Deep Learning for the Early Detection of Harmful Algal Blooms and Improving Water Quality Monitoring

A thesis submitted In Partial Fulfilment
for the degree of Doctor of Philosophy
in Computer Science

September 2022

By
Onatkut Dagtekin BSc, MSc
Department of Computer Science and Technology
University of Hull

For everything in the world; For civilization, for life, for success, the one true guide is science. Seeking a guide other than science is heedlessness, ignorance and heresy.

MUSTAFA KEMAL ATATÜRK

Contents

List of Tables

List of Figures

Abstract

Climate change will affect how water sources are managed and monitored. The frequency of algal blooms will increase with climate change as it presents favourable conditions for the reproduction of phytoplankton. During monitoring, possible sensory failures in monitoring systems result in partially filled data which may affect critical systems. Therefore, imputation becomes necessary to decrease error and increase data quality. This work investigates two issues in water quality data analysis: improving data quality and anomaly detection. It consists of three main topics: data imputation, early algal bloom detection using in-situ data and early algal bloom detection using multiple modalities.

The data imputation problem is addressed by experimenting with various methods with a water quality dataset that includes four locations around the North Sea and the Irish Sea with different characteristics and high miss rates, testing model generalisability. A novel neural network architecture with self-attention is proposed in which imputation is done in a single pass, reducing execution time. The self-attention components increase the interpretability of the imputation process at each stage of the network, providing knowledge to domain experts.

After data curation, algal activity is predicted using transformer networks, between 1 to 7 days ahead, and the importance of the input with regard to the output of the prediction model is explained using SHAP, aiming to explain model

behaviour to domain experts which is overlooked in previous approaches. The prediction model improves bloom detection performance by 5% on average and the explanation summarizes the complex structure of the model to input-output relationships.

Performance improvements on the initial unimodal bloom detection model are made by incorporating multiple modalities into the detection process which were only used for validation purposes previously. The problem of missing data is also tackled by using coordinated representations, replacing low quality in-situ data with satellite data and vice versa, instead of imputation which may result in biased results.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Acknowledgements

The past three years have been challenging in every way, and I could not have achieved what I have without the support of my family, friends and colleagues.

I would like to express my gratitude to my principal supervisor, Dr. Nina Dethlefs, for everything she has done for me these past three years. She is the perfect supervisor anyone could hope for. You have guided me not only academically and intellectually but also helped me grow as a person. I would also like to thank Dr. Chandrasekhar Kambhampati for chairing my PhD panels and providing invaluable feedback and advice about my research. I would like to thank Dr. Yongqiang Cheng for his feedback during my PhD panels and for encouraging me to participate in the PGR seminars to talk about my research. I would like to thank Dr. Naomi Greenwood and Dr. Veronique Creach for providing the data for this work and for their domain expertise that aided and guided my research. I would like to thank the University of Hull VIPER team for providing the infrastructure for my experiments and the University of Hull for funding this research. Thanks to all my colleagues in the Big Data Analytics research group.

I would like to thank my mother, Fethiye Aygul Dagtekin, my father, Omer Faruk Dagtekin, and my brother, Dr. Orhon Can Dagtekin, for their unwavering support not only during my PhD but throughout my life; without them, I would not be where I am. I would also like to extend my thanks to my friends back home and abroad.

Acronyms

ANN Artificial Neural Network

ARIMA Autoregressive Integrated Moving Average

AUC ROC Area Under the Receiver Operating Characteristic Curve

BR Bayesian Ridge

CEFAS Centre for Environment, Fisheries and Aquaculture Science

chl-a chlorophyll-a

CMS Copernicus Marine Service

CNN Convolutional Neural Network

CRF Conditional Random Fields

DBM Deep Boltzmann Machine

DTW Dynamic Time Warping

ELBO Evidence Lower Bound

GAIN Generative Adversarial Imputation Network

GAN Generative Adversarial Network

HAB Harmful Algal Bloom

IF Isolation Forest

k-NN k-Nearest Neighbours

LIME Local Interpretable Model-Agnostic Explanations

LSTM Long-Short Term Memory

MAE Mean Absolute Error

MAR Missing at Random

MCAR Missing Completely at Random

MERIS Medium Resolution Imaging Spectrometer

MICE Multiple Imputation by Chained Equations

MLP Multilayer Perceptron

MNAR Missing Not at Random

MODIS Moderate Resolution Imaging Spectroradiometer

MSE Mean Square Error

MSFD Marine Strategy Framework Directive

NDWI Normalized Difference Water Index

NLP Natural Language Processing

PAR Photosynthetically Active Radiation

RF Random Forest

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SAI Self-Attention Imputer

SARIMA Seasonal Autoregressive Integrated Moving Average

SeaWiFS Sea-viewing Wide Field-of-view Sensor

SHAP Shapley Additive Explanations

SOM Self-Organising Map

SVM Support Vector Machine

SVR Support Vector Regressor

TF-Conv Transformer-Convolution

VAE Variational Autoencoder

VIIRS Visible Infrared Imaging Radiometer Suite

XGBoost eXtreme Gradient Boosting

Chapter 1

Introduction

Water is vital in every aspect of life, from the ocean's depths to our bodies. It is heavily used in agriculture, electricity generation and other industrial applications (Pereira, 2017; Fthenakis and Kim, 2010; Flörke et al., 2013). Therefore, continuous monitoring of water quality is crucial to detect pollution, ensure that various natural cycles are not disrupted by anthropogenic activities and assess the effectiveness of beneficial management measures taken under defined protocols such as the EU Water Framework Directive (WFD) (Directive, 2000) and Marine Strategy Framework Directive (MSFD) (Directive, 2008). With increasing capability and low cost of sensors, constant monitoring has become widespread within research programmes providing high quality and in situ data. With the improving remote sensing technology, satellites such as Moderate Resolution Imaging Spectroradiometer (MODIS) (Justice et al., 1998) and Sea-viewing Wide Field-of-view Sensor (SeaWiFS) (McClain et al., 2004) and Sentinel provide detailed spatial and temporal data for water quality monitoring.

Harmful Algal Blooms (HABs) are outlier phenomena when algae multiply rapidly due to several factors, such as available light and nutrient flow Shumway et al. (2018). Algal blooms either naturally occur and pass away or begin due to extreme nutrient flow, i.e. eutrophication, and exacerbate. The extreme nutrient flow is caused by fertilisers or sewage from industrial zones or sewage pipes from residential areas, and it can affect freshwater sources Shumway et al. (2018).

Algae are key autotroph species which form the base of food webs in marine ecosystems through photosynthesis. With the increasing temperatures due to climate change, the frequency of algal blooms is expected to increase and will be seen in new regions (Wells et al., 2015). In addition to the ecological impacts, the occurrence of algal blooms has negative economic impacts. These include drinking water treatment costs, as these blooms produce deadly toxins and increase the cost of preserving biodiversity due to the disruption of the food chain (Dodds et al., 2009). Regions where these blooms are frequent see lower sales in sectors related to tourism, such that the by-product of HABs cause foul smells and irritation in the eyes and lower income from fisheries as the fish population is affected by the produced toxins (Bechard, 2020; Karlson et al., 2021).

Eventually, the scarcity of water will increase due to the effects of global warming. By 2050, it is expected that 3.1 billion people will experience water scarcity with additional economic and agricultural effects (Nations, 2021). To create a more sustained Earth, the United Nations defined seventeen goals, two directly related to water quality; Clean Water and Sanitation and Life Below Water, Goals 6 and 14, respectively (Assembly, 2015). This project closely relates to the goal of Life Below Water and could be extended to the goal of Clean Water for inland bodies of water.

Most of the study areas for this phenomenon are in East Asia (Lake Taihu, The South China Sea, The East China Sea and The Yellow Sea), The United States (The Great Lakes and The Gulf of Florida), The Baltic Sea and The Mediterranean Sea (Sebastiá-Frasquet et al., 2020). The majority of the study periods are also short, typically less than a year (Sebastiá-Frasquet et al., 2020).

The detection ranges are either short or have long intervals between detection periods. In this thesis, the aim is to predict HABs in the North and Irish Sea over a 10-year period, aiming to predict blooms 1 to 7 days before they occur using various modalities of data.

1.1 Explainability

Deep learning models consist of a large number of parameters which are not comprehensible. Therefore, models need to be summarised in an understandable format. In the context of HAB detection, the models must be suitable for domain experts to understand and utilise. Therefore, explainability models are required to move from a black-box approach. A section of this project addresses this issue by explaining the relationship between the model's input and output, contributing to the notion of explainability in the context of water quality and observing how model inputs affect the model output. As algal blooms cause public health issues, the general public should be able to obtain information about the future status of areas of interest. Although the monitoring sites covered by this project are not in proximity to populated areas, it can serve as a starting point for explainability in the water quality domain as the majority of the works regarding the detection of algal blooms and AI neglect the issue of explainability and focus on the performance of the model.

1.2 Research Questions

Deep learning and machine learning methods have previously been utilized for water quality data imputation and HAB detection (Sebastiá-Frasquet et al., 2020; Aissia et al., 2017). Although the performances of the models are satisfactory two key properties are often overlooked: generalisability and explainability of the models. The proposed solutions are often tested only on a single type of water body which limits usability as different bodies of water differ in susceptibility to change (Yang et al., 2020; Li et al., 2014; Lin et al., 2018; Song et al., 2015). Therefore, testing model performance on bodies of water with different properties is essential. Explainability is minimally explored in the domain of water quality data imputation and HAB detection which is essential for models to deployed to the field and be understandable by domain experts (Park et al., 2022). Different modalities such as satellite and in-situ could be used to detect HABs Current approaches use different modalities for validation purposes (Cannizzaro et al., 2009; Vannah and Chang, 2013). Each modality exposes different properties in the data to predict blooms. In-situ data tracks nutrients and phytoplankton activity and satellite data tracks nutrients and colour changes in the water. When analysed simultaneously, detection performance could be improved. Based on the identified problems, this thesis aims to answer the following questions:

1. What are the models for filling missing data that can be used to improve the quality of a water quality dataset and in what ways could the complexity of the process be visualised for interpretability?

2. What are the models that can aid the early detection of HABs, and how could the predictions of these models be interpreted?

3. In what ways data from multiple sources could be fused to improve the detection of HABs?

1.3 Objectives

The general objectives of this project are:

1. Exploration and comparison of data imputation for time series data: Commonly used methods for imputation in general (Section 2.2) and imputation for water quality data (Section 3.1)

2. Survey and evaluation of algal bloom detection methods and models: Variables used in detection (Section 2.1.4), algal bloom detection using in-situ data (Section 4.1) and satellite data (Section 5.1)

3. Investigation and assessment of model explainability for deep learning models (Section 4.1.2)

4. Analysis and review of multimodal learning in the context of algal bloom detection (Section 5.2 and Section 5.1)

1.4 Hypotheses

The following hypotheses are constructed in the scope of this thesis:

1. *It is hypothesised that missingness of water quality data is missing at random; therefore, observed variables could be used to recover missing ones.*

   This is realised by creating a novel deep learning model for imputation in partially observed water quality data.

2. *It is hypothesised that using a context-based approach for labelling algal blooms would result in more generalised models applicable for different locations.*

   This is implemented using a logarithmic polynomial function to label the data and train a novel detection model to test its generalisability among different locations.

3. *It is hypothesised that various modes of data could be used simultaneously for detecting algal blooms and would result in better models compared to unimodal detection models.*

   This is explored by gathering different modes of data such as satellite and in-situ data, training a fusion model and comparing unimodal baselines with the results.

1.5 Contributions

The contributions of this project include:

1. An imputation model for the four moorings of the Centre for Environment, Fisheries and Aquaculture Science (CEFAS) with a self-attention component.

2. A prediction model that consists of a transformer network and convolutional components to predict the anomalous behaviour of phytoplankton from the imputed data. This part of the work includes a flexible labelling method for creating classes from the daily mean of observed dissolved oxygen data.

3. An explanation model for the importances of input features with relation to the output variable for the previously created prediction model.

4. An additional prediction model that builds onto the second contribution that uses satellite imagery data and convolutional architectures that aim to improve prediction performance.

Each chapter of this thesis was partially published as a conference article:

- Chapter 3: **Dagtekin, Onatkut**, and Dethlefs, Nina. *Imputation of Partially Observed Water Quality Data Using Self-Attention LSTM.* The 2022

International Joint Conference on Neural Networks.

- Chapter 4: **Dagtekin, Onatkut**, and Dethlefs, Nina. *Modelling Phytoplankton Behaviour in the North and Irish Sea with Transformer Networks.* Proceedings of the Northern Lights Deep Learning Workshop. 2022.
- Chapter 5: **Dagtekin, Onatkut**, and Dethlefs, Nina. *Multimodal Approach to Early Detection of Harmful Algal Blooms.* Workshop on Machine Learning for Earth Observation: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. 2022.

1.6 Outline

This thesis consists of four chapters. Chapter 2 includes related work for the machine learning and deep learning models used for the task of data imputation and outlier detection in environmental science. Chapter 3 includes information about recent approaches to data imputation using machine learning and deep learning and introduces a novel architecture that lowers the imputation error for the used dataset. Chapter 4 includes information about the recent approaches to algal bloom detection that use different modes of data and introduces a novel model for detecting anomalous behaviour of phytoplankton 1 to 7 days before it occurs. This chapter also includes explanation models for the prediction model to observe how input variables affect the output of the model. Chapter 5 includes information and experiments about introducing multimodality to the problem of algal bloom detection using various types of satellite data. This chapter also introduces the use of coordinated representations to replace data modalities for the problem of HAB detection.

Chapter 2

Literature Review

Algal blooms have been historically documented for centuries, with the phenomenon receiving attention in the 1970s with the first conference on the topic (Anderson et al., 2002; Smith and Daniels, 2018; Shumway et al., 2018). Algal blooms can be naturally occurring and keep natural cycles in balance. The issue arises when certain algal species that produce toxins increase their population in a habitat resulting in the water body becoming inhabitable. These are called HABs.

This chapter includes essential information about how the phenomena of algal blooms occur, their effects and changes in how and where they may happen. The next part includes information about monitoring this event, starting from issues about monitoring and the nature of the problem in terms of observed data. Information on how to use machine learning and deep learning is also provided, such as the models that could be used for detecting algal blooms.

2.1 Algal Blooms

Algae are key species consisting of photosynthetic eukaryotes in a marine ecosystem for which they serve as the base of the food chain. They are the primary producers of oxygen in the world and provide aquatic environments sustenance through photosynthesis using their chloroplast, which also gives them their colour. The colour of the algae depends on the level they thrive in the water body such that algae that live on the surface are green coloured as they can easily access light, whereas the red coloured ones are deeper in the water body as less sunlight reaches the deeper parts and red coloured pigments aid them to capture more sunlight. They are useful for degrading plastic, thus reducing pollution (Chia et al., 2020).

HABs occur when the population of phytoplankton increases rapidly due to nutrient overload, causing environmental changes such as sunlight blocking and oxygen depletion (Kahru and Mitchell, 2008). These changes affect the ecosystem and public health since the consumption of aquatic life affected by these blooms poses a health risk (Falconer et al., 1994). HABs may occur due to eutrophication which is the increase of nutrients such as phosphorus and nitrogen in the water (Harper, 1992). The sources for these chemicals are pollutants such as cleaning products and fertilisers from agricultural activities. The colour of the bloom depends on the species in the water body; the reddish blooms occur in the ocean caused by dinoflagellates, and the green ones are caused by cyanobacteria. Figure 2.1 visualises the different types of algal blooms. Figure 2.3 visualises the process of algal blooms and eutrophication. Blooms are initialised by the entry of an extreme amount of nutrients into a water body, which facilitates algae growth, covering the surface. This blocks sunlight from reaching lower layers disrupting the photosynthetic activities of plants in those layers. This disruption results

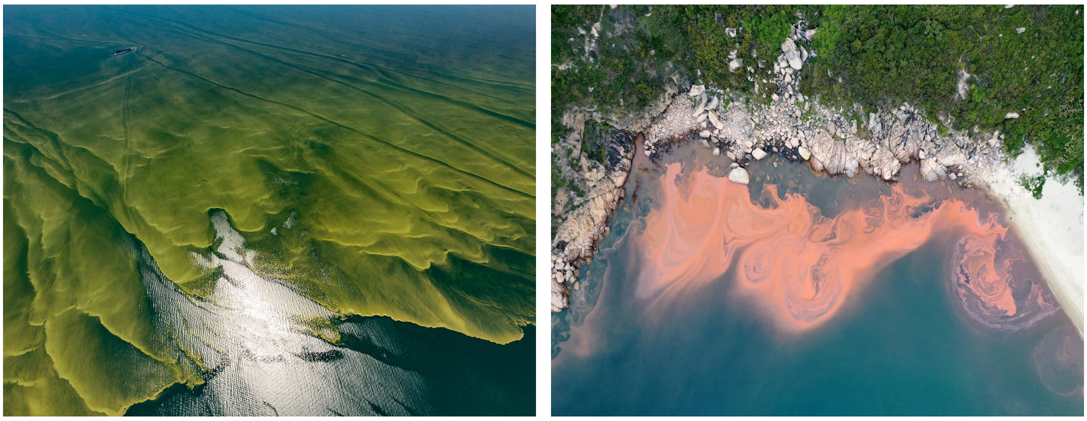

Figure 2.1: Left: An occurrence of a green coloured algal bloom (Zachary, 2017) Right: An occurrence of a red coloured algal bloom (Chau, 2022)

in a lowered concentration of oxygen which causes the death of heterotrophic species and an increase in the decomposer population. The increasing decomposer population further reduces the dissolved oxygen concentration, disrupting the habitat. The effects are propagated to the upper layers when the nutrient flow stops, and the algae start to die off and are decomposed.

This phenomenon can be frequently observed and has been studied in populated and highly industrialised areas of the world, as seen in Figure 2.2. The studies focus on Asia and the U.S., with the Mediterranean and the Baltic Sea being the second most studied. The North Sea and the Southern Hemisphere (with the exception of certain regions around Antarctica) are less studied. The latter could be explained by the economic status of countries and the availability to make complementary solutions to satellite imagery available.

