang, J. Bin, and Z. Liu. Wind turbine ice assessment through inductive transfer learning. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, 2018a.

D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si. A data-driven design for fault detection of wind turbines using random forests and xgboost. *IEEE Access*, 6: 21020–21031, 2018b.

Juan Zhang, Qi Yu, Fashan Zheng, Chao Long, Zuxun Lu, and Zhiguang Duan. Comparing keywords plus of wos and author keywords: A case study of patient adherence research. *Journal of the Association for Information Science and Technology*, 67, 02 2015. doi: 10.1002/asi.23437.

Peter Zhang, Eddy Patuwo, and Michael Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14:35–62, 03 1998. doi: 10.1016/S0169-2070(97)00044-7.

Xiangzhou Zhang, Yong Hu, Kang Xie, Shouyang Wang, E.W.T. Ngai, and Mei Liu. A causal feature selection algorithm for stock prediction modeling. *Neurocomputing*, 142: 48–59, 10 2014. doi: 10.1016/j.neucom.2014.01.057.

Zhang Jinhua et al. Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model. *Applied Energy*, 241(C):229–244, 2019. doi: 10.1016/j.apenergy.2019.0. URL https://ideas.repec.org/a/eee/appene/v241y2019icp229-244.html.

P. Zhao, J. Xia, Y. Dai, and J. He. Wind speed prediction using support vector regression. In *IEEE Conference on Industrial Electronics and Applications*, pages 882–886, Taichung, Taiwan, June 2010. IEEE.

Yingying Zhao, Dongsheng Li, Ao Dong, Dahai Kang, Qin Lv, and Li Shang. Fault prediction and diagnosis of wind turbine generators using scada data. *Energies*, 10(8):1210, 2017. doi: 10.3390/en10081210.

Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. doi: 10.1109/iccv.2017.557.

Zeng-Wei Zheng, Yuan-Yi Chen, Xiao-Wei Zhou, Mei-Mei Huo, Bo Zhao, and Min-Yi Guo. Short-term wind power forecasting using empirical mode decomposition and rbfnn. *International Journal of Smart Grid and Clean Energy*, 2:192–199, 01 2013. doi: 10.12720/sgce.2.2.192-199.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, 2018.

Caichao Zhu and Yao Li. Reliability analysis of wind turbines. In *Stability Control and Reliable Performance of Wind Turbines*. October 2018. doi: 10.5772/intechopen.74859.

Q. Zhu, H. Li, Z. Wang, J. Chen, and B. Wang. Short-term wind power forecasting based on lstm. *Dianwang Jishu/Power System Technology*, 41:3797–3802, 12 2017. doi: 10.13335/j.1000-3673.pst.2017.1657.

Radoslaw Zimroz, Walter Bartelmus, Tomasz Barszcz, and Jacek Urbanek. Statistical data processing for wind turbine generator bearing diagnostics. In Tahar Fakhfakh, Walter Bartelmus, Fakher Chaari, Radoslaw Zimroz, and Mohamed Haddar, editors, *Condition Monitoring of Machinery in Non-Stationary Operations*, pages 509–518, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-28768-8.

Ørsted. Ørsted Offshore Operational Data . https://orsted.com/en/our-business/offshore-wind/offshore-operational-data. Accessed: 2021-01-02.

Šarūnas Raudys. Evolution and generalization of a single neurone: I. single-layer perceptron as seven statistical classifiers. *Neural Networks*, 11(2):283–296, 1998. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(97)00135-4. URL http://www.sciencedirect.com/science/article/pii/S0893608097001354.