2.1.1 Impact

Ecological Impact The increasing population of algae covering the water surface has an impact on the aquatic life in the lower layers. With no sunlight to perform photosynthesis, the food cycle gets disrupted, resulting in the deaths of

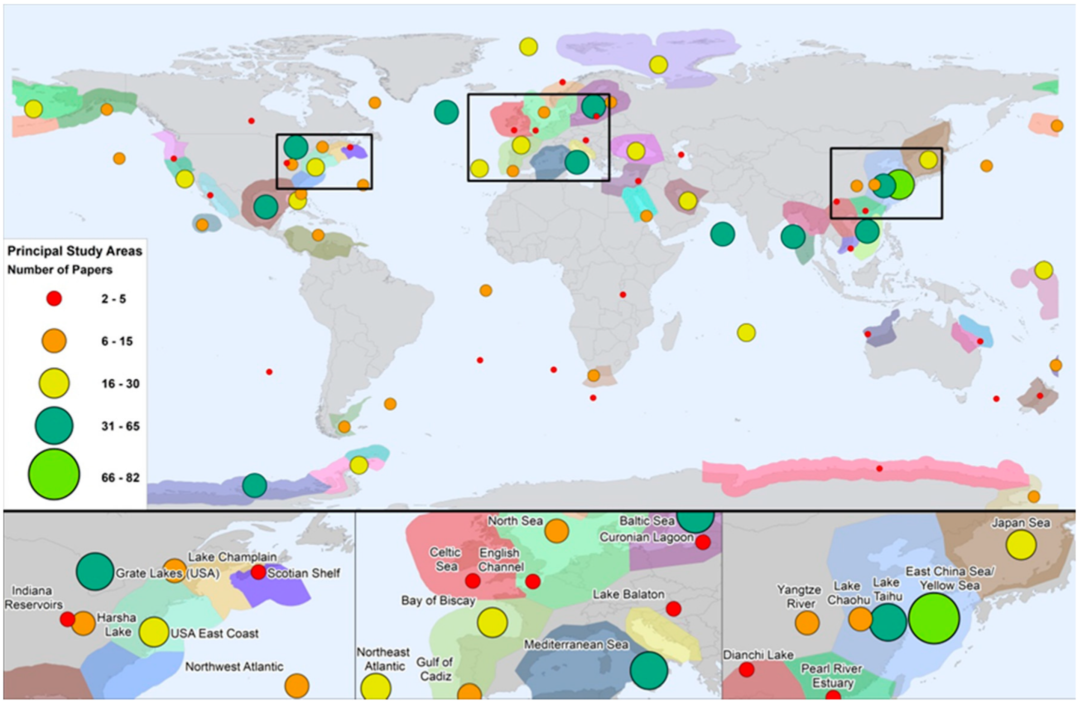

Figure 2.2: Regions where HABs were studied using satellite imagery Sebastiá-Frasquet et al. (2020). The focus is on The Great Lakes, Gulf of Florida in North America, The Mediterranean and The Baltic Sea in Europe, The Arabian Sea, The Bay of Bengal, The South China Sea, The East China Sea and Lake Taihu in Asia. The studies focus on mostly the northern hemisphere.

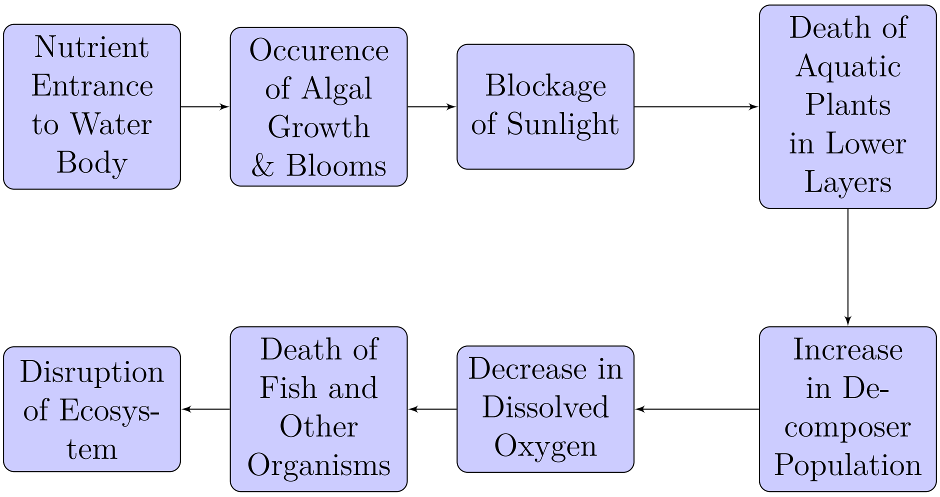
Nutrient Entrance to Water Body
Occurence of Algal Growth & Blooms
Blockage of Sunlight
Death of Aquatic Plants in Lower Layers
Increase in Decomposer Population
Decrease in Dissolved Oxygen
Death of Fish and Other Organisms
Disruption of Ecosystem

Figure 2.3: Flowchart of Eutrophication Process

species and disruption of the food web (Shumway et al., 2018). Severe effects of HABs include the creation of hypoxic zones that cannot support life. HABs cause various problems in aquatic life, such as reduced embryo development, damage to organs such as the liver and kidneys and inhibited growth (Shumway et al., 2018).

Normally, the transfer of toxins was through species of molluscs (oysters and clams); however, the dynamics have altered to include species in the food chain (Shumway et al., 2018). Seabirds consume the affected organisms, which may make them susceptible to environmental conditions and/or changes in their habitats and might cause problems during migrations Shumway et al. (2018).

Economic Impact These include drinking water treatment costs and an increase in the cost of preservation of biodiversity (Dodds et al., 2009). Regions where these blooms are frequent see lower sales in sectors related to tourism and lower income from fisheries (Bechard, 2020; Karlson et al., 2021). A recent bloom in 2019 in Norway, Tromsø and Nordland cost >100 million $ in damages to fisheries (Karlson et al., 2021). In 2009, a reduction in touristic and recreational activities was observed in Mocrocks Beach and Long Beach in Washington, US, due to algal blooms, which resulted in an estimated loss of 10.6 million $ (Dyson and Huppert, 2010). Occurrences of blooms can lead to the extension of monitoring programmes to include other species such as squids, octopus and fish, increasing the cost (Shumway et al., 2018).

Public Health Impact The toxicity depends on the species of algae present, some species are noted in Table 2.1. The toxins released by some species are many times more potent than cyanide and cobra toxin, showing the harm algal blooms are capable of. The bioaccumulation of these toxins causes an increase in concentration and causes harm to livestock, pets or the general population. The accumulated toxins cause various types of shellfish poisoning and sometimes lead to fatalities (Zingone and Enevoldsen, 2000).

The occurrence of the event also draws public attention in some cases, resulting in changes in public opinion. The warning given in north-west Ohio about the tap water being hazardous after an algal bloom affected the tap water usage in the general population up to a year (Ames et al., 2019). Annually 60.000 cases are reported due to phycotoxin-induced intoxications (Gerssen et al., 2010). Although the percentage of toxic species is very low, benthic species can act as vectors for toxins (Shumway et al., 2018).

2.1.2 Factors

The mechanisms of algal blooms are inherently complex. Shumway et al. (2018) categorise these factors into two:

1. Rate of changes in the introduction of species to new areas: Natural means; river currents or ship transportation activities; ballast water

Toxin	Source	Toxicity(fold)
Cyanide		1
Muscarin	Amanita muscaria, fungus	9
Okadaic acid	Algae, dinoflagellates (e.g. Dinophysis spp.)	50
Domoic acid	Algae, diatoms (Pseudo-nitzschia spp.)	80
Prymnesine	Algae, haptophytes (e.g. Prymnesium parvum)	350
Cobra toxin	Cobra snake	500
Saxitoxin	Algae, dinoflagellates (e.g. Alexandrium spp., Pyrodinium bahamense)	1 100
Ciguatoxin	Algae, dinoflagellates (Gambierdiscus toxicus)	22 000
Tetanus toxin	Bacterium (Clostridium tetanii)	1 000 000

Table 2.1: Toxicity of different compounds to mice Zingone and Enevoldsen (2000)

2. Rate of changes in current conditions to a more suitable one that aids the reproduction of species: Nutrient flow from external sources such as; industrial activities or storms

An algal bloom occurs when a species and nutrients “get there”, “are there”, and “stay there” (Shumway et al., 2018). Both of these factors must be sustained and satisfied to an extent for a bloom to occur. Human activities aid in the occurrence of the phenomenon by supporting (**2**), but it may not be the sole cause for these blooms to occur (Smayda, 2002). Non-anthropogenic examples of (**2**) include species interactions, nutrient flow in the ecosystem, and temperature changes (Sunda et al., 2006; Wells et al., 2015).

Getting There The main cause of the introduction of species to new areas is through the ballast water of ships (Hallegraeff, 2010). Harmful algae have been previously detected on ballast water discharge locations (Hallegraeff, 1998). Due to these discharges, previously rare algal species can reproduce and cause blooms (Rigby and Hallegraeff, 1996). The global transport patterns also show the spread of species, some of which are toxic, through different marine habitats by identifying the chemical and physical conditions and the algae and bacteria populations of ballast tanks from various ships (Burkholder et al., 2007b).

The increasing pollution is another factor for algal blooms. With the increasing population, new solutions were needed for agricultural practices, which came with more efficient fertilisers that contain nitrogen and phosphorus (Smil, 2004). This also brought increasing run-off of these elements to water bodies affecting the increased frequency of HABs. Animal husbandry is another cause of these blooms, as the waste produced by this process is high in nitrogen and phosphorus (Burkholder et al., 2007a; Mallin et al., 2015).

Being There The amount of nitrogen and/or phosphorus can cause an increasing number of blooms, with discharge increasing with population density (Shumway et al., 2018). Regions like the Baltic Sea are seeing more frequent blooms due to discharge from anthropogenic activities (Olenina et al., 2010). This is also observed in Sebastiá-Frasquet et al. (2020), with the Baltic Sea being studied frequently. The area covered by these blooms is also increasing due to this discharge from the increased use of fertilisers. In China, fertiliser use has increased by three times in the last 30 years, resulting in an increased frequency of algal blooms on the region's coasts (Ti and Yan, 2013).

In an undisrupted water body, algal succession occurs in a cycle as conditions change, with one species of algae dominating in each phase (Kelly and Linda, 1996). The increasing amount of nutrients will disrupt this cycle and cause algal blooms.

The existence of nutrient overload is not sufficient for algal blooms. The nutrient ratio of nitrogen:phosphorus is also a limiting factor (Shumway et al., 2018). Different algal species may be limited by different elements. Silicate is an important element for blooming as beneficial phytoplankton use it in their cell walls, but others do not (Shumway et al., 2018). Unlike nitrogen and phosphorus, silicate is not present in sewage; therefore, ratios of nitrogen and phosphorus with silicate have increased due to the increase in anthropogenic activities (Shumway et al., 2018).

Staying There A species holding ground in a habitat depends on physical factors. These factors affect both the population growth and nutrient density. Events such as upwellings where cold water at the bottom of a water body rises to the top, combined with nutrient overload may increase the occurrence of algal blooms (Shumway et al., 2018).

In off-shore environments, small-scale turbulence and stratification facilitate the development of these blooms (Shumway et al., 2018). Stratification causes cells to populate a certain layer in the water body, receiving light from above and nutrients from below.

Anthropogenic activities affect the conditions when and where these blooms occur. Dam constructions affect river flow and discharge and prevent the movement of organisms (Vörösmarty et al., 2010; Shumway et al., 2018). Construction of these dams affects regions such that the species are replaced by different ones due to water flow/salinity. One such case is the replacement of large diatoms with flagellates and cyanobacteria in the San Francisco Bay Delta over a span of 10 years (Lehman et al., 2005; Glibert and Burkholder, 2011).

2.1.3 Effect of Climate Change on Algal Blooms

Climate change will alter many environmental conditions, such as temperature, nutrients and light. This affects the species and/or nutrients to “be there” or “get there” (Shumway et al., 2018).With the increasing temperatures due to climate change, the frequency of algal blooms is expected to increase and be seen in new regions (Wells et al., 2015). Figure 2.4 visualises the main factors. It is speculated that an increase in sea surface temperature will trigger more blooms in the future (Sarkar, 2018).

The ice melt caused by climate change will affect stratification, nutrients, available light and grazing, affecting the occurrence of algal blooms (Boyd and Doney, 2003). The increasing temperature affects natural cycles and cell capability depending on species' optima (Shumway et al., 2018). The warming increases the toxicity of harmful algal species combined with the dissolved carbon dioxide in the water (Shumway et al., 2018; Davis et al., 2009; Fu et al., 2012).

Global warming will also affect the carbon cycle resulting in a pH increase in the oceans. Cyanobacteria thrive in acidic environments, which may increase blooms related to these species and dominate interspecies competition (O'Neil et al., 2012). Sensitive aquatic life might be endangered by the pH fluctuations during algal blooms (Kelly and Linda, 1996).

The rainfall patterns may alter due to climate change resulting in droughts or increasing extreme events such as storms and altering the water properties such as flow and nutrients (Shumway et al., 2018). This may result in a differing frequency of algal blooms.

2.1.4 Detection Methods

Two different approaches have been applied for HAB detection: using satellite data, analysed in Section 5.1, or in-situ data, analysed in Section 4.1. HAB detection using artificial intelligence can be done by predicting:

- chlorophyll-a (chl-a)
- dissolved oxygen
- toxins
- cell density

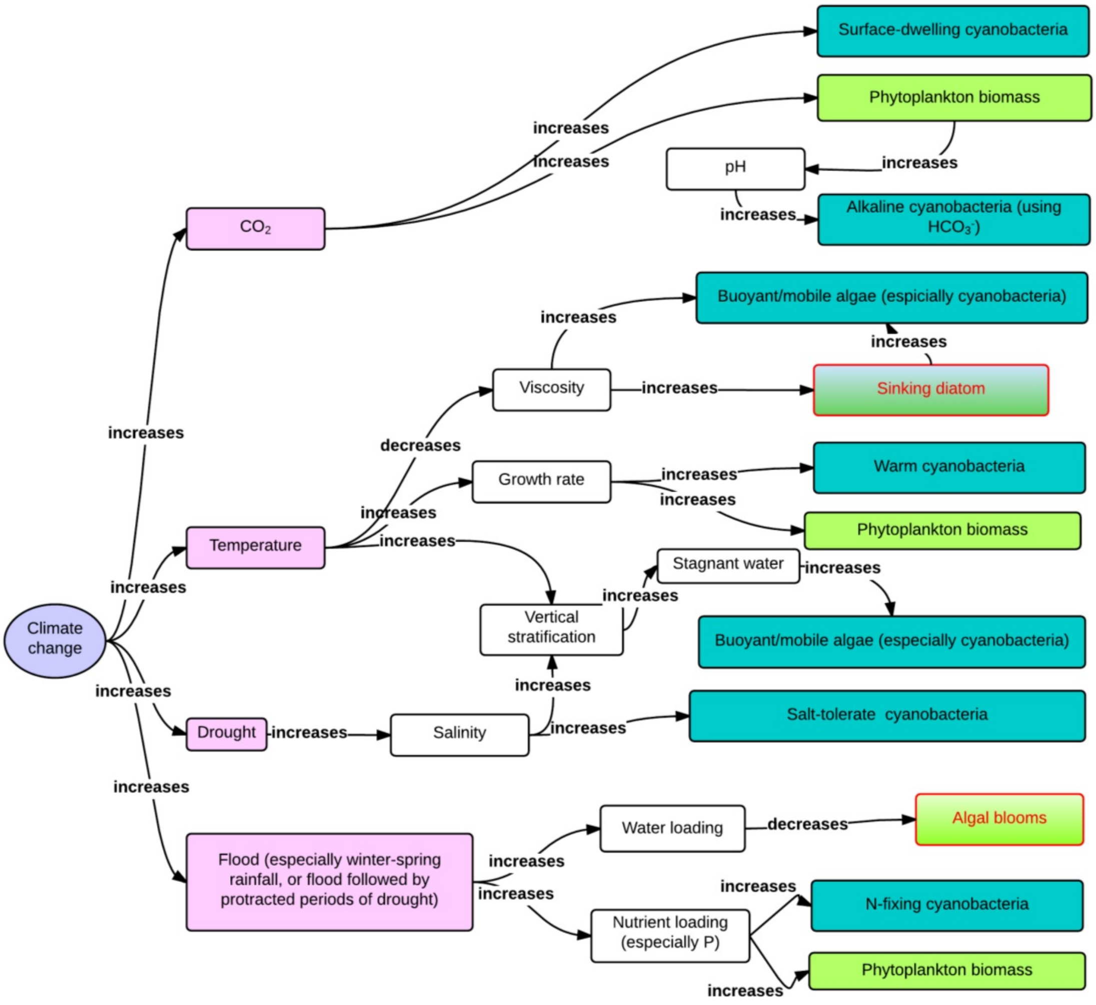

Figure 2.4: Climate change impact on algal blooms (Lin, 2017). Adapted from (Paerl and Huisman, 2008)

All of these variables initially increase with higher photosynthetic activity and/or cell reproduction. The chl-a concentration increases during an algal bloom due to increased photosynthetic activity. In contrast, the oxygen concentration increases initially with high photosynthetic activity and drops afterwards due to the increasing decomposer population as the algae start to die off. Bacteria use the dissolved oxygen in the water to decompose dead organisms, creating CO_2 in the process (Shukla et al., 2008). The chl-a concentration changes from species to species and during the day (Kelly and Linda, 1996). Similarly, oxygen concentration increases during the day but decreases at night as photosynthetic activity halts as no sunlight is received. The algae concentration may also differ through a water body, affecting the quality of in-situ chl-a and cell density measurements (Kelly and Linda, 1996). This can be addressed by frequent sampling and/or sampling various locations in a water body if possible. It should be noted that the behaviour of inland waters and seawater differ as seawater bodies can act like large reservoirs, so they are less susceptible to change. The majority of the works that use remote sensing data use chl-a as the target variable (Khan et al., 2021).

Algal blooms can be detected in various ways. Statistical methods could be used for detection. Shutler et al. (2012) use the approach of McKenna et al. (2000) with SeaWiFS and MODIS for the north-west European shelf near Shetland Isles, Scotland, comparing the results with in-situ data. A similar approach is used in Shutler et al. (2010) to detect blooms in the north-east Atlantic with SeaWiFS. Shukla et al. (2008) apply non-linear models to detect algal blooms by predicting the density of the algal population. Binding et al. (2018) use regression models and a rule-based approach to analyse past algal blooms in Lake Winnipeg, Canada. Autoregressive models such as Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) were widely used for algal bloom detection (Chen et al., 2015b; Qin et al., 2017; Al Shehhi and Kaya, 2021; Kim, 2016). These models have decreasing applicability as they cannot model non-linearities (Cruz et al., 2021).

Machine learning and deep learning methods could be used for detection as well. Common methods used include Recurrent Neural Network (RNN), Artificial Neural Network (ANN) and other classic machine learning methods like clustering and Support Vector Machine (SVM) or a combination of these methods (Huang et al., 2015; Kang et al., 2010).

2.2 Data Imputation

With the increasing availability of data collection, data ubiquity is observed in many domains. This deluge can cause a decreasing data quality with missing entries which must be filled before further analysis. The process of filling up missing data is defined as imputation. The first efforts of imputation were made by Allan and Wishart (1930) and Yates (1933), with the process formally defined by Dempster et al. (1977) (Van Buuren, 2018). Addressing the problem of missing data was revived by Rubin (1978) with multiple imputation, which still serves as a baseline today (Van Buuren, 2018).

There are different types of missing data; Missing Completely at Random (MCAR), Missing at Random (MAR) and Missing Not at Random (MNAR). MCAR occurs when missingness does not depend on any variables, such as when the data collection process is handled improperly, leading to low data quality. MAR occurs when missingness depends on the observed variables, such as sensors on a measuring device might shut down at certain times during the day when measurements are known to be certain values to conserve energy. MNAR occurs when missingness depends on both observed and unobserved variables such that sensors might not measure if a concentration of a compound is too high or too low. Formally, the missing data types can be defined as Van Buuren (2018):

$$\begin{aligned}
&MCAR : Pr(R = 0|Y_{obs}, Y_{mis}, \psi) = Pr(R = 0|\psi)\\
&MAR : Pr(R = 0|Y_{obs}, Y_{mis}, \psi) = Pr(R = 0|Y_{obs}, \psi)\\
&MNAR : Pr(R = 0|Y_{obs}, Y_{mis}, \psi)
\end{aligned} \tag{2.1}$$

where R is the missing data matrix, ψ is the parameters of the imputation model, and Y is the data matrix.

The occurrence of MAR indicates that the missing variables in a dataset can be derived from known ones by modelling the relationship between the missing and present variables. The only definite way of deciding on the type of missingness is to obtain the missing data, which is impossible (Little and Rubin, 2019). Due to the mechanism, it is impossible to directly test for MAR, but Little's Test for MCAR shows partial insight into the type of missingness of the data (Van Buuren, 2018). The test compares the mean vector and covariance matrix of cases of complete data and cases with missing data to test their identicality (Little, 1988). However, such tests are not widely used and are not practical as some part of the data might be missing systematically (Van Buuren, 2018; Jaeger, 2006; Enders, 2010). There is no such test for the distinction between MAR and MNAR (Van Buuren, 2018). In the context of water quality, the statistical methods show that the conditions of MAR are satisfied such that the potassium (K) could be imputed using the sodium (Na) values due to correlation (Güler et al., 2002).

The model parameters for imputation can be learned by randomly omitting parts of complete samples and modelling the relationship between missing and known variables per sample. After the imputation process, predictions about variables can be made. Tasks include outlier detection in cycles and early warning systems.

According to Rubin (1978), there are two challenges in data imputation:

- Imputing a value will never be absolutely correct. If it were, then that value would not have been missing.
- To reasonably impute a value, you need to create a model that maps missing data to observed data.

Therefore, a model needs to be generalisable to overcome these challenges. In addition to generalisability, the models need to be explainable to convey the meaning of the data to other researchers (Rubin, 1978).

Deep learning and machine learning methods have been used extensively for imputation. Zhang et al. (2019a) use k-Nearest Neighbours (k-NN) and linear regression for imputation. Choudhury and Kosorok (2020) apply a modified k-NN that uses mutual information to the missing data problem while taking the labels of the samples into account. Yilmaz and Aydin (2019) use k-NN imputation for simulated data. Santos et al. (2020) use k-NN with a modified distance metric for heterogeneous data. The main disadvantage of k-NN is the computation time and memory requirement for larger datasets, the issue of incorrect predictions for imbalanced data and the difficulty of hyperparameter choices. The intuitive approach and non-parametric approach make it suitable for small datasets. Linear regression is not suitable in cases where data has many outliers and is non-linear.

Variations of Singular Value Decomposition have been used for imputation (Troyanskaya et al., 2001; Mazumder et al., 2010; Cai et al., 2010). Shu et al. (2014) and Papadimitriou et al. (2013) apply Principal Component Analysis based approaches for data imputation. Principal Component Analysis is compatible with linear data and when the covariance of the dataset is important. Caillault et al. (2020) implement Dynamic Time Warping (DTW) to impute missing data in various datasets. DTW is computationally intensive for long sequences. Spatio-temporal approaches have been adopted for imputation, where data is collected for tasks such as traffic tracking and video surveillance. Yi et al. (2016) enable multi-view learning in the temporal and spatial domain in global and local views for data imputation enabling the model to use various information to impute the data. Liu et al. (2019b) impute data using a non-autoregressive approach with a divide and conquer approach, imputing data based on past values and the predicted future values, which removes the compounding error of autoregressive models.

Folguera et al. (2015) use Self-Organising Maps (SOMs) for imputing missing variables. SOMs enable to visualise clusters in the data, increasing interpretability. Mulia et al. (2015) implement an ANN with a Genetic algorithm for imputation, enabling the learning of non-linearities in the data. Auto-encoders have been extensively applied for the task of data imputation in several domains (Boquet et al., 2019; Beaulieu-Jones et al., 2017; Tran et al., 2017). Auto-encoders can reduce dimensionality non-linearly with higher generalisation, unlike methods such as Principal Component Analysis. Bansal et al. (2021) apply kernel regression, convolutions, and multi-head attention for data imputation. Generative Adversarial Network (GAN) architectures have been adopted for data imputation (Yoon et al., 2018; Lee et al., 2019; Luo et al., 2018). However, GANs have problems such as; non-convergence where parameters oscillate and mode collapse where the generator overfits to a subset of the data. Cao et al. (2018) use recurrent components for imputation and assume the missing values belong to the RNN graph. Using RNN enables the modelling of temporal properties. The transformer architecture introduced by Vaswani et al. (2017) has also been utilised for imputation (Sucholutsky et al., 2019). The main advantage of transformers over RNN models is that during the training phase, the transformer does not need to unfold the whole sequence and process it while reducing training time. All methods mentioned in this part are deep learning methods which come with the disadvantage that they require large amounts of training data and GPUs to train rapidly and efficiently. The majority of deep learning models contain a high number of parameters, making them unexplainable. Attention models (Section 2.5.4) and explainability models (Section 4.1.2) attempt to summarise models in various ways to alleviate this issue.

2.3 Anomaly Detection

Anomaly detection is defined as the identification of uncommon events that deviate from the dataset's normal behaviour. An example is credit card fraud, where a stolen credit card can be blocked by identifying anomalous transactions that deviate from the distribution. Other tasks include intrusion detection, air quality etc. (Buczak and Guven, 2015; Chen et al., 2017). Commonly used methods for anomaly detection include GAN, Variational Autoencoder (VAE), distance-based models, and clustering-based models (Chalapathy and Chawla, 2019).

There are three different types of anomalies Mehrotra et al. (2017):

- *Point anomaly:* This anomaly is caused by deviation from previously known data points, as in the example of credit card fraud
- *Contextual anomaly:* Data points can be considered anomalous given the context. Power consumption in residential areas might be lower at night and higher after work hours. Given the context of time, it might be deemed anomalous when power consumption spikes in the middle of the night.
- *Discords or collective anomalies:* This type of anomaly occurs when a region of a time series is entirely different from the rest, and the irregularity is encountered multiple times over observations. This is encountered in medical data where irregularities during monitoring might indicate an illness or disease.

Point anomalies can occur in any dataset as they are independent, collective anomalies require relationships between data points such as time series, and contextual anomalies depend on the contextual information in the data (Chandola et al., 2009).

Missing data can be considered a contextual anomaly as the observations might depend on unseen conditions during data collection. Photosynthetically Active Radiation (PAR) depends on sunlight hours such that after no light is received, the observations might include missing data for the PAR variable.

Algal blooms rarely occur, making them anomalies. Due to the nature of the event, they can be both contextual anomalies depending on the monitored variables or collective anomalies, as during the incident, spikes can be observed for certain variables, which are outlined in Section 2.1.4.

Anomaly Detection in Environmental Science

Many domains use deep learning to detect anomalies ranging from cybersecurity to medicine (Chalapathy and Chawla, 2019). Environmental science utilises many methods for anomaly detection.

Toxic metals like mercury, arsenic and lead might be present in water. These pollutants must be tracked to ensure public health and safety. ANNs have been used to predict toxic metals in rivers (Singha et al., 2020).

Agricultural activities must be monitored closely to ensure that the finest products are obtained with the best practices, and any disruptions must be detected. Examples include the detection of plant diseases and fruit grading using Convolutional Neural Networks (CNNs) (Sladojevic et al., 2016; Ismail and Malik, 2021).

Air quality monitoring and detection of abnormalities are crucial for public health and transportation. Tong et al. (2019) use bidirectional Long-Short Term Memory (LSTM) networks to track 2.5 particulate matter. Effects of climate change can also be observed by detecting anomalous zones in the ozone (Harrou et al., 2018).

2.4 Machine Learning

Machine learning is the process of developing software that can identify patterns in data using heuristics and apply the learned patterns to future data to make predictions without the use of explicit programming (Alpaydin, 2020). The learning process can be divided into two: training (learning) and testing (inference). During the training phase, the model minimises error according to an objective function under a number of constraints. During the testing phase, the model is given previously unseen data, and its generalisability is tested. Depending on the task, different learning paradigms are applied.

Supervised Learning In this scenario, the labels for the samples are known, and the model creates connections between the data X and the labels Y under certain assumptions (Alpaydin, 2020). The tasks can be divided into two categories: regression and classification (Alpaydin, 2020). Classification is the labelling of samples into different pre-determined categories, and regression is the prediction of continuous variables (Goodfellow et al., 2016). The types of classification include multiclass classification, where the possible number of classes is $n > 2$ and multilabel classification, where a sample can be assigned $n > 1$ labels.

Unsupervised Learning In this scenario, the labels for the samples are not known, and the model generates labels from the data under certain assumptions. The most common approach for unsupervised learning is clustering, where a model classifies samples by identifying common features (Russell and Norvig, 2002).

Semi-supervised Learning This type of learning is between supervised and unsupervised learning, as the learning is done with partially labelled data. The

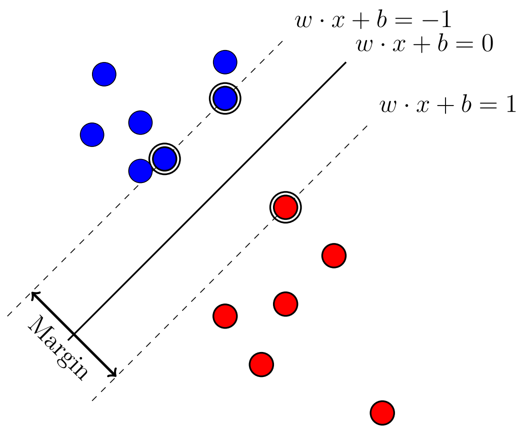

Figure 2.5: Visualisation of SVM. The support vectors are illustrated with dashed lines and the relating sample with double circles.

unlabelled data is tagged using a learning model, and further training is done with a complete dataset. With the deluge of unstructured data for real-world tasks, this type of learning is becoming more common (Goodfellow et al., 2016).

Reinforcement Learning In this scenario, the learning is done using an agent and its interactions with a dynamic environment. During an interaction, the agent receives a reward or punishment, moving closer to a state where it learns to perform the specific task (Russell and Norvig, 2002).

2.4.1 Support Vector Machine

SVM, developed by Vapnik (1963), is a linear method that aims to minimise the objective function in Equation 2.2 using a number of support vectors on supporting hyperplanes which maximises the margin between classes where the optimal margin is the inverse of the weight vector w.

$$E = \sum_{i=1}^{N} max(0, 1 - y_i f(x_i) - e_i) + \frac{1}{2}\sum_{j=1}^{d} w_j^2 + C\sum_{i=1}^{N} e_i \tag{2.2}$$

Alternatively, the objective function can also be defined in Dual Lagrangian form as in Equation 2.3.

$$L = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \tag{2.3}$$

The first term in the equation is the hinge loss. The second term of the equation is the margin between the decision boundary and the support vectors. The last term, e_i, is the slack variable which gives the model the ability to make slight misclassifications enabling a "soft" margin SVM. The coefficient C is the regularisation parameter for the softness of the margins. Hinge loss is undifferentiable at point $x = 1$; as a result, methods like stochastic gradient descent become unusable. Therefore, this optimisation problem can be solved by quadratic programming. SVMs can be trained with small amounts of data and will always reach a global optimal given that the dataset is linearly separable. An SVM is visualised in Figure 2.5.

Kernel Trick

The vanilla SVM is only able to classify linearly separable data as the mapping of classes is; $w^T x - c$. An extension of SVM can separate non-linearly separable data using the kernel trick developed by (Boser et al., 1992). The kernel trick implicitly transforms the space data into a new space in which it is linearly separable.

The weights in SVM take the form of $w_j x_j$, which makes the classification function equal to $f(x') = (\sum_{j=1}^{N} \alpha_j w_j x_j)^T x'$. In the new space ω, the exact transformations do not need to be calculated for each point, as only $\omega(x_i)^T \omega(x')$ is needed, i.e., the dot products of these two points in the transformed space, which is supplied by the kernel that satisfies Mercer's Theorem. Commonly used kernels are polynomial, $(1 + x_i^T x')^d$, and Gaussian, $e^{-\gamma(x_i - x')^2}$.

SVMs have been used for several tasks like facial expression classification, text classification and sound classification (Dino and Abdulrazzaq, 2019; Colas and Brazdil, 2006; Uzkent et al., 2012). SVMs are frequently used in algal bloom detection (Li et al., 2014; Vilas et al., 2014; Yang et al., 2020).

2.4.2 Random Forest

Random Forest (RF) is an ensemble of decision trees that perform their task via voting, with each tree in this model being slightly different from another (Breiman, 2001). In classification, the voting process is done by mode, whereas in regression, it is done by mean voting. RFs create different decision trees by bagging the data, i.e. creating small subsets of the data with or without repeating the previous data points and using this subset to train a tree (Breiman, 1996). While creating the splits for the trees, the algorithm can also take a subset of features into account, further diversifying the trees in the ensemble (Ho, 1998). This process is done numerous times, creating n trees. The main aim is to combat overfitting caused by a single decision tree. The tree nodes are split based on Gini impurity (Equation 2.4) or information gain (Equation 2.5).

$$I_G = 1 - \sum_{i=1}^{C}(p_i^2) \tag{2.4}$$

$$\begin{aligned} I_E &= \sum_{i=1}^{C}(-p_i * log_2)(p_i) \\ I_G &= I_E - I_{E|X} \end{aligned} \tag{2.5}$$

RF is less prone to overfitting, given the diversity of the individual decision trees. These models are interpretable in terms of input-output importances. It is able to show which feature is the most important one for making predictions. The inference procedure of an RF is visualised in Figure 2.6.

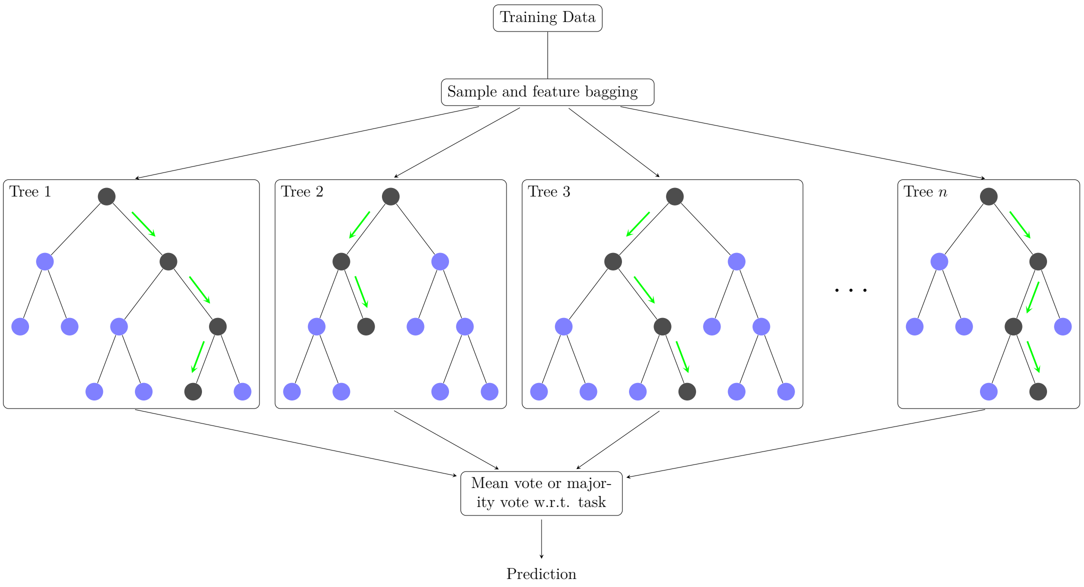
Training Data
Sample and feature bagging
Tree 1
Tree 2
Tree 3
. . .
Tree n
Mean vote or majority vote w.r.t. task
Prediction

Figure 2.6: Visualisation of RF

RFs have been used for various tasks, such as land cover classification and traffic accident detection (Rodriguez-Galiano et al., 2012; Dogru and Subasi, 2018). It has also been used for algal bloom detection (Yang et al., 2020; Derot et al., 2020; Yajima and Derot, 2018).

2.4.3 Isolation Forest

A well-known method of detecting outliers is the Isolation Forest (IF) method (Liu et al., 2008). The model works based on two properties:

- the anomaly is the minority class
- the anomalies have different attributes than normal instances

The notion of isolation comes from the idea that in a tree, anomalies usually create shorter path lengths as they are different from the majority of the data. Using many trees results in different trees targeting different anomalies (Liu et al., 2008). It should be noted that IF is an unsupervised model.

In cases where anomalies are close to normal instances, normal trees create long path lengths concealing the anomaly, which also leads to incorrect classification of the normal data points. With the use of subsampling, the probability of concealment is reduced as the tree only uses a fraction of the data, therefore creating a shorter path length for the anomaly.

This method has been used for network anomaly detection, credit card fraud and fault detection on electricity generators (Tao et al., 2018; John and Naaz, 2019; Hara et al., 2020). IFs have been used for algal bloom detection (Mehrabian and Pahlevan, 2019; Almuhtaram et al., 2021).

2.4.4 XGBoost

Developed by Chen and Guestrin (2016), eXtreme Gradient Boosting (XGBoost) is a highly scalable tree boosting model using an ensemble of trees. The model builds its work upon gradient tree boosting, unlike ensemble models such as RF. Ensemble trees can be defined as (Chen and Guestrin, 2016):

$$y_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{2.6}$$

where $F = f(x) = w_{q(x)}(q : R^m \to T, w \in R^T)$ is the possible space of regression trees. q is the structure of each tree and produces an output from the inputs. T is the number of leaves in the tree. f_k is a function of the tree structure q and leaf weights w. Each tree is trained such that tree t is built greedily on the errors of the previous trees. Unlike decision trees, the leaves do not contain the class label from that tree but continuous values (Chen and Guestrin, 2016). To obtain a final prediction, the values in the leaves are summed. The regularised objective function for XGBoost is as follows (Chen and Guestrin, 2016):

$$\begin{aligned} L &= \sum_i l(\hat{y}_i, y_i) + \sum_i \Omega(f_k) \\ \Omega(f_k) &= \alpha T + \frac{1}{2}\lambda ||w||^2 \end{aligned} \tag{2.7}$$

where $l(\hat{y}_i, y_i)$ is the loss function for the prediction and target value, $\Omega(f_k)$ is the regularisation component to avoid overfitting, and T is the number of leaves in each tree.

The tree models are trained in an additive manner using gradient tree boosting. For each tree prediction, y_i^t, the following objective is minimised:

$$L^t = \sum_i l(\hat{y}^{t-1}, y_i + f_t(x_i)) + \sum_i \Omega(f_t) \tag{2.8}$$

where $l(\hat{y_i}^{t-1}, y_i)$ is the loss function for the prediction and target value for tree t. The optimisation is done with second-order approximation, resulting in the following equations for optimum weight per leaf, w_j, and minimum loss per tree, $L^t(q)$:

$$\begin{aligned} w_j &= -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} + \lambda} \\ L^t(q) &= -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \alpha T \end{aligned} \tag{2.9}$$

where I_j is the instance set of leaf j. Unlike decision trees, the splits are not based on Gini impurity or information gain but a different equation:

$$L_{split} = \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \alpha \tag{2.10}$$

where I_L and I_R are the instance set of left and right split, respectively, and $I = I_L \cup I_R$. Two additional regularisation methods are used during training; shrinkage and column subsampling. Shrinkage scales newly added weights by a constant factor to facilitate learning by reducing the importance of each existing tree and the future trees to improve the performance (Chen and Guestrin, 2016). Column subsampling selects a subsample of features, same as in RFs, reducing the training duration.