Appendix A

Details of SCADA features utilised from the LDT and ENGIE datasets

Variable Name in LDT Data (Source Domain)	Variable Name in ENGIE Data (Target Domain)	Description (As per ENGIE Data)
Pitch_Deg_Mean	Ba_avg	Pitch Angle Mean Value (deg)
Pitch_Deg_Min	Ba_min	Pitch Angle Minimum Value (deg)
Pitch_Deg_Max	Ba_max	Pitch Angle Maximum Value (deg)
Pitch_Deg_Stdev	Ba_std	Pitch Angle Standard Deviation (deg)
SubPtchPrivHubtemp_Mean	Rt_avg	Hub Temperature Mean Value (deg celsius)
SubPtchPrivHubtemp_Min	Rt_min	Hub Temperature Minimum Value (deg celsius)
SubPtchPrivHubtemp_Max	Rt_max	Hub Temperature Maximum Value (deg celsius)
SubPtchPrivHubtemp_Stdev	Rt_std	Hub Temperature Standard Deviation (deg celsius)
SubPcsPrivRefGenSpeedInching_Mean	DCs_avg	Generator Converter Speed Mean Value (rpm)
SubPcsPrivRefGenSpeedInching_Min	DCs_min	Generator Converter Speed Minimum Value (rpm)
SubPcsPrivRefGenSpeedInching_Max	DCs_max	Generator Converter Speed Maximum Value (rpm)

SubPcsPrivRefGenSpeedInching_Stdev	DCs_std	Generator Converter Speed Standard Deviation (rpm)
Power_kW_Mean	P_avg	Active Power Mean Value (kW)
Power_kW_Min	P_min	Active Power Minimum Value (kW)
Power_kW_Max	P_max	Active Power Maximum Value (kW)
Power_kW_Stdev	P_std	Active Power Standard Deviation (kW)
ReactivePower_kVar_Mean	Q_avg	Reactive Power Mean Value (kVAr)
ReactivePower_kVar_Min	Q_min	Reactive Power Minimum Value (kVAr)
ReactivePower_kVar_Max	Q_max	Reactive Power Maximum Value (kVAr)
ReactivePower_kVar_Stdev	Q_std	Reactive Power Standard Deviation (kVAr)
AuxConsumptionApparentPwr_Mean	S_avg	Apparent Power Mean Value (kVA)
AuxConsumptionApparentPwr_Min	S_min	Apparent Power Minimum Value (kVA)
AuxConsumptionApparentPwr_Max	S_max	Apparent Power Maximum Value (kVA)
AuxConsumptionApparentPwr_Stdev	S_std	Apparent Power Standard Deviation (kVA)
SubIprPrivPwrFactor_Mean	Cosphi_avg	Power Factor Mean Value
SubIprPrivPwrFactor_Min	Cosphi_min	Power Factor Minimum Value
SubIprPrivPwrFactor_Max	Cosphi_max	Power Factor Maximum Value
SubIprPrivPwrFactor_Std	Cosphi_std	Power Factor Standard Deviation
SubPcsPrivRefGenSpeedInching_Mean	Ds_avg	Generator Speed Mean Value (rpm)
GenBearingtemp1_Mean	Db1t_avg	Generator Bearing 1 Temperature Mean Value (deg celsius)
GenBearingtemp1_Min	Db1t_min	Generator Bearing 1 Temperature Minimum Value (deg celsius)

GenBearingtemp1_Max	Db1t_max	Generator Bearing 1 Temperature Maximum Value (deg celsius)
GenBearingtemp1_Stdev	Db1t_std	Generator Bearing 1 Temperature Standard Deviation (deg celsius)
GenBearingtemp2_Mean	Db2t_avg	Generator Bearing 2 Temperature Mean Value (deg celsius)
GenBearingtemp2_Min	Db2t_min	Generator Bearing 2 Temperature Minimum Value (deg celsius)
GenBearingtemp2_Max	Db2t_max	Generator Bearing 2 Temperature Maximum Value (deg celsius)
GenBearingtemp2_Stdev	Db2t_std	Generator Bearing 2 Temperature Standard Deviation (deg celsius)
GenStatortemp2_Mean	Dst_avg	Generator Stator Temperature Mean Value (deg celsius)
GenStatortemp2_Min	Dst_min	Generator Stator Temperature Minimum Value (deg celsius)
GenStatortemp2_Max	Dst_max	Generator Stator Temperature Maximum Value (deg celsius)
GenStatortemp2_Stdev	Dst_std	Generator Stator Temperature Standard Deviation (deg celsius)
GBoxOpShaftBearingTemp1_Mean	Gb1t_avg	Gearbox Bearing 1 Temperature Mean Value (deg celsius)
GBoxOpShaftBearingTemp1_Min	Gb1t_min	Gearbox Bearing 1 Temperature Minimum Value (deg celsius)
GBoxOpShaftBearingTemp1_Max	Gb1t_max	Gearbox Bearing 1 Temperature Maximum Value (deg celsius)

GBoxOpShaftBearingTemp1_Stdev	Gb1t_std	Gearbox Bearing 1 Temperature Standard Deviation (deg celsius)
GBoxOpShaftBearingTemp2_Mean	Gb2t_avg	Gearbox Bearing 2 Temperature Mean Value (deg celsius)
GBoxOpShaftBearingTemp2_Min	Gb2t_min	Gearbox Bearing 2 Temperature Minimum Value (deg celsius)
GBoxOpShaftBearingTemp2_Max	Gb2t_max	Gearbox Bearing 2 Temperature Maximum Value (deg celsius)
GBoxOpShaftBearingTemp2_Stdev	Gb2t_std	Gearbox Bearing 2 Temperature Standard Deviation (deg celsius)
GenHeWaterInlettemp_Mean	Git_avg	Generator Inlet Temperature Mean Value (deg celsius)
GenHeWaterInlettemp_Min	Git_min	Generator Inlet Temperature Minimum Value (deg celsius)
GenHeWaterInlettemp_Max	Git_max	Generator Inlet Temperature Maximum Value (deg celsius)
GenHeWaterInlettemp_Stdev	Git_std	Generator Inlet Temperature Standard Deviation (deg celsius)
GearBoxTemperature_DegC_Mean	Gost_avg	Gearbox Oil Sump Temperature Mean Value (deg celsius)
GearBoxTemperature_DegC_Min	Gost_min	Gearbox Oil Sump Temperature Minimum Value (deg celsius)
GearBoxTemperature_DegC_Max	Gost_max	Gearbox Oil Sump Temperature Maximum Value (deg celsius)
GearBoxTemperature_DegC_Stdev	Gost_std	Gearbox Oil Sump Temperature Standard Deviation (deg celsius)
NacelleOrientation_Deg_Mean	Ya_avg	Nacelle Angle Mean Value (deg)

NacelleOrientation_Deg_Min	Ya_min	Nacelle Angle Minimum Value (deg)
NacelleOrientation_Deg_Max	Ya_max	Nacelle Angle Maximum Value (deg)
NacelleOrientation_Deg_Stdev	Ya_std	Nacelle Angle Standard Deviation (deg)
NacInsidetemp_Mean	Yt_avg	Nacelle Temperature Mean Value (deg celsius)
NacInsidetemp_Min	Yt_min	Nacelle Temperature Minimum Value (deg celsius)
NacInsidetemp_Max	Yt_max	Nacelle Temperature Maximum Value (deg celsius)
NacInsidetemp_Stdev	Yt_std	Nacelle Temperature Standard Deviation (deg celsius)
WindSpeed1_Mean	Ws1_avg	Wind Speed 1 Mean Value (m/s)
WindSpeed1_Min	Ws1_min	Wind Speed 1 Minimum Value (m/s)
WindSpeed1_Max	Ws1_max	Wind Speed 1 Maximum Value (m/s)
WindSpeed1_Stdev	Ws1_std	Wind Speed 1 Standard Deviation (m/s)
WindSpeed2_Mean	Ws2_avg	Wind Speed 2 Mean Value (m/s)
WindSpeed2_Min	Ws2_min	Wind Speed 2 Minimum Value (m/s)
WindSpeed2_Max	Ws2_max	Wind Speed 2 Maximum Value (m/s)
WindSpeed2_Stdev	Ws2_std	Wind Speed 2 Standard Deviation (m/s)
WindSpeed_mps_Mean	Ws_avg	Average Wind Speed Mean Value (m/s)
WindSpeed_mps_Min	Ws_min	Average Wind Speed Minimum Value (m/s)
WindSpeed_mps_Max	Ws_max	Average Wind Speed Maximum Value (m/s)
WindSpeed_mps_Stdev	Ws_std	Average Wind Speed Standard Deviation (m/s)
WindVane_T_Mean	Wa_avg	Absolute Wind Direction Mean Value (deg)