With its parallelisability and fast optimisation, XGBoost is being used in various tasks, such as; intrusion detection, accident detection and leakage detection in water networks (Jiang et al., 2020; Parsa et al., 2020; Wu et al., 2022). XGBoost was utilised for algal bloom detection, both stand-alone and in conjunction with other models (Shan et al., 2022; Ghatkar et al., 2019; Izadi et al., 2021).

2.5 Deep Learning

Deep learning is a sub-area of machine learning where stacked neurons that can have different characteristics are interlinked to solve complex problems. The inspiration comes from the human nervous system, where neurons in a network "fire" when sensible input is received. At each step along the way, the received input becomes more and more abstract, opening up the possibility to learn representations and extract features at different depths, removing the need for a feature extraction step in a workflow (Najafabadi et al., 2015). The model's parameters are adjusted using optimisation methods such as stochastic gradient descent and backpropagation, where the weights and biases of the model are updated starting from the final layer to the initial layer (Bottou, 2010).

A neural network may contain a large number of parameters. Therefore, using slower components like CPUs might not be feasible. The introduction of GPUs to training neural networks by Steinkraus et al. (2005) and Chellapilla et al. (2006) created new possibilities for the rapid development of neural networks, resulting in increased popularity with general process GPUs being used for training neural networks. Since then, deep learning has been used for various tasks in domains such as Natural Language Processing (NLP), computer vision, environmental science and finance (Mayr et al., 2016; Heaton et al., 2017; Pham et al., 2018; Jabreel and Moreno, 2019).

In contrast to machine learning, deep learning models require huge samples of data to make sense of it, as the models have a large number of parameters. With the increasing capabilities of sensor technology, an increase in the use of deep learning can be observed in environmental science with deep learning methods gaining traction in the agricultural and water quality domain (Kamilaris and Prenafeta-Boldú, 2018; Chen et al., 2020).

2.5.1 Multilayer Perceptron

Multilayer Perceptron (MLP)s, synonymously called ANNs or deep feed-forward networks, consist of at least three layers of nodes; an input layer, hidden layer and output layer, each consisting of neurons and feed data from one layer to the next. Each neuron multiplies the input received from the previous layer by weights and adds them together with a bias term. The resulting value is passed to an activation function, resulting in a non-linear output, and fed as input to the next layer. Common activation functions include ReLU (Nair and Hinton, 2010), Softmax (Bridle, 1989), Tanh and Sigmoid (Han and Moraga, 1995). Additions for regularisations can also be made with components such as batch normalisation and dropout layers. The structure of an MLP is visualised in Figure 2.7.

MLPs have been utilised for many tasks such as stock market prediction, weather forecasting and water quality prediction (Narvekar and Fargose, 2015; Billah et al., 2016; Sarkar and Pandey, 2015). MLPs have been used for algal bloom detection in various regions (Luo et al., 2017; Shamshirband et al., 2019;

Muttil and Chau, 2006).

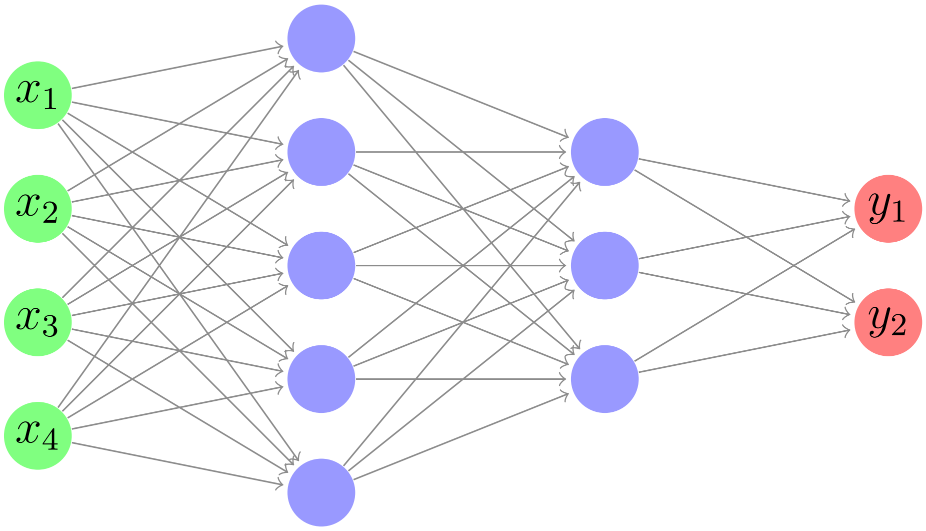

Figure 2.7: MLP with two hidden layers

2.5.2 Convolutional Neural Networks

MLPs are suitable for data where each datapoint is independent. Data formats such as images are not suitable to model with MLPs, as images contain spatial information. CNNs solve this issue by including the information from the vicinity of the datapoint in the modelling procedure using 2D kernels. The components of CNNs usually include a number of convolutional layers of different sizes, activation functions and pooling to overcome the issue of overfitting and supply translation invariance. Common architectures of CNNs include VGG, U-net and Res-Net. Many of these architectures also make use of transpose convolution layers, upsampling layers and skip connections.

The convolutional operation for images takes place on a 2D plane. 1D convolutions can also be used for data like time series. The only difference is that the filter dimensionality is reduced. Figure 2.8 visualises the operation that takes place in 2D convolution.

CNNs are used in many tasks where image data is involved, such as object detection, segmentation and tracking (Ren et al., 2015; Milletari et al., 2016; Son et al., 2017). CNNs have been utilised for HAB detection, where satellite imagery is used as the data source (Hill et al., 2020; Cao et al., 2022; Park et al., 2019; Pyo et al., 2020).

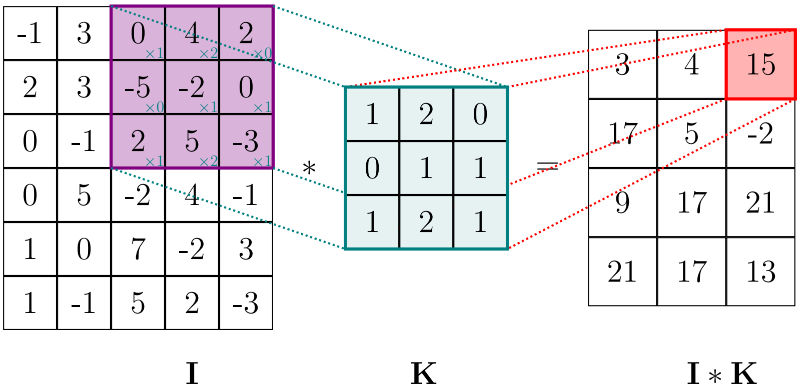

Figure 2.8: Sample convolution operation

2.5.3 Recurrent Neural Networks

ANNs cannot handle the modelling of data with temporal properties, as it assumes prior outputs do not affect the current output. RNNs were introduced to tackle this problem (Rumelhart et al., 1985). The backpropagation of RNNs differ from ANNs such that each time step in the input is included sequentially from end to start during weight updates, visualised in Figure 2.9 (Werbos, 1990). This enables the model to remember past information, but it may cause issues such as vanishing and exploding gradients that affect the model's ability to learn and generalise (Pascanu et al., 2013; Bengio et al., 1994).

To tackle the issues about the gradient, variations of RNNs were proposed; LSTM (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Chung et al., 2014). Additions were made to the architectures with attention

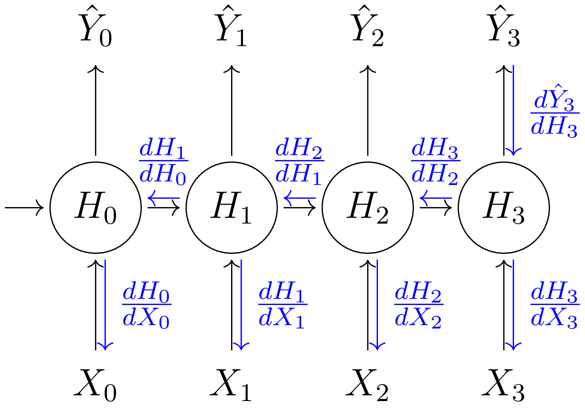

Figure 2.9: Backpropagation through time in RNN

components that focus on necessary parts of the input (Bahdanau et al., 2014; Luong et al., 2015). The concept of gradient clipping was also introduced to tackle this problem (Mikolov et al., 2012).

Unlike ANNs, RNNs can also be used for a variety of problems with various mappings; one-to-one, one-to-many, many-to-one and many-to-many. One-to-one relationships are where the sample is a single input and output, such as image classification (Liu et al., 2017). ANNs are only able to model one-to-one relationships. One-to-many tasks are where the sample is a single input, but the output is a sequence. Tasks include image captioning and pose estimation (Lee et al., 2018; Li and Chen, 2018). Many-to-one relationships where the input is made up of multiple elements such as time series and the output is a single variable. Examples include classifying a sequence from n different classes, such as tweet classification or predicting a continuous value, such as stock value prediction (AL-Rashdi and O'Keefe, 2019; Liu et al., 2018). In many-to-many relationships, both the input and the output include varying number of elements. Machine translation is an example of a many-to-many relationship (Huang et al., 2018).

Long Short Term Memory

Developed by Hochreiter and Schmidhuber (1997), LSTM addresses the problem of extreme changes in the weights of cells in traditional recurrent neural networks. The cell's outputs and gate mechanisms enable it to model the recent context rather than the last input only, which increases its predictive power. LSTMs have been used for tasks such as flood prediction and stock prediction (Chen et al., 2015a; Le et al., 2019). The structure of an LSTM cell is depicted in Figure 2.10.

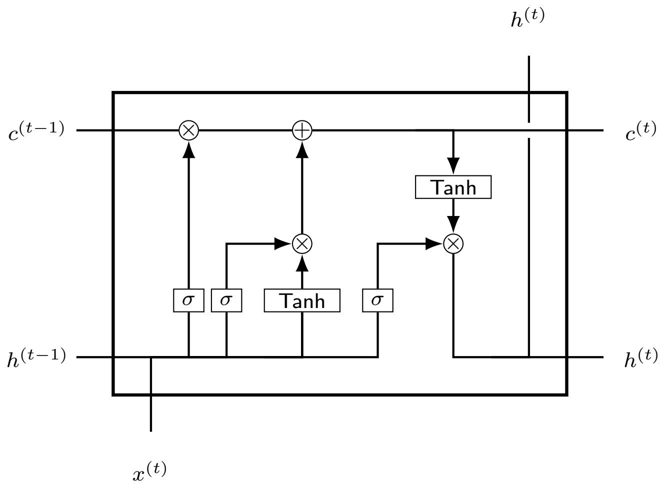

Figure 2.10: Structure of an LSTM cell

The outputs of the cell are as follows Hochreiter and Schmidhuber (1997):

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
\tag{2.11}
$$

where h_t is the hidden state, c_t is the cell state at time t, and x_t is the input at time t. h_{t-1} is the hidden state at $t-1$. i_t, f_t, g_t, o_t are input, forget, cell, and output gates. σ is the sigmoid function, and $\odot$ is the element-wise product. These gates control the flow of information through the cell, removing old or incorporating new information. Miscellaneous additions were also made, such as adding peephole connections where input, forget, and output gates are allowed to incorporate information from the cell state (Gers et al., 2002).

The bidirectional LSTM is an advancement of the LSTM that captures information both from the past and the future with a backward and a forward pass, using a separate state for each, incorporating knowledge from both directions for prediction (Schuster and Paliwal, 1997). The structure of a bidirectional LSTM is visualised in Figure 2.11. The operations in bidirectional LSTMs are as follows:

$$
\begin{aligned}
h_f^t &= f(w_{h_1}^f x_t + w_{h_2}^f * h_{t-1}^f + b_h^f) \\
h_b^t &= f(w_{h_1}^b x_t + w_{h_2}^b * h_{t+1}^b + b_h^b)
\end{aligned}
\tag{2.12}
$$

The resulting values h_f^t and h_b^t at each time step are concatenated and forwarded to other layers for prediction.

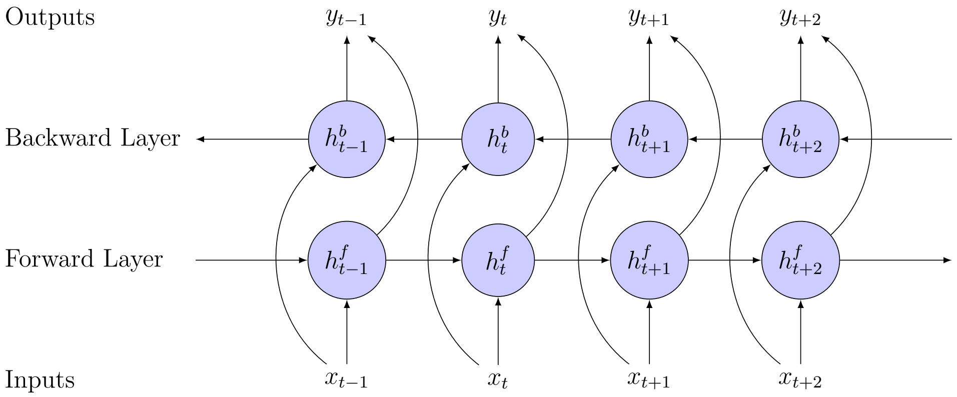

Figure 2.11: Bidirectional LSTM

LSTMs have been used for tasks such as text classification and sentiment analysis (Xu et al., 2017, 2019). They have been used for HAB detection (Cho et al., 2018; Shin et al., 2020).

2.5.4 Attention Models

Attention-based learning aids a model in shifting its focus on a number of inputs while evaluating the current one. Attention components are frequently used in encoder-decoder architectures, such as in (Sutskever et al., 2014). The use of an encoder-decoder architecture pertains that the data dimensionality will be reduced, and only relevant information will be used during the training and prediction stages. Attention is divided into various types, such as local and global attention (Luong et al., 2015), self-attention or intra-attention (Vaswani et al., 2017). Global and local attention is used to explain the behaviour of a neural network with respect to the relationship between input and output, whereas self-attention is used to explain relationships between input elements.

Bahdanau Attention

This model adds the use of context vectors to the encoder-decoder architecture. The ith element of the context vector focuses on the words surrounding the ith input. The conditional probability of the outputs can be denoted as follows (Bahdanau et al., 2014):

$$p(y_i|y_1, ..., y_{i-1}, \mathrm{x}) = g(y_{i-1}, s_i, c_i) \tag{2.13}$$

where s_i is the hidden state at time i, c_i is the context vector. The context vectors are dependent on a sequence of annotations $(h_1, ..., h_{T_x})$ which contains information about the dependencies between the inputs (Bahdanau et al., 2014). The context vector is a weighted sum of the elements of these annotations (Bahdanau et al., 2014). The weight is calculated by:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.14}$$

where e_{ij} is an alignment model, which shows how input i and output j are related based on the output of the hidden state s_{i-1} and the annotation h_j. The alignment model is learned similar to a feed-forward network (Bahdanau et al., 2014). The connections between these variables result in generating the next step of the decoder and, ultimately, the output of the next decoder state (Bahdanau et al., 2014). The encoder uses a bidirectional RNN architecture to focus on the future inputs and the past inputs by concatenating the forward and the backward encoder hidden states (Bahdanau et al., 2014).

Luong Attention

This type of attention by Luong et al. (2015) introduces global and local attention and different types of functions for the alignment model. There are three types of functions for the global attention:

$$score(h_t, \bar{h_s}) = \begin{cases} h_t^T \bar{h_s} & dot \\ h_t^T W_a \bar{h_s} & general \\ v_a^T \tanh(W_a[h_t; \bar{h_s}]) & concat \end{cases} \tag{2.15}$$

The variable-length alignment vector can be calculated as Luong et al. (2015):

$$a_t(s) = align(h_t, \bar{h_s}) = \frac{exp(score(h_t, \bar{h_s}))}{\sum_{s'} exp(score(h_t, \bar{h_{s'}}))} \tag{2.16}$$

The encoder for this model uses the hidden states at the top of the LSTM layer instead of concatenating backward and forward layers like Bahdanau et al. (2014). This model also avoids the recurrence applied by Bahdanau et al. (2014) and uses more diverse functions.

The local attention mechanism focuses on a smaller subset than global attention to diminish the effects of distance. The process is as follows (Luong et al., 2015):

- Generate aligned position p_t for time t.
- Derive context vector c_t by applying weighted average on the set of source hidden states with a window size of D, i.e., $[p_t - D, p_t + D]$, which is chosen empirically.
- Create a local alignment vector a_t with a fixed size of $2D + 1$.

Depending on the type of alignment (monotonic or predictive), different alignment vectors are created. The mentioned attention types are visualised in Figure 2.12.

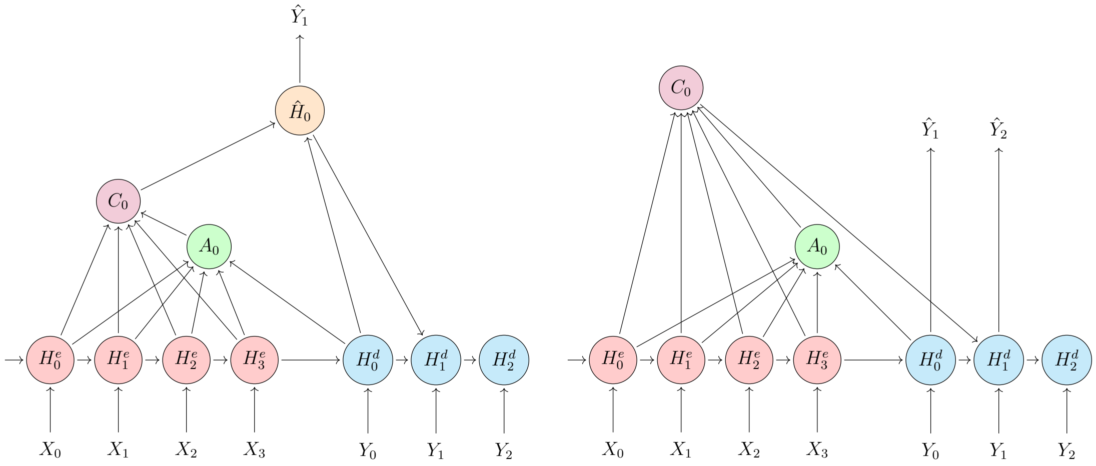

Figure 2.12: Luong vs. Bahdanau Attention. The difference between two attention models is how the output of the decoder network is calculated.

Self-Attention

Attention mechanisms in deep learning aid the interpretability of the model by showing how much focus is given to a certain input and output (Bahdanau et al., 2014; Luong et al., 2015). Self-attention is another method of calculating attention where focus is given only to the input (Vaswani et al., 2017). The attention is calculated by using three vectors: query (Q), key (K) and value (V) which are randomised initially. The equation is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}) * V \tag{2.17}$$

The output of the component is a matrix of scores with the shape Nxm, where N is the number of samples and m is the number of features d_k is a hyperparameter. The component is visualised in Figure 2.13.

A more general case of self-attention is the multi-headed self-attention, where a different number of scaled dot-product attention components are executed in parallel and concatenated afterwards. The application of multi-headed self-attention enables the mechanism to attend to different representation subspaces

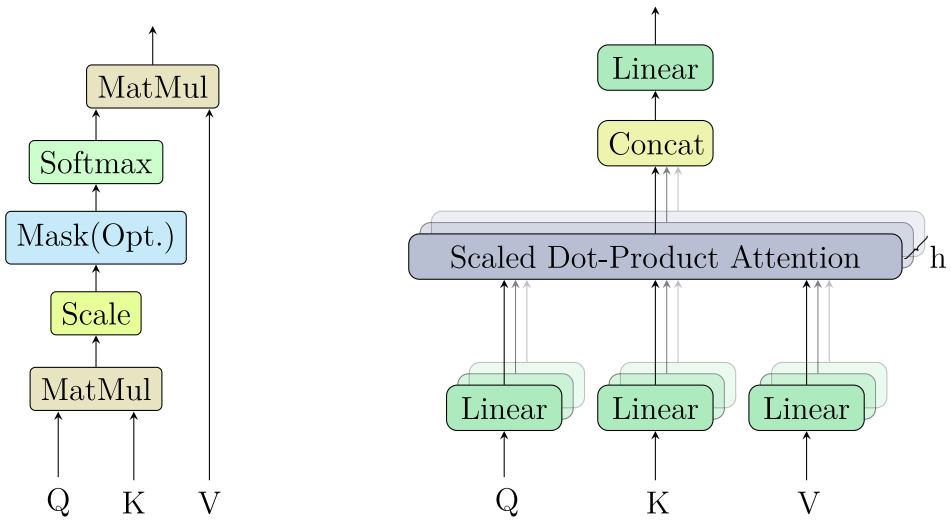

Figure 2.13: Self-attention and multi-head self-attention

simultaneously (Vaswani et al., 2017). Multi-head attention is calculated by Equation 2.18.

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, ..., head_n)W^O \\ where\ head_i &= Attention(QW_i^Q,\ KW_i^K,\ VW_i^V) \end{aligned} \tag{2.18}$$

2.5.5 Transformer Networks

Developed by Vaswani et al. (2017), the transformer network models sequences using dense layers and self-attention. The architecture uses an encoder-decoder approach like the previous methods, where the input is passed through multiple encoders, and the last encoder's output is given as input to all the decoders (Vaswani et al., 2017). The self-attention component removes the requirement of the unfolding of sequences during training, accelerating the process. However, during the testing phase, predictions are made at each timestep, similar to RNNs. The architecture of the transformer is visualised in Figure 2.14.

The encoder part of the model includes the multi-head self-attention layer, a neural network, skip connections and add & normalisation layers (Vaswani et al., 2017). The decoder contains a masked multi-head self-attention layer, a multi-head attention layer where the output of the encoder is received and a linear layer. Addition & normalisation layers and skip connections are included between components. The multi-head attention component receives the query vector from the previous layer and the last encoder's output. The first decoder receives input from the embedded output variable(s) and the last encoder's output.

Initially, the transformer architecture was used for NLP tasks; with increasing popularity, it has been used for tasks such as finance, computer vision and data imputation (Dosovitskiy et al., 2020; Liu et al., 2019a; De Waele et al., 2022). Variations include architectures such as BERT, where only the transformer encoder is used and GPT, where only the transformer decoder is used (Devlin et al., 2018; Radford et al., 2018).

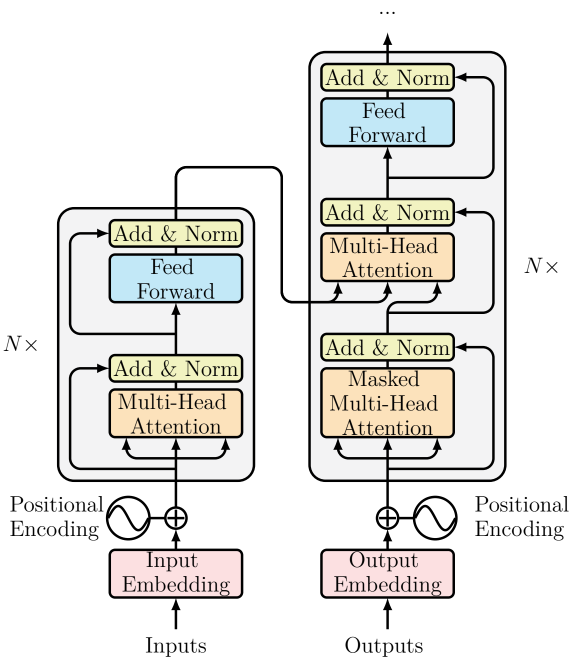

Figure 2.14: Illustration of the transformer architecture (Vaswani et al., 2017)

2.5.6 Variational Auto-encoder

The primary use of an auto-encoder is to reduce the dimensionality of the inputs to capture the essential details of the data. The main drawback of the vanilla auto-encoder is its inability to generalise distributions, thus, it cannot generate new data points accurately. To overcome this problem, VAEs have been proposed (Kingma and Welling, 2013). VAEs address this problem by learning the distribution of the data in lower dimensions instead of the representation. By doing so, the model captures the essential properties of the data.

As observed in Equation 2.19, the posterior distribution $p_\theta(Z|X)$ is intractable due to the integral on the right-hand side. However, an approximation of the posterior distribution, $q_\phi(Z|X)$, can be made using neural networks, which is the encoder part of the network. The likelihood $p_\theta(X|Z)$ can be estimated and is learned during training which is the decoder part of the network. The aim of the network is to satisfy the equation $q_\phi(Z|X) \approx p_\theta(X|Z)$. The VAE makes use of the reparametrisation trick to skip using resources over complex integrals by sampling from a normal distribution of $\mathcal{N}(0,1)$ and scaling it by the standard deviation of the distribution and adding the mean (Kingma and Welling, 2013).

$$p_\theta(Z|X) = \int \frac{p_\theta(X|Z)p(Z)}{p(X)} dZ \tag{2.19}$$

The objective function of the auto-encoder is as follows (Kingma and Welling, 2013):

$$\ln p_\theta(X) = \text{ELBO} + KL(q_\phi(Z|X)||p_\theta(Z|X)) \tag{2.20}$$

$$KL(Q||P) = \int_{\mathbb{R}^d} p(x) \frac{p(x)}{q(x)}\, dx \tag{2.21}$$

where $KL(q_\phi(Z|X)||p_\theta(Z|X))$ is the KL divergence between the encoder and the decoder and Evidence Lower Bound (ELBO) is the evidence lower bound, which solves the issue of intractability by transforming the inference problem to an optimisation one, defined in Equation 2.23. KL divergence measures the distance between two distributions given by Equation 2.21. KL divergence is non-negative therefore resulting in:

$$\ln p_\theta(X) \geq \text{ELBO} \tag{2.22}$$

The optimal parameters of the network should aim to maximise ELBO defined by Equation 2.23, where the first term is the reconstruction loss in log-likelihood and the second term is the KL divergence between the encoder network and the prior distribution $p(Z)$ that forces regularisation.

$$ELBO = E_{q_\phi(Z|X)}[\ln p_\theta(Z|X)] - KL[q_\phi(Z|X)||p(Z)] \tag{2.23}$$

VAEs have been used for tasks such as text classification and text generation (Xu et al., 2017; Semeniuta et al., 2017). VAEs have been used for imputation purposes in the context of traffic and milling circuit data (Boquet et al., 2019; McCoy et al., 2018). Variations of VAEs include denoising VAE and stacked VAE (Vincent et al., 2010, 2008). VAE can be used for imputing MCAR data (Gondara and Wang, 2018).

2.5.7 Generative Adversarial Network

GAN is a type of generative neural network that consists of two components, a generator and a discriminator (Goodfellow et al., 2014). A generator takes in random noise and creates a data sample, and the discriminator takes the generated data as an input and outputs a fake or real label. The discriminator is trained with both real data and data from the generator. The generator is trained with the feedback received from the discriminator. These two components

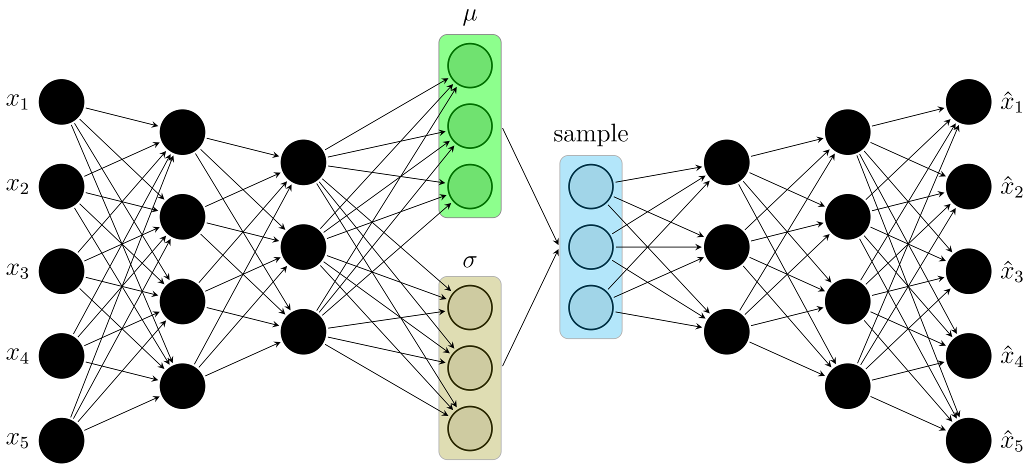

Figure 2.15: Architecture of VAE (Kingma and Welling, 2013). The input is encoded to a lower dimension normal distribution. The decoding process uses the sampled vector from the encoded distribution and creates the reconstruction of the original input.

compete with each other, aiming to reach Nash equilibrium which can be observed with the objective in Equation 2.24 (Goodfellow et al., 2014). The learning that takes place is a supervised one, separate from other generative approaches. GAN is visualised in Figure 2.16.

$$\begin{gathered}\min_G(\max_D E(G,D)) \\ E(G,D) = \frac{1}{2}E_{x \sim p_t}[1 - D(x)] + \frac{1}{2}E_{z \sim p_z}[1 - D(G(z))]\end{gathered} \tag{2.24}$$

where $E(G,D)$ is the function to be minimised according to the function D and maximised according to the function G. $D(G(z))$ is the function applied by the discriminator to the output of the generator where a point z is sampled from a distribution of p_z which is usually random noise, and $D(x)$ is equal to the output of the discriminator of an input x sampled from a distribution of p_t.

GANs have mostly been used for generating images (Zhu et al., 2017; Abdal et al., 2019). The main aim of a GAN is to model the latent space of the data to create new samples. Due to this behaviour, GANs can be used for imputing data (Yoon et al., 2018).

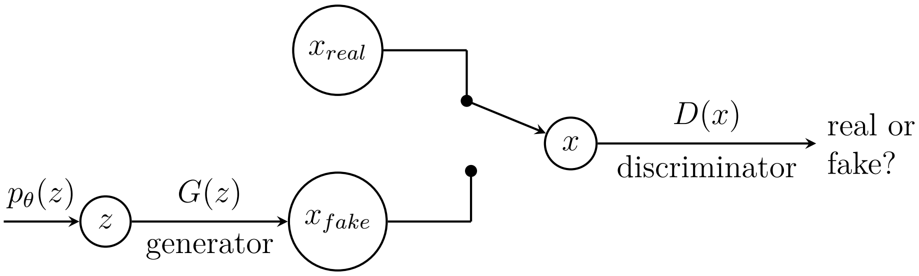

Figure 2.16: A sample GAN. The generator receives random noise as input and generates new data points. The discriminator receives either real data or fake data created by the generator and labels the sample as fake or real.

2.6 Summary

This chapter introduced the background about algal blooms, how they occur, their impacts and factors that lead to their occurrences, and what might the future trends be due to the changes in the climate. Due to the nature of the problem, constant monitoring is necessary to detect these anomalies. During monitoring, certain issues may occur that lead to corrupted data. In the following section (Section 2.2), the solution to this problem, data imputation, is discussed. When a complete dataset is obtained with imputation, algal blooms can be predicted in regions using machine learning and/or deep learning. This section (Section 2.5) included common methods that were used for this task.

Chapter 3

Imputation for Water Quality Data

Partial or incomplete data may be returned from in situ monitoring networks due to external factors such as biofouling, electrical/mechanical failures or refinement of data due to quality assurance procedures, leading to misconstrued statistical analysis of the gathered data.

There are different approaches for addressing the problem of missing data imputation. The simplest solution would be to remove the rows of missing data. However, such a solution might affect the quality of the remaining data depending on the miss percentage and the temporal properties of variables. There are simple methods such as mean/median imputation or constant/zero imputation and multivariate solutions such as Multiple Imputation by Chained Equations (MICE) and regression (Ratolojanahary et al., 2019; Khalifeloo et al., 2015). With the increasing popularity and availability of deep learning and machine learning models, models such as random forests and neural networks are also used for imputation (Stekhoven, 2015; Kim et al., 2017).

This chapter compares various deep learning and machine learning models used for data imputation and proposes a novel architecture that uses a self-attention component in combination with LSTMs to improve the effectiveness and interpretability of data imputation in the context of water quality. The proposed model is compared to different imputation methods; mean imputation, MICE with Bayesian ridge regressor (Rodríguez et al., 2021) and k-NN (Jadhav et al., 2019), GAN (Yoon et al., 2018), VAE (Boquet et al., 2019), RNN (Zhang et al., 2019b). These models were chosen as each makes different assumptions about the data. VAE and Bayesian Ridge assume that the data is normally distributed and model the data with Bayesian probability. GANs aim to learn the latent distribution of data using Nash Equilibrium. k-NN uses distance as a similarity metric for imputation. Recurrent models, the Luong attention model and our approach expose the temporal properties of the data. The proposed model outperforms the baselines in three of the four sites.

Neural network models are black-box processes by default where no information is given about the prediction process due to the sheer number of calculations. This results in the reduction of interpretability of the process. The model is also tested on three other locations with different properties, outlined in Section 3.2.2, to test the generalisability of the model. The model proposed in this chapter performs imputation with a single pass imputing multiple variables.

The self-attention component proposed in this chapter gives insight into how samples interact with each other at different stages of the network, increasing interpretability, as opposed to other neural network models and guides the model to increase its performance. The model performs imputation with multiple numbers of missing variables as opposed to single variable imputation done by other recurrent neural networks. The models were tested with varying missing rates starting from 5% to 95%.

Section 3.2 outlines the details of the models compared in this chapter and introduces the novel architecture. Section 3.3 includes the experimental setting and its results. Section 3.4 compares the proposed model and the previous methods based on results. Section 3.5 provides an overall view of the development process and future research directions.

3.1 Related Work

Imputation of missing data for the domain of water quality has been done in several approaches. Zhang et al. (2019b) use an encoder-decoder LSTM model with attention and sliding window approach for imputation. Zhang and Thorburn (2021) modify the previous model by altering the context vector to include separate weights before and after missing parts of data. The use of sliding windows in both works enables the model to focus on certain parts of the input. The imputation performed by these works imputes completely unobserved datapoints rather than partially observed ones resulting in limited usability. Kim et al. (2015) compare ANN, SOM and a Soil and Water Assessment Tool with data from Taehwa River, South Korea. This study is limited to a single river with low and high water flow and is not tested for open water bodies. This study also shows that simulation tools could be used for imputation under low water flow conditions. Rodríguez et al. (2021) compare inverse distance weighting, RF regressor, Ridge regression, Bayesian Ridge (BR) regression, AdaBoost, Huber regressor, Support Vector Regressor (SVR) and k-NN regressor for data imputation for Santa Lucía Chico River, Uruguay, with missingness between 50% and 70%. Tabari and Hosseinzadeh Talaee (2015) compare SOM and Radial Basis Function (RBF) networks in the context of water quality data imputation. The study area of this work is limited to a single basin which reduces the generalisability of the approaches. The use of non-linear models improves the imputation process as the data can be exploited further. RF and SVD have been applied to the imputation of water quality data for missing rates of 10%, 20%, 30% and 70% (Kim et al., 2019b). SVD being a linear model, results in lower performance as water quality data can be non-linear (Yang and Moyer, 2020). Ratolojanahary et al. (2019) compare RFs, Boosted Regression Trees, k-NN and SVR using water quality data from Oursbelille, France, with an 82% miss rate. Nieh et al. (2014) compare mean, median and multiple imputation in the context of microbial water quality data with 45% and 53% miss rates. Osman et al. (2018) compare Gaussian Process Regression, Principal Component Analysis, Decision Trees, ANN, Multiple Imputation and EM models. Shu et al. (2021) implement a GRU autoencoder to impute river water quality data. Mulia et al. (2015) use SOM with wavelet decomposition to impute water temperature data in Johor Strait, Malaysia. Chivers et al. (2020) use k-NN, RF, SVM and ANN to impute and classify rainfall data in 37 stations around the UK between 0.01% and 50% miss rates. The advantages and disadvantages of the used approaches can be found in Section 2.4.2 (RF), Section 2.4.1 (SVM), Section 2.5.1 (ANN), and Section 2.5.3 (LSTM).

The majority of the mentioned methods for water quality data imputation focus on improving the performance of the model for a single water body. The proposed model achieves better performance at different monitoring locations with different properties. The attention component also provides information between the input elements of the model from start to finish providing a different explanation than most approaches. The testing of the model is done with eleven different values within the range of $[5\%, 95\%]$ miss rates. In previous work, the majority of the models are tested within the range of $[10\%, 70\%]$ miss rates or discrete values such as $20\%, 50\%, 70\%$ miss rates. Our experimentation setting reflects the real-world phenomenon where datasets might have high miss rates and data become unusable.

3.2 Methodology

3.2.1 Problem Definition

Consider a time series consisting of n observations with k features denoted by Equation 3.1 where m_n is the missing data, x_n is the observation and $\backslash$ is the operation of separating the missing data from the observed data.

$$M^T = \{x_1 \backslash m_1, x_2 \backslash m_2, ..., x_n \backslash m_n\} \tag{3.1}$$

The objective is to recover the missing data as accurately as possible with the knowledge of the observed while keeping the observed data unchanged as much as possible. The function applied by the model, $f(x)$, should return the imputed and the reconstructed values for all data points in the time series. The proposed model consists of various deep learning components to find a suitable function for $f(x)$ using training and validation datasets and a test set for model comparison.

3.2.2 The Data

Several datasets were considered for experimentation for this work which are:

1. CEFAS (UK) Dataset: Dataset for tracking phytoplankton activity in the North and Irish Sea

    - Frequency: 20-30 minutes at four locations
    - Date Range: 2002-2019 (Depending on location)

2. Environmental Agency (UK) Dataset: Dataset for tracking nutrients in U.K. inland waters (∼100 nutrient categories)

    - Frequency: Sporadic
    - Date Range: 2002-2019 (Depending on location)

3. National Centers for Coastal Ocean Science (US): Dataset for tracking phytoplankton species and nutrients around the U.S East Coast

    - Frequency: Sporadic
    - Date Range: 2001-2017

4. Finnish Phytoplankton Database: Dataset for tracking phytoplankton species in and around Finland

    - Frequency: At least once a year
    - Date Range: 2000-2019

. The CEFAS dataset was chosen due to its high data quality and high sampling frequency.

The data was collected by ESM2 and ESMx data loggers at four different moorings depicted in Figure 3.1. The data was collected as a part of the National Marine Monitoring Programme (NMMP) to monitor eutrophication regarding the Convention for the Protection of the Marine Environment of the North-East Atlantic (OSPAR) and MSFD assessments. These programmes aim to protect marine life around Europe against issues such as overfishing, excessive amount of nutrients and plastic pollution (Leonardo et al., 2011). The whole dataset was partitioned into four fractions based on location. Each of the datasets is expected to have different characteristics due to their locations, such that the Liverpool

	mean	std	min	max	description	Unit
fluors	1.16	1.77	0.01	42.32	Chlorophyll Fluorescence	arb. unit
ftu	8.52	11.54	0.01	221.22	Turbidity	Formazin Turbidity Unit (FTU)
o2conc	9.19	1.00	5.40	16.04	Dissolved Oxygen	mg/l
sal	33.92	1.13	25.76	35.45	Salinity	PSS78 (Practical Salinity Scale)
temp	11.56	4.33	1.74	21.33	Temperature	°C
depth_0	225.70	384.91	0.00	2566.80	PAR at 0 meter	μE/m^2s
depth_1	69.15	171.46	0.00	1622.70	PAR at 1 meter	μE/m^2s
depth_2	44.47	116.16	0.00	1617.50	PAR at 2 meters	μE/m^2s

Table 3.1: Summary of the dataset variables

buoy is near a maritime route, WestGab is near wind farms, TH1 is near the mouth/delta of Thames, and Dowsing is in the open sea.

Models may be able to expose these spatial differences among the buoys and the temporal properties. The periodicity and the relationship between the variables were analysed by Blauw et al. (2018), Blauw et al. (2012) and Heffernan et al. (2010) with varying date ranges and locations by performing wavelet analysis. The periodicities of variables depend on the season and range between 6 hours to 24 hours.

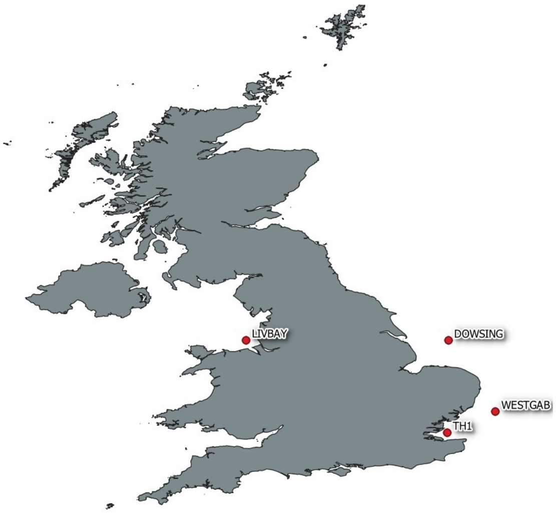

Figure 3.1: Locations of moorings. Each mooring is expected to have different properties based on their location such that the Liverpool buoy is near a maritime route, WestGab is near wind farms, TH1 is near the mouth/delta of The Thames and Dowsing is in the open sea.

Table 3.1 contains a summary of the whole dataset. Chlorophyll fluorescence is caused by algal activity through photosynthesis. Turbidity is the cleanliness of the water. Dissolved oxygen increases with photosynthetic activity and is used for respiration and decomposers. Salinity measures the concentration of salt in water. PAR is the light received by algae that can be used for photosynthetic activity. The data were collected at 30-minute intervals at each station between the dates 01/01/2009 and 04/08/2019. Before normalisation, PAR columns of the data were imputed with zero imputation with regard to the sunset and sunrise time according to the observation date. The operations result in a 54.05% miss rate for TH1, a 65.99% miss rate for LIVBAY, a 56.39% miss rate for DOWSING, and a 56.59% miss rate for WESTGAB. Miss percentages reported relates to rows with at least one value missing.

3.2.3 Missingness Analysis

The figures in this section were obtained using the *missingno* library (Bilogur, 2018). Figure 3.3 shows the number of observed instances per feature showing which feature has the most effect on missingness in a particular dataset. Judging by the counts only, there is no clear pattern of missingness that can be deducted as the value counts and ratios differ for each location.

Figure 3.2 shows the correlation between variables regarding missingness. The values range between 1 and -1, 1 being equal to both variables appearing together, -1 being equal to only one of the two variables appearing and values close to 0 indicates no correlation with respect to missingness. It should be noted that variables with no missing entries do not appear in the figure. The notable values in the heatmaps are the values of (depth_1, depth_2) which could be forced by PAR treatment to the dataset and the (salinity, o2conc) relationship, which could be due to a single device taking both measurements. The rest of the correlations

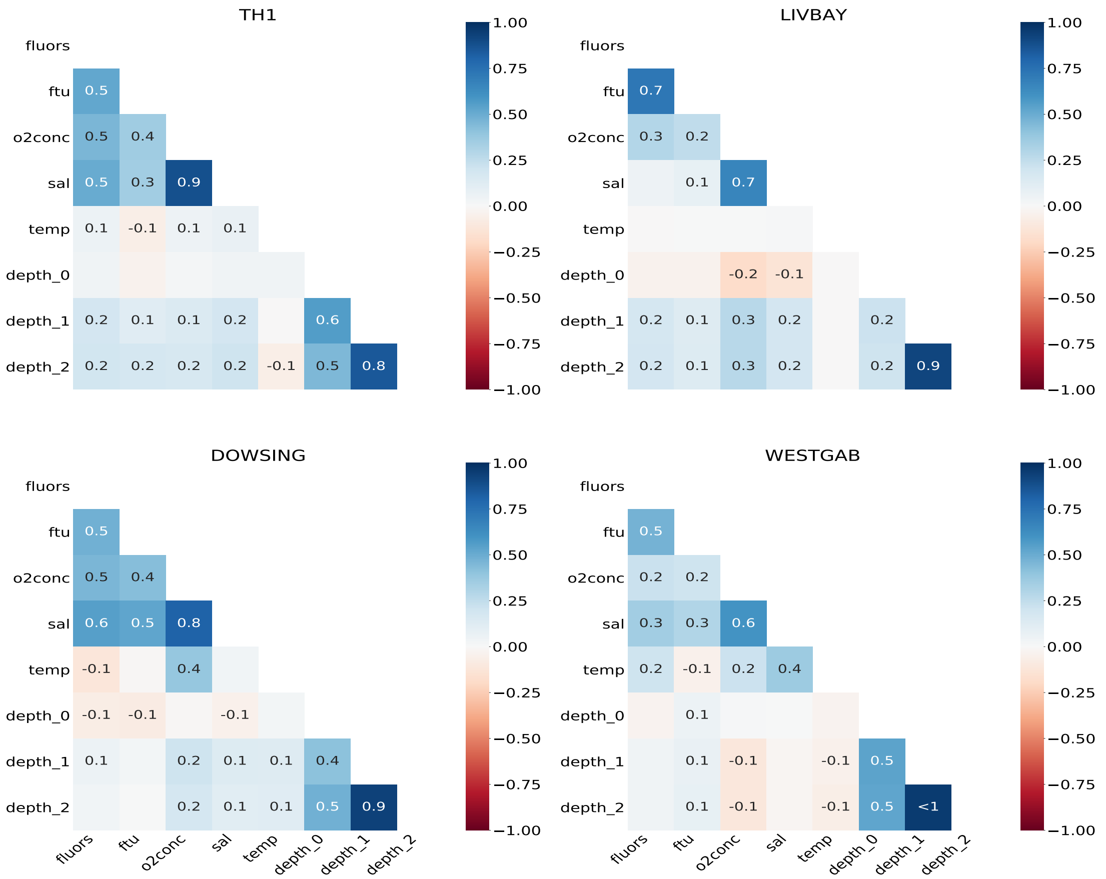

Figure 3.2: Heatmap of missingness of four locations. The values range between 1 and -1, 1 being equal to both variables appearing together, -1 being equal to only one of the two variables appearing and values close to 0 indicates no correlation with respect to missingness.

differ and depend on the mooring location.

Figure 3.4 uses a hierarchical clustering algorithm based on nullity correlation, aiming to minimise distance (Bilogur, 2018). It should be noted that the dendrogram closely correlates with the heatmap as tuples close to 1 or -1, i.e. tuples with no missing variables or missing both variables, have less distance between their clusters. The distance values get smaller as the pattern in missingness becomes more explicit.

The model selection for data imputation was done by omitting values from the complete samples and imputing these omitted samples using the trained methods.

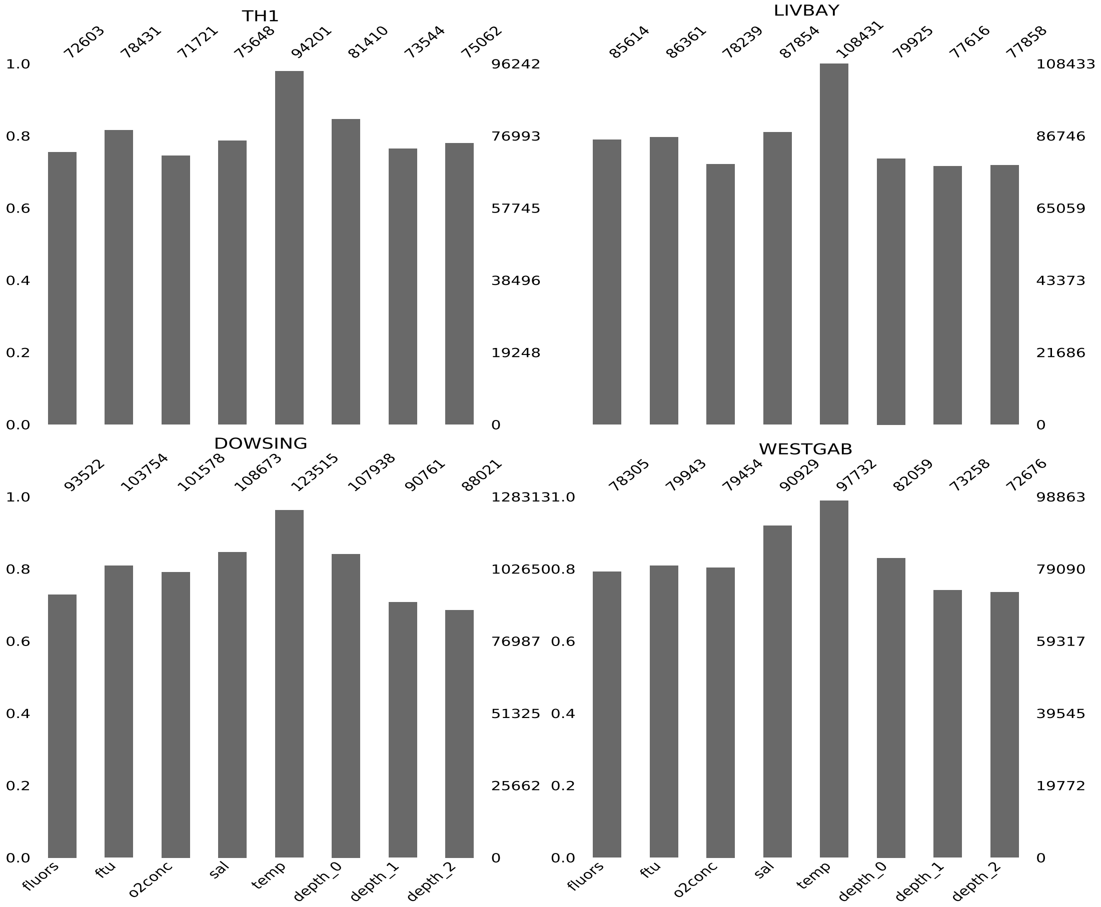

Figure 3.3: Bar plot of number of rows regarding the count of observed instances in four locations

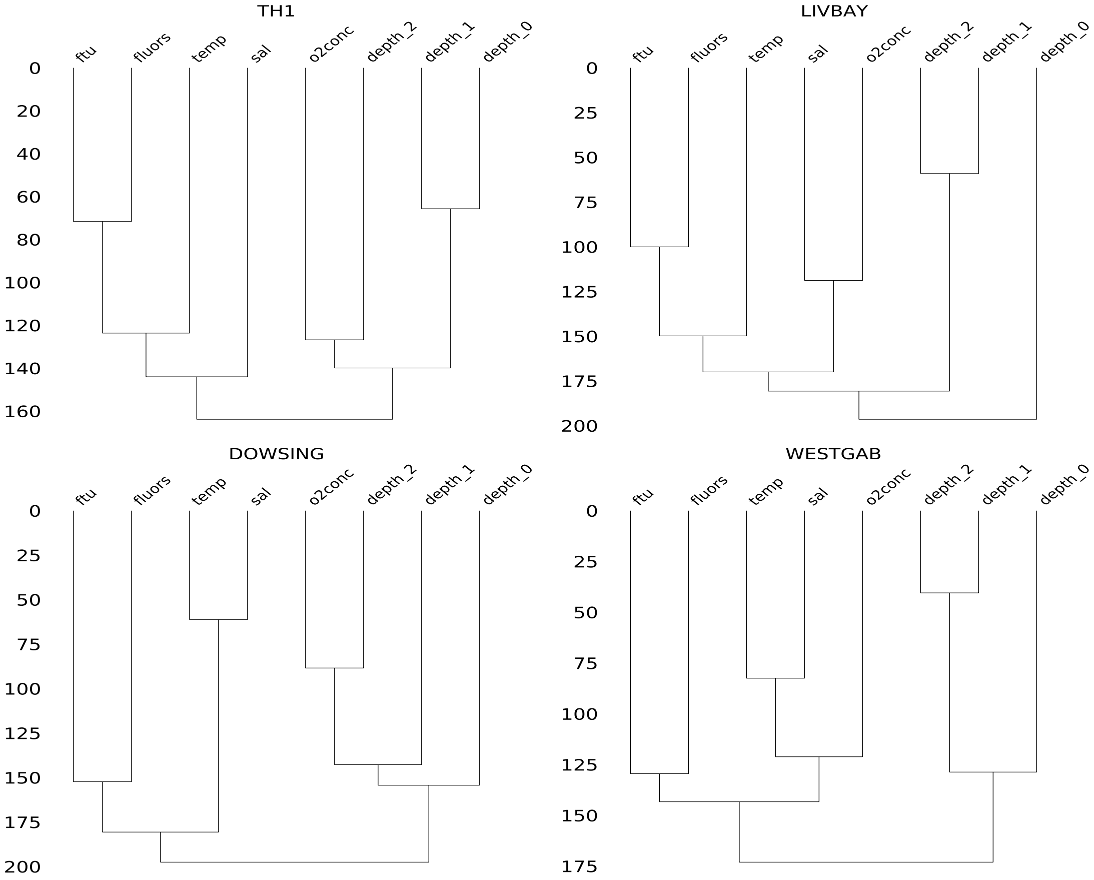

Figure 3.4: Dendrogram of missing features in four locations. The distances are calculated using nullity correlation where closely connected features can be clustered at short distances and appear together, be missing together or one variable might always be present while the other is missing.

The error was calculated using Root Mean Square Error (RMSE). The same error type was used for the prediction stage.

Multiple Imputation using Chained Equations

The weight optimisation task can be applied to data imputation in the form of multiple imputation. This type of imputation starts with an initial value for imputing variables and is trained to converge towards the ground truth using certain heuristics. MICE is a statistical data imputation method that consists of six steps (Azur et al., 2011). These steps are as follows:

1. The data is initially imputed via a simple imputation of choice, such as mean or median imputation as placeholders.
2. The placeholder variables for one feature are set to missing.
3. A regression of choice is done on the missing variable from Step 2.
4. The values from the regression take the place of the missing variable.
5. Steps 2-4 are repeated for other variables that have missing observations.
6. Steps 2-5 are repeated for the whole dataset a number of times or until the change reaches below a threshold value.

Several heuristics can be used to select the variable in Step 2, such as starting from the most missing or least missing. MICE assumes the data is MAR (Azur et al., 2011).

Two different regressors for MICE were used in this chapter: k-NN and BR regressor. k-NN calculates the mean of the nearest k neighbours with a distance metric for each feature and assigns it as the new value. BR regressor assumes

Equation 3.2, where α is a random variable estimated from the data. Output y is assumed to be normally distributed around Xw (Neal, 2012).

$$p(y|X, w, \alpha) = \mathcal{N}(y|Xw, \alpha) \tag{3.2}$$

The BR regressor calculates the weight matrix, w, according to Equation 3.3. The parameters λ and α are estimated using log marginal likelihood (Neal, 2012).

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1} I_p) \tag{3.3}$$

Generative Adversarial Imputation Network

The components of a Generative Adversarial Imputation Network (GAIN) work in the following way: the generator imputes missing data and passes the output to the discriminator, then the discriminator tries to distinguish between the imputed data and the observed data per variable, comparing the output to the mask matrix of the ground truth (Yoon et al., 2018). The hint matrix, H, depends on a mask matrix that is fed into the discriminator to ensure that the generated samples belong to the observed distribution of the data (Yoon et al., 2018). The mask matrix is defined as $M = (M_1, ..., M_d)$ taking values in $\{0, 1\}^d$. Due to the existence of the hint matrix, the function D becomes $D : X \times H \rightarrow [0, 1]^d$, where the i-th element of $D(x, h)$ corresponds to the probability that the i-th component of x was observed on the condition that $X = x$ and $H = h$. The architecture of GAIN is illustrated in Figure 3.5.

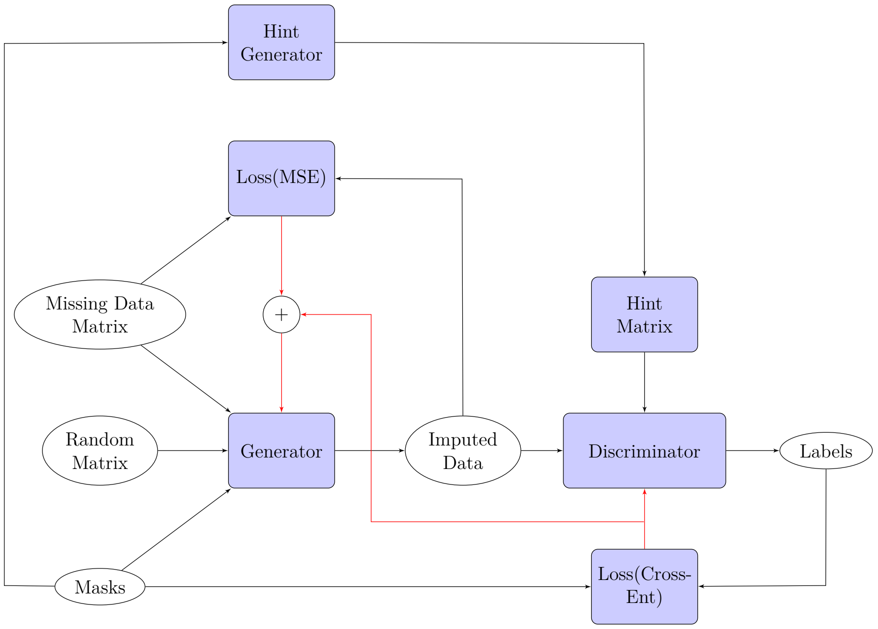

Figure 3.5: Architecture of GAIN (Yoon et al., 2018). The red arrows indicate back propagation. The missing data is given into the generator together with the masks and random noise. The imputed data out of the generator is given to the discriminator which outputs the missingness per variable using an additional hint matrix that contains partial information about the true masks. The generator is trained on both Mean Square Error (MSE) of imputation and reconstruction and the cross-entropy loss of the masks and discriminator output. The discriminator is trained on cross-entropy loss of masks only.

The following equation defines the output behaviour of the generator (Yoon et al., 2018):

$$\begin{aligned}\bar{X} &= G(X, M, (1-M) \odot Z) \\ \hat{X} &= M \odot \bar{X} + (1-M) \odot \bar{X}\end{aligned} \tag{3.4}$$

where Z is random noise, M is the mask matrix, X is the data with missing values, $\bar{X}$ is the imputed data for each variable in X, $\hat{X}$ is the imputed matrix and $\odot$ is element-wise multiplication. Similar to GAN, GAIN's objective function is as follows (Yoon et al., 2018):

$$\begin{gathered}\min_G(\max_D E(G, D)) \\ E(G, D) = E_{\hat{X},M,H}[M^T \log \hat{M} + (1-M)^T \log(1-\hat{M})]\end{gathered} \tag{3.5}$$

where $\hat{M}$ is the output of the discriminator, and $D(\hat{X}, H)$ and $\hat{X}$ are defined in Equation 3.4. The main drawback of this method is that it does not take temporal properties of the data into account. The data is assumed to be MCAR for this model (Yoon et al., 2018).

3.2.4 Proposed Models

Imputation Model

The proposed model, named Self-Attention Imputer (SAI), in Figure 3.6, uses the attention model introduced by Vaswani et al. (2017) with the addition of LSTMs for temporal analysis and a linear layer at the end since a regression task is executed. Similar to Cao et al. (2018), a backward pass through the data is done, but this is executed at the same pass using only the input batch. Instead of using a single self-attention component for a biLSTM layer exposing the periodical information known previously, using separate self-attention components enables the model to give different weights in backward and forward directions. The self-attention component increases the interpretability of the neural network by assigning weights between samples given as input. Moreover, this entails that the relationship between samples might not be linear depending on the missingness of the variables. The model was based on the evidence that water quality data had MAR properties and the statistical analysis of periodicity, which justifies the use of LSTMs for this task (Güler et al., 2002; Blauw et al., 2018, 2012; Heffernan et al., 2010). The pseudocode for the imputation is given in Algorithm 1. The models were tested with varying missing rates ranging from 5% to 95%. Compared baselines with their parameters (Earlystopping with a patience of 20 epochs with 10^{-5} tolerance was used during training for deep learning models):

- mean imputation

- MICE with kNN - k=25
- MICE with Bayesian Ridge regressor - # of iterations = 100, tolerance = 10^{-5}
- VAE - batch size = 32, Adam used as the optimiser with learning rate = 10^{-4}, two linear layers with ReLU activation for encoder and decoder. Hidden size of four for μ and σ.
- GAIN - batch size = 32, Adam used as the optimiser, discriminator learning rate = 10^{-4}, generator learning rate = 10^{-5}, discriminator trained every 5 epochs, discriminator with three linear layers, two with ReLU activations and one with sigmoid, generator with three linear layers, all with ReLU activation.
- GAIN-LSTM - same hyperparameters as GAIN except for the discriminator with an LSTM and a linear layer with sigmoid activation, generator with four LSTMs and a linear layer with ReLU activation.
- Luong attention model - batch size = 32, Adam used as the optimiser with learning rate = 10^{-4}, encoder hidden size = 16, # of encoder/decoder layers = 1, attention type used = general

Regression Model

The prediction model consists of a 1-D convolution layer, a bidirectional LSTM layer and a linear layer similar to Jin et al. (2020), as depicted in Figure 3.7. The data was imputed using the self-attention imputer trained with 60% of missing data from WestGab buoy. The WestGab data was chosen due to the high percentage of non-imputed dissolved oxygen variable. The kernel used for the

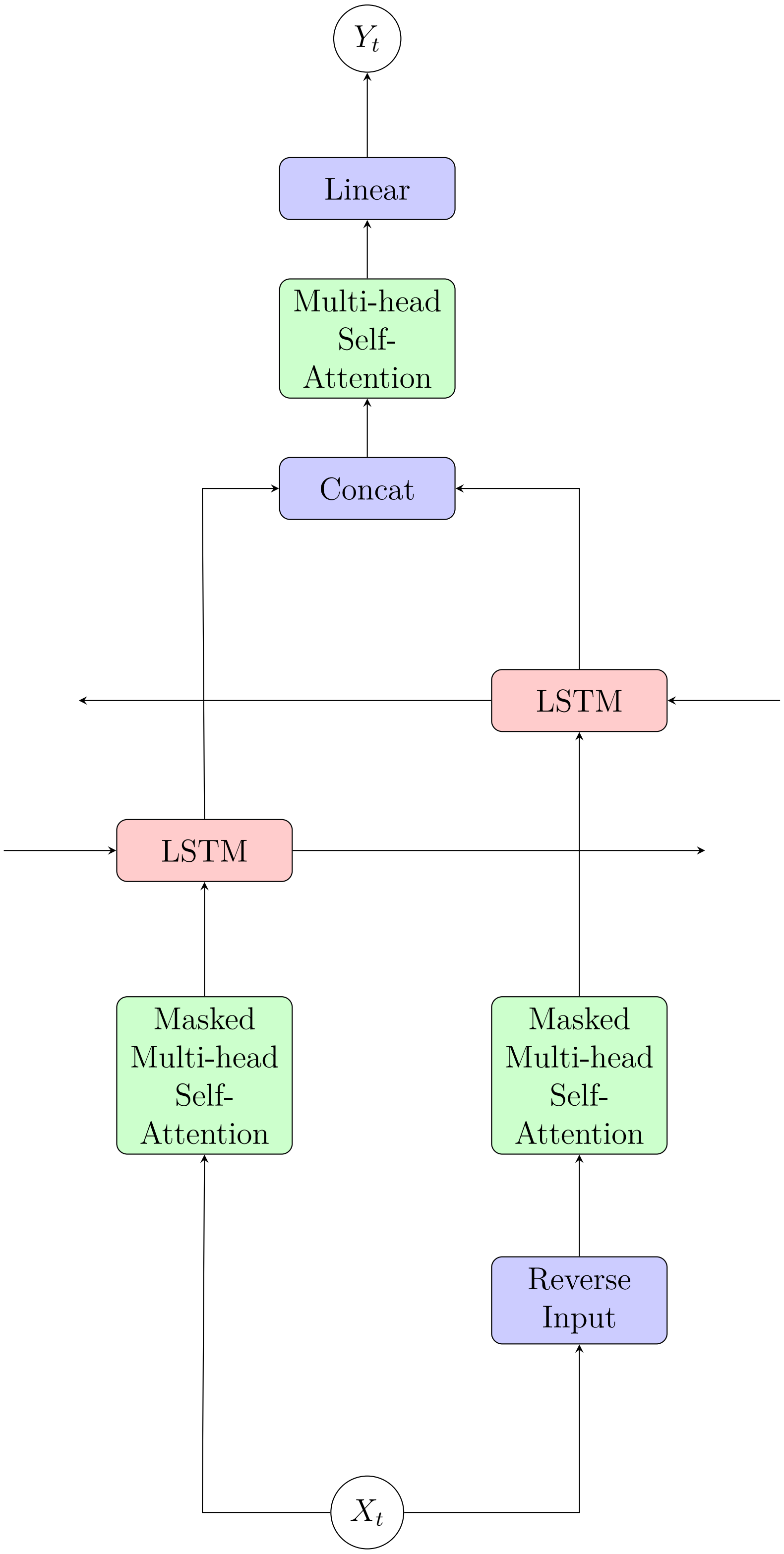
Y_t
Linear
Multi-head Self-Attention
Concat
LSTM
LSTM
Masked Multi-head Self-Attention
Masked Multi-head Self-Attention
Reverse Input
X_t

Figure 3.6: Proposed architecture for imputation. The input passed through masked multi-head attention layers in forward and backward directions resulting in different attention weights for each biLSTM layer direction. The resulting tensors of biLSTM layer are concatenated and fed into a multi-head self-attention and a linear layer respectively. The output is the imputed vector.

Algorithm 1 Imputer training training (single batch)

```
Ensure: $X_{missing}$ = tensor of$(time\_step, batch\_size, 8)$
Ensure: $X_{masks}$ = tensor of$(time\_step, batch\_size, 8)$
  $X_{reverse} \leftarrow X_{missing}.flip()$
  $X_{missing} \leftarrow attention(X_{missing}, X_{masks})$
  $X_{reverse} \leftarrow attention\_reverse(X_{reverse}, X_{masks})$
  $X_{missing} \leftarrow lstm\_forward(X_{missing})$
  $X_{reverse} \leftarrow lstm\_reverse(X_{reverse})$
  $X_{concat} \leftarrow concat(X_{missing}, X_{reverse})$
  $X_{concat} \leftarrow attention\_lstm(X_{concat}, X_{masks})$
  $X_{concat} \leftarrow activation(linear(X_{concat}))$
  $loss \leftarrow mse(X_{real}, X_{concat})$
  $loss.backward()$
```

convolution layer is $2x2$ with a stride of 1. By predicting the dissolved oxygen, anomalies in the phytoplankton behaviour can be detected. The predicted value is normalised dissolved oxygen value.

3.2.5 Additional Experiments

Two additional experiments were run to observe if the design choices that were made previously were harming the imputer performance. The first one is observing if the rationale behind choosing the softplus function was correct, and the other is to observe how using different initial imputation values affect the performance of the model.

Effect of Activation Function

Various functions are applied to the model at this stage. The setting of the model was identical to the initial experimentation, with only the activation function changing. The used activation functions are: ReLU, GeLU, Leaky ReLU and Softplus. The same data for initial experiments were used for training and testing. The results of the experimentation can be seen in Figure 3.9.

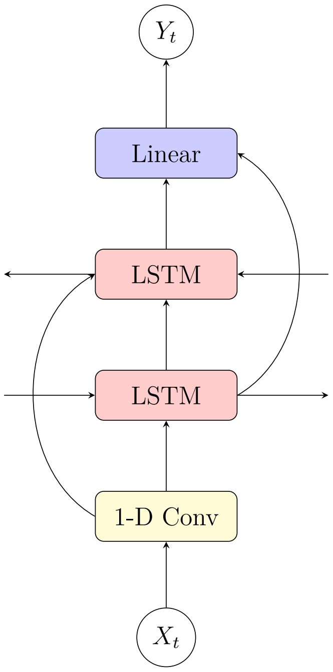

Figure 3.7: Proposed architecture for prediction. The input is passed through a 1-D Convolutional layer, a bi-LSTM layer and a linear layer. The output is a single float variable.

Effect of Initial Imputation Value

Various initial imputation values are applied to the data at this stage. The setting of the model was identical to the initial experimentation, with only the initial imputation value changing. The values used for initial imputation were: -1, -10, -100, 10, 100, and 1. The same data for initial experiments were used for training and testing. The results of the experimentation can be seen in Figure 3.10.

3.3 Results

The complete datapoints were randomly set to missing according to a certain percentage by masking. The data was normalised using min-max normalisation using all the available locations. All the models were trained with 70% of the WestGab

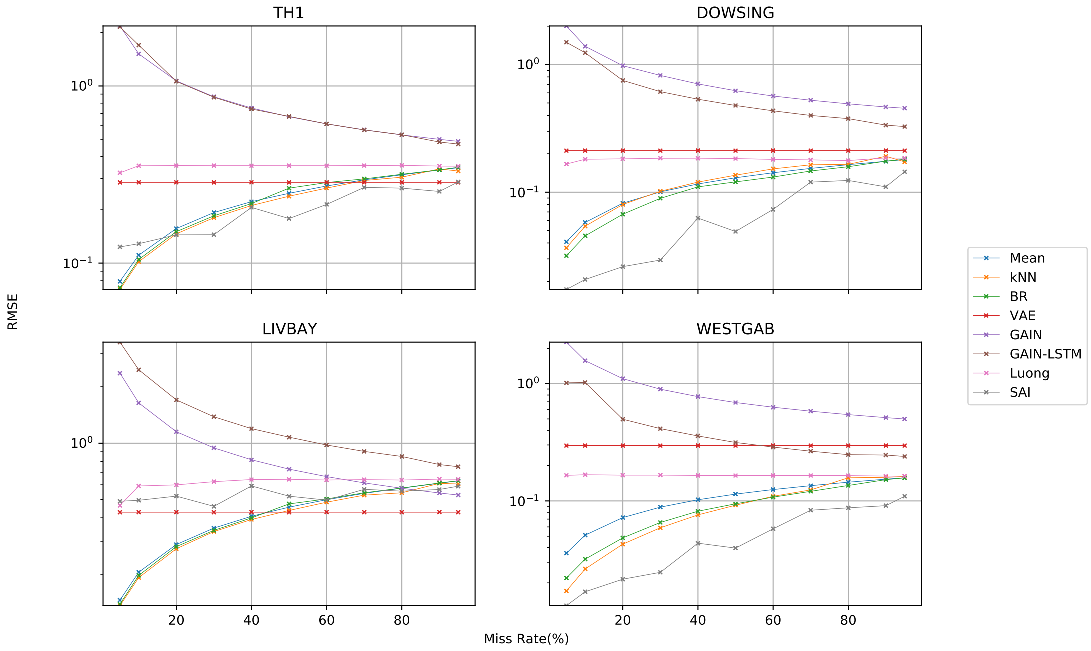

Figure 3.8: Comparison of algorithms for four datasets at different missing percentages

data to observe the imputation performance of the model across datasets with different time ranges and spatial properties while using information only from a single dataset. MSE was used as the loss function where applicable. For prediction, WestGab was chosen due to the low percentage of missingness for the target variable. This means the model would be able to learn the true distribution of the target variable instead of the imputations. Adam optimiser was used for all deep learning models (Kingma and Ba, 2014).

$$z' = \frac{z - min(x)}{max(x) - min(x)} \tag{3.6}$$

Min-max normalisation is defined by Equation 3.6, where z' is the value after normalisation, z is the original data point, $min(x)$ and $max(x)$ is the minimum and maximum values of each feature column, forcing the data to be defined in $[0, 1]$. The missing values were initially imputed with -1 as a placeholder. Using

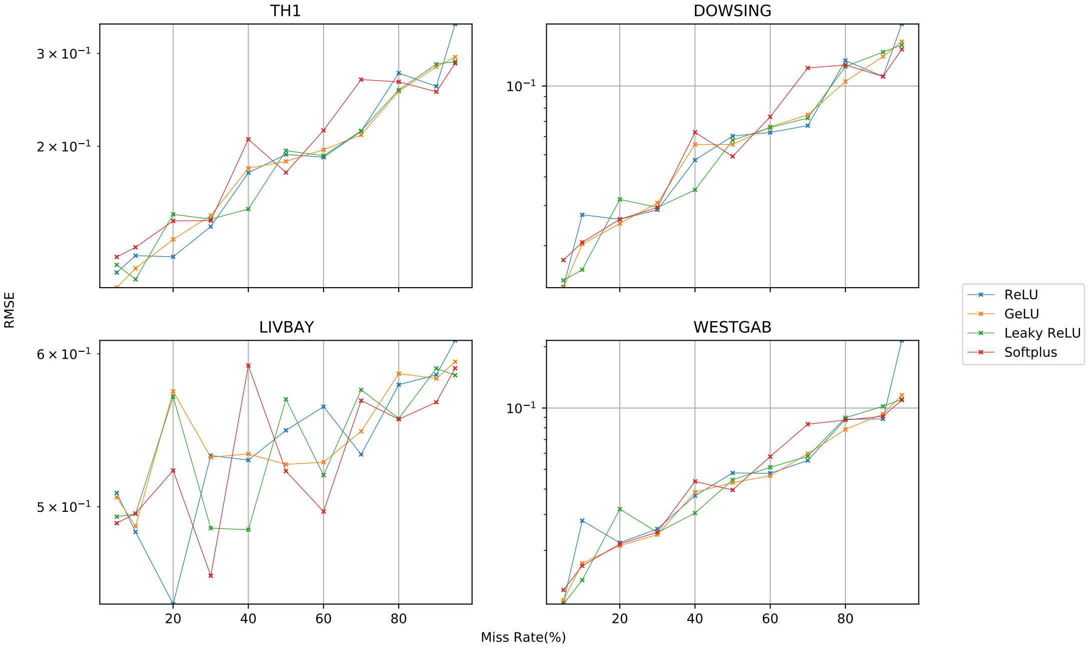

Figure 3.9: Effect of different activation functions for the linear layer on model performance

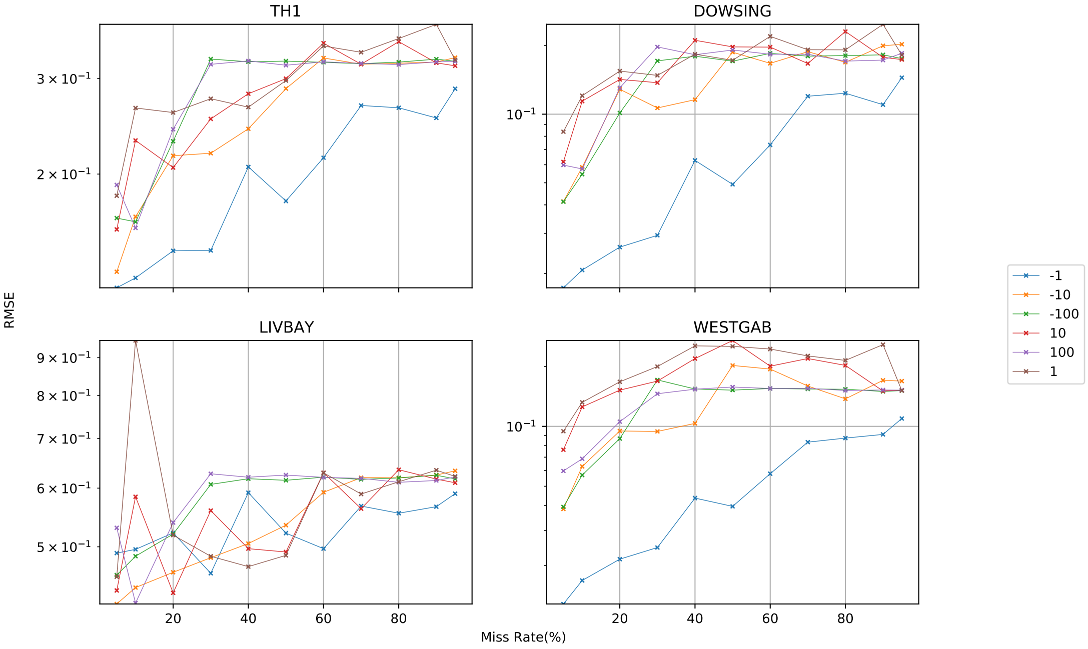

Figure 3.10: Effect of different initial imputation values on model performance

this type of normalisation enables models to use the softplus activation function at the end of linear layers, denoted by Equation 3.7, where *beta* is a hyperparameter. Use of a ReLU was avoided due "dying" since $f(x) = 0, x < 0$.

$$\text{Softplus}(x) = \frac{1}{\beta} * \log(1 + \exp(\beta * x)) \tag{3.7}$$

The models' results were compared using RMSE denoted by Equation 3.8, where y_i' is the value predicted value by the model and y_i^2 is the ground truth. All neural network models were trained with an early stopping criteria of patience 20 and a delta of 10^{-5}. If early stopping was not applied after 300 epochs, training was terminated. The models take each observation as a timestep with a batch size of 32 and 8 features.

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(y_i' - y_i)^2}{n}} \tag{3.8}$$

The GAIN model was tested in two different settings, one with linear layers and another with LSTM layers which included a linear layer at the end, named GAIN and GAIN-LSTM, respectively. A VAE was trained with imputation and reconstruction error without using a missingness matrix simultaneously, contrary to (McCoy et al., 2018; Jun et al., 2019). It should be noted that the neural network models do reconstruction and imputation, whereas MICE and mean only perform imputation. Data with MAR properties assumes that the missing values can be imputed with the observed variables, so the reconstruction loss of the overall network has to be taken into account for deep learning models, whereas for MICE and mean imputation no such assumption is necessary since they do not modify observed variables.

Figure 3.8 visualises experimentation results where the proposed model outperforms the other models after 40% of missing data in at least two of the datasets.

Table 3.2 refers to the prediction task of dissolved oxygen in four datasets after the missing data was imputed. The reconstructed values by the self-attention imputer were replaced with original values before training for the prediction task.

	Error(RMSE)			
	TH1	DOWSING	LIVBAY	WESTGAB
Conv-LSTM	0.0840	0.0806	0.1289	0.0740

Table 3.2: RMSE of prediction for all datasets

3.4 Discussion

The GAIN algorithm is used for imputing MCAR data (Yoon et al., 2018). RMSE of MCAR and GAIN-LSTM show that water quality data is not MCAR due to the model's performance on the supplied locations. Exposing the temporal properties of the data by using GANs under the assumption of the MCAR mechanism does not aid the imputation performance except WestGab and Dowsing. The data was assumed to be MAR, as seen from the other models, given more evidence of the data, i.e. lower miss rates, RMSE always decreases. The poor performance of the GAIN imputer at low rates of missing values shows that the model is not fit for reconstruction purposes. The possible reasons behind the GAIN model's behaviour might be non-convergence, mode collapse or diminishing gradient of the generator. The differences between the models come from the limits to understand the data with the lowest amount of evidence. At lower miss rates, the models apart from VAE and GAIN perform better since non-imputed data is abundant, and the model is able to model the missingness.

Different Bayesian approaches were applied with VAE and BR. Both models map the distributions of data to a Gaussian distribution; however, VAE maps it to a lower distribution by encoding to a lower dimension, sampling from this

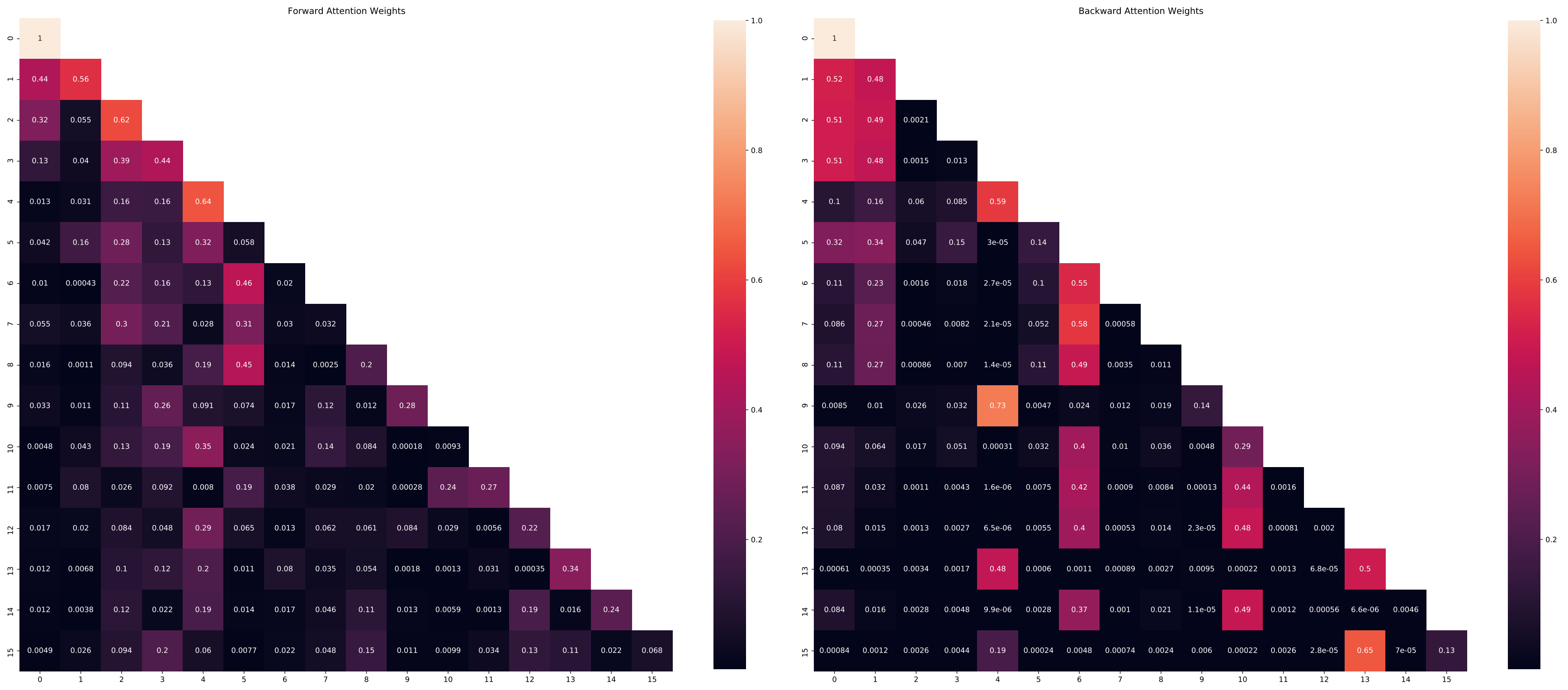

Figure 3.11: Sample heatmap of self-attention scores for a sequence length of 16. The higher scores indicate more attention to that part of the input. The component produces different scores in backward and forward directions.

distribution and decoding to the original distribution. For this task, VAE maps the data distribution together with the missingness, whereas BR assumes that the data and its parameters are normally distributed in its original space and does no reconstruction. It should be noted that the VAE model shows signs of underfitting as the training is terminated after 19 epochs for all rates of missingness and did not improve under early stopping limits. Therefore, VAE was not considered a suitable model for imputation as it is clear that it does not learn from the data. SAI focuses on the important sections of the input instead of modelling the latent distribution of the samples as a whole; therefore, it is less prone to underfitting and does not encode the data to latent dimensions.

The scope of the dataset for experimentation has high percentages (>50%) of missing data in all of the datasets, even after data treatment. The proposed model is aimed to focus on a higher percentage of missing data. Previous work (Blauw et al., 2018, 2012) has shown that there are semi-daily and daily cycles; in spite of skips in the training data, the proposed model, SAI, is able to impute the data effectively regardless of miss rates in the majority of the locations.

Using a different attention mechanism benefits the performance of the model. Luong attention focuses on the relationship between input and output, whereas the proposed model uses a mechanism of Vaswani et al. (2017) which shifts the focus solely to the input of the component. Figure 3.11 visualises the attention mechanism used before the two LSTMs. The multi-head attention component uses ReLU as an activation function, resulting in weights with ≥ 0 where no attention is paid to components with 0 weights. This also shows that the bidirectionality of the model helps it focus on different aspects of the data in different directions and forces the focus on key components of the data. The attention mechanism used by Luong focuses on all of the encoder hidden states and the current decoder hidden state. The self-attention component focuses only on the input, whereas Luong attention focuses on the relationship between the input and the output. Application of different neural network architectures results in different RMSE values such that Luong's RMSE has less deviation depending on the dataset.

The k-NN model shows that the data points show similar properties at low missing percentages, as seen from Figure 3.8, since the model uses nearest neighbours where feature X is not missing. As the ratio of complete datapoints are decreasing, the performance drops drastically to 0.12-0.16 between 70-95% miss rates for WestGab and to 0.52-0.61 for LivBay between the same miss rates for the k-NN model. The high missing % of the problem makes the k-NN model unsuitable for this task compared to SAI. For lower missing percentages (<%40), the neural network models have to shift the focus to reconstruction rather than imputation; still, the model is able to do both tasks in the majority of the cases presented. Since MICE and mean models do not need to do reconstruction, as information is removed from the data, RMSE increases.

The overall performance of SAI gives insight into the missingness properties of these locations. The missingness mechanism of Dowsing and WestGab are similar as the RMSE values of SAI, k-NN, and BR show the same pattern. The missingness pattern of LivBay differs from the other three sites since each tested model had higher RMSE rates for that specific location.

The prediction model was trained and tested on both imputed and non-imputed data. From the results in Figure 3.2, it can be deduced that the imputation model is able to generalise the different distributions to an extent, as the highest RMSE was attained by LivBay data with an RMSE of 0.1289. It shows that the locations have different distributions relating to the dissolved oxygen concentration.

The test for different activation functions shows that there is no best option for the activation function for all cases. For the 50% miss rate, using Softplus might be feasible as the error observed is lower for all datasets. For lower miss rates, alternative functions such as ReLu might be feasible.

Neural networks learn best with smaller values. From Figure 3.10, it can be deduced that using a small out of sample value is beneficial to the learning process of the model in the majority of the cases. LivBay is the outlier dataset in all experiments, which entails a difference in distribution for data in that location.

3.5 Conclusion

This chapter compared various machine learning and deep learning methods for the task of data imputation in the context of water quality data. Introducing a different architecture and attention mechanism improves imputation performance where data is missing above 50%. The attention mechanism increases the interpretability of the model at different stages, aiding data understanding.

The additional experiments investigated the choices of certain hyperparameters during training: the activation function at the final layer of the network and the initial imputation value. It was found that using -1 as the initial imputation value yielded the best results as the data was normalized in a $[0, 1]$ range, and the provided value was out of distribution. The lower and closer values to 0 aided the network in learning more efficiently. The role of the final activation function was also investigated. It was found that the optimal function depends on the site and the miss rate.

Future research directions include usage of different loss functions to reduce the effect of reconstruction loss on the model and broader experimentation on well-known datasets to test the generalisability of the architecture. Ensembles of neural network architectures could be applied together to minimize the effect of reconstruction loss. Transfer learning techniques could be applied to improve the prediction of dissolved oxygen and the imputation. The generative approaches, particularly the GAIN model, could be explored further to observe if the behaviours of the models are due to the missingness properties or hyperparameter choices.

The data used in this chapter was obtained through in situ measurements which are highly frequent. Other forms of data, such as ship-based data, obtain measurements less frequently. The proposed model could be tested on such data in the future.

Chapter 4

Algal Bloom Prediction with Time Series

HABs occur when the population of phytoplankton increases rapidly, causing environmental changes such as sunlight blocking and oxygen depletion (Kahru and Mitchell, 2008). These changes affect the ecosystem and public health since the consumption of aquatic life affected by these blooms poses a health risk (Falconer et al., 1994). In some cases, HABs occur due to eutrophication caused by nutrient overload. The occurrence of eutrophication involves the creation of oxygen deprived zones due to the extreme number of deceased plants and animals, resulting in dead zones with no ability to support life and requiring external action to restore the habitat (Chislock et al., 2013).

With the increasing temperatures due to climate change, it is expected that the frequency of algal blooms is expected to increase and will be seen in new regions (Wells et al., 2015). In addition to the ecological impacts, the occurrence of algal blooms has negative economic impacts. These include cost increases in drinking water treatment and the preservation of biodiversity (Dodds et al., 2009). Regions where these blooms are frequent see lower sales in sectors related to tourism and lower income from fisheries (Bechard, 2020; Karlson et al., 2021).

To prevent this phenomenon from occurring, preventive measures could be taken, which include early detection models that benefit from in-situ data and harness the power of machine learning.

In this chapter, a new model is proposed that improves the detection of outlier activities in certain locations of the North Sea and the Irish Sea using in-situ data and a flexible labelling method with varying ranges of detection and a longer range of time which was not taken into account in the majority of the approaches, with transformer networks and convolution operations. Our approach generates a possible sequence at day $x + i$, i ranging from 1 to 7, using observations at day x with a representation learning approach and filtering the necessary parts of the generated sequence to predict a label. In addition, the reasoning behind the predictions is explained using Shapley Additive Explanations (SHAP) to aid experts in understanding the effects of observations. The scope of this chapter aims to detect the beginning of these blooms due to the mechanics of the phenomenon. It has been observed that using a representation learning approach results in a better model, performing 5% better on average than the baselines.

Section 4.2 outlines the details of the models compared in this chapter and introduces the novel architecture for outlying behaviour detection and the explanation model for observing the effect of the input on the output. Section 4.3 includes the experimental setting and its results. Section 4.4 compares the proposed model and the baselines.

4.1 Related Work

The majority of approaches apply thresholding to categorise labels and forecast future behaviour or apply regression to the problem of HAB detection using dissolved oxygen or chl-a as the target variable, both of which increase with higher photosynthetic activity from algae, as chlorophyll-a is used to capture sunlight and carry out photosynthesis to produce oxygen and glucose. The chl-a concentration increases during an algal bloom due to increased photosynthetic activity, whereas the oxygen concentration increases initially with high photosynthetic activity and drops afterwards due to the increasing decomposer population. It should be noted that the behaviour of inland waters and sea water differ as sea water bodies can act like large reservoirs, so they are less susceptible to change.

The most common approaches lean towards using RFs, SVMs and ANNs to predict algal blooms, which are explained in Section 2.4.2, Section 2.4.1 and Section 2.5.1. A small number of models make use of XGBoost, which is explained in Section 2.4.4. Park et al. (2015) use ANNs and SVMs to predict chl-a concentration in Juam and Yeongsan Reservoir, South Korea, 7 days ahead. Derot et al. (2020) use RF and k-NN over a 34-year long time series to predict four different types of cyanobacteria and their densities in Lake Geneva, Switzerland. Park et al. (2021) apply SVMs and ANNs to detect blooms in Changnyung-Haman Reservoir, South Korea. Yajima and Derot (2018) use RF to predict the chl-a concentration in Urayama Reservoir and Lake Shinji, Japan. Yang et al. (2020) use sensory data to predict HABs using AdaBoost with SVM and RF in Yuyuantan Lake, China. Chen et al. (2015b) use an ARIMA model to predict the chl-a concentration in Lake Taihu and Meiliang Bay, China, comparing it with a multivariate linear regression model. Jiang et al. (2016) use a Continuous Hidden Markov Model with adaptive exponential weighting and Principal Component

Analysis to predict the toxins produced by the algae during blooms. Hidden Markov Models use high amounts of memory and compute time during optimisation, making them unsuitable as the model becomes larger. Hidden Markov Models assume that the state at time t is dependent only on the state at $t-1$, which may not be the case for several problems. RNNs can model a state at time t using information from all previous states. Principal Component Analysis is only able to model linear relationships, making it unsuitable for certain scenarios, as it uses the covariance matrix to generate components. Li et al. (2014) compare ANNs, regression networks and SVMs in the context of predicting chl-a values 7 or 14 days ahead for Tolo Harbour, Hong Kong. Shin et al. (2020) compare SVRs, RFs, XGBoost and LSTMs to predict the chl-a concentrations in Nakdong River, South Korea. This work shows that in short-term predictions, ensemble models such as XGBoost perform better than other deep learning and machine learning models. Lui et al. (2007) use autoregressive models to predict chl-a in a 2-hour and daily period in Crooked Island, Hong Kong. This type of model is easy to implement but only limited to polynomial relationships. Cannizzaro et al. (2009) use ship-based data along the shore of West Florida with supplementary satellite data to aid interpretation, using thresholding to classify blooming. The data is collected between 2000-2006 with 13 different cruises, each lasting between 2-5 days. Mellios et al. (2020) use ML methods such as RFs, k-NNs, and SVMs and correlation coefficients to predict HABs in the lakes in North Europe. The thresholding method used in this work is based on cyanobacterial biomass and categorised into three. Ye et al. (2014) apply hybrid evolutionary algorithms to detect blooms in real-time in Xiangxi Bay Reservoir, China, to predict blooms 1-7 days ahead. The main drawback of evolutionary algorithms is that the generated solution may not be optimal and is heavily restricted to the search space, limiting the usability of the model. Park et al. (2022) use XGBoost to predict HABs in

Geum River, South Korea. In this work, SHAP is also used for filtering features and creating a better model.

Cho et al. (2014) use ANNs combined with correlation and feature selection to predict the dissolved oxygen value in Lake Juam, South Korea. Muttil and Chau (2006) use ANNs and Genetic Programming to detect HABs in Tolo Harbor, Hong Kong, predicting chl-a concentration. Guo et al. (2020) predict sea surface temperature and salinity to detect outlier events in Tolo Harbor, Hong Kong, using ANNs. Yim et al. (2020) use auto-encoders to detect the chl-a levels and the cyanobacteria cell counts using hyperspectral data in Baekje Reservoir, South Korea. Xiao et al. (2017) combine wavelet analysis with neural networks to predict the cyanobacteria density 1-day ahead in Silang Reservoir, China and Lake Winnebago, USA. Using wavelet analysis aids the model in exploiting the temporal properties of the data, aiding performance. Guallar et al. (2016) predict populations of two bloom-forming microalgae using ANNs with a long-term time series (1990-2015) in Alfacs Bay, Spain. Qin et al. (2017) combine ARIMA and Deep Belief Networks to predict red tide biomass in Zhousan and Wenzhou Coastal Area, China. Shamshirband et al. (2019) predict the chl-a value 1 to 3 days ahead, using a combination of an ensemble of ANN with Discrete Wavelet Transform. Using Discrete Wavelet Transform enables analysis of temporal properties of the data. Yi et al. (2018) use Extreme Learning Machines to predict chl-a values 7 days ahead along several weirs on the Nakdong River, South Korea. Extreme Learning Machines were developed as an alternative to backpropagation as the latter requires high amounts of compute time. However, with the developments in and use of GPUs in training deep learning models, Extreme Learning Machines have become outdated. Zhang et al. (2016) use stacked Restricted Boltzmann Machines at the East China Sea coast to predict algal cell density.

These models are not favoured due to the computational complexity during training. Wang et al. (2020) utilise recursive Deep Boltzmann Machines (DBMs) using algal density as the target variable. Huynh et al. (2022) use self-attention and GANs to predict algal blooms in Karlsruhe, Germany. Information about GANs can be found in Section 2.5.7.

Various types of recurrent neural networks have been used for algal bloom detection. For details about the models, see Sections 2.5.3. Lee and Lee (2018) use LSTMs for predicting chl-a values in four rivers of South Korea. Cho and Park (2019) predict the chl-a concentration using Merged LSTMs in Geum River, South Korea. Shan et al. (2022) implement an XGBoost-LSTM approach to detect algal blooms using in-situ data in Three Gorges Reservoir, China. They use the XGBoost model as a feature selector for the LSTM. The feature selection is also tested with SVMs and ANNs. Yu et al. (2020) predict chl-a concentration in Dianchi Lake, China using Wavelet Analysis and LSTM. Wang and Xu (2020) use temporal attention combined with LSTM to predict the chl-a value at most 12 hours ahead in Fujian, China. Cho et al. (2018) use sensory data to predict the chl-a in certain locations in South Korea with LSTMs. They aimed to predict the chl-a concentration a day ahead and 4 days ahead