WindVane_T_Min	Wa_min	Absolute Wind Direction Minimum Value (deg)
WindVane_T_Max	Wa_max	Absolute Wind Direction Maximum Value (deg)
WindVane_T_Stdev	Wa_std	Absolute Wind Direction Standard Deviation (deg)
WindVane_1_Mean	Wa_avg	Wind Vane 1 Direction Mean Value (deg)
WindVane_1_Min	Wa_min	Wind Vane 1 Direction Minimum Value (deg)
WindVane_1_Max	Wa_max	Wind Vane 1 Direction Maximum Value (deg)
WindVane_1_Stdev	Wa_std	Wind Vane 1 Direction Standard Deviation (deg)
temp_2_Mean	Ot_avg	Outdoor Temperature Mean Value (deg celsius)
temp_2_Min	Ot_min	Outdoor Temperature Minimum Value (deg celsius)
temp_2_Max	Ot_max	Outdoor Temperature Maximum Value (deg celsius)
SubIprPrivFreq_Mean	Nf_avg	Grid Frequency Mean Value (Hz)
SubIprPrivFreq_Min	Nf_min	Grid Frequency Minimum Value (Hz)
SubIprPrivFreq_Max	Nf_max	Grid Frequency Maximum Value (Hz)
SubIprPrivFreq_Stdev	Nf_std	Grid Frequency Standard Deviation (Hz)
SubPcsPrivGridVoltagePhaseTR_Mean	Nu_avg	Grid Voltage Mean Value (V)
SubPcsPrivGridVoltagePhaseTR_Min	Nu_min	Grid Voltage Minimum Value (V)
SubPcsPrivGridVoltagePhaseTR_Max	Nu_max	Grid Voltage Maximum Value (V)
SubPcsPrivGridVoltagePhaseTR_Stdev	Nu_std	Grid Voltage Standard Deviation (V)
RotorSpeed_rpm_Mean	Rs_avg	Rotor Speed Mean Value (rpm)
RotorSpeed_rpm_Min	Rs_min	Rotor Speed Minimum Value (rpm)
RotorSpeed_rpm_Max	Rs_max	Rotor Speed Maximum Value (rpm)

RotorSpeed_rpm_Stdev	Rs_std	Rotor Speed Standard Deviation (rpm)
MainBearingtemp1_Mean	Rbt_avg	Rotor Bearing Temperature Mean Value (deg celsius)
MainBearingtemp1_Min	Rbt_avg	Rotor Bearing Temperature Minimum Value (deg celsius)
MainBearingtemp1_Max	Rbt_avg	Rotor Bearing Temperature Maximum Value (deg celsius)
MainBearingtemp1_Stdev	Rbt_avg	Rotor Bearing Temperature Standard Deviation (deg celsius)

Appendix B

Details of Nodes in the Knowledge Graph

Node label	**Description**	**Properties**
System	Reference to the Study Turbine (LDT)	location, name, rated_power, type
Environment	Reference to the Study Turbine Environment	name
Blades	Wind Turbine Blades Sub-System	name, InspectionActivities, CorrectiveActivities, PreventiveActivities, visualinspection_image_url,image_url
ESC	Electric, Sensor & Control Sub-System	name
FCS	Foundation & Concrete Section Sub-System	name, image_url, CorrectiveActivities
HydraulicSystem	Hydraulic Sub-System	name, image_url, CorrectiveActivities
CommNetwork	Communications & Network Sub-System	name, image_url, CorrectiveActivities
Converter	Converter Sub-System	name, PredictiveActivities, PreventiveActivities
DriveTrain	Drive Train Sub-System	name, PreventiveActivities
Yaw	Yaw Sub-System	name, fno
ParkBrake	Park Brake Sub-System	name
YawBrake	Yaw Brake Sub-System	name, fno
MainShaft	Main Shaft Sub-System	name
PowerCabinet	Power Cabinet Sub-System	name
PitchSystem	Pitch Sub-System	name
Transformer	Transformer Sub-System	name, image_url, CorrectiveActivities

FunctionalGroup	Reference to all Functional Groups	name, contents
IPR	IPR Functional Group	name, fno
PitchInterfaceAlarm	Pitch System Interface Alarms Functional Group	name, fno
PitchEFCMon	Pitch System EFC Monitoring Sub-System	name, fno
PCS	Power Conditioning System Functional Group	name, fno
PPD	Partial Performance Degraded Functional Group	name, fno
Yaw	Yaw System Functional Group	name, fno
HydraulicSys	Hydraulic System Functional Group	name, fno
Gearbox	(i) Gearbox Functional Group (ii) Gearbox Sub-System	(i)name, fno (ii)name, image_url, fno, CorrectiveActivities
NoFault	No Fault Functional Group	name, fno
WindAlr	Wind Condition Alarms Functional Group	name, fno
GLLP	Ground Line & Lightning Protection Sub-System	name
MVTR	Moisture Vapour Transmission Rate Functional Group	name, fno
Test	Test Rig Functional Group	name, fno
Pitch	Pitch System Functional Group	name, fno
SCADA	Reference to all SCADA features	name
Alarm1 to Alarm26 (both inclusive)	26 different alarm types in the LDT	description, alarm_no
Feature	Reference to distinct SCADA features	name, description, unit, feature_no
MaintenanceAction	Reference to all maintenance actions	name, contents
FaultEvents	Reference to all fault events	name
FaultEvent1 to FaultEvent57 (both inclusive)	57 different types of fault events	details

Preventive	Reference to all preventive actions	name, Lineups, DriveTrainActivities, Cleaning, CheckWearSlackLineups, ImpParameters,OilChanges, SettingPressures, FunctionalChecks, Filters, Corrosion, LostScuffsOrGaps, FatLiqours, Sampling, Retightening, PlasticsOrDegradedGums
PrevAct1 to PrevAct233 (both inclusive)	233 distinct preventive maintenance actions	name, gen_periodicity, details, act, initial_periodicity, activities
Predictive	Reference to all predictive actions	name, contents
PredAct1 to PredAct11 (both inclusive)	11 distinct predictive maintenance actions	details, activities, image_url
Corrective	Reference to all corrective actions	name, contents
CorrAct1 to CorrAct57 (both inclusive)	57 distinct corrective maintenance actions	activities, image_url

Appendix C

Tags utilised in development of the domain-specific templates

Table C.1 Comprehensive details of various tags used in developing QC templates, along with example QC pairs

Tag	**Description**	**Example Question (Q)**	**Example Cypher Query (C)**
<fng-name>	14 different Functional group names pertaining to faults in different turbine sub-components e.g. *Gearbox*, *Pitch System EFC Monitoring* etc. For normal operation of the turbine, a *No fault* functional group is used.	What are important details for the *<fng-name>* named wind turbine functional group?	*MATCH(n{name:"<fng-name>"}) RETURN n*
<scadaname>	Labelled names of 102 SCADA features e.g. *Pitch_Deg_Mean*, *Power_kW_Mean* etc.	What are details of *<scadaname>* wind turbine SCADA feature.	*MATCH(n:Feature{name:"< scadaname>"}) RETURN n*

<*scada featureno*>	Indices of 102 SCADA features (from *0* to *101*)	What fault events are caused by wind turbine SCADA feature number <*scadafeatureno*>?	*MATCH(n:Feature{feature_no:< scadafeatureno>})-[:RELATESTO]-(p) RETURN p*
<*alarmno*>	Labels for all alarms available (from *901* to *926*)	What fault events are related to alarm <*alarmno*> in the wind turbine?	*MATCH(n{alarm_no:"<alarm no>"})-[:RELATESTO]-(p) RETURN p*
<*alarmdes*>	Detailed descriptions of all alarms in the turbine e.g. *Wind Speed Above Max Start*, *(DEMOTED) Gearbox Filter Manifold Pressure 1 Shutdown* etc.	What components are affected by <*alarmdes*> wind turbine alarm?	*MATCH(n{description:"<alarm des>"})-[:AFFECTS]-(p) RETURN p*
<*fevent-details*>	Detailed descriptions of 57 different types of fault events available in the KG e.g. *Temperature measurement module failure. Error in Temperature card (RTD)* etc.	What SCADA features contribute <*fevent-details*> fault event in the wind turbine?	*MATCH(n:FaultEvents)-[:TYPE]-(p{details:"<fevent-details>"})-[:RELATESTO]-(q:Feature) RETURN q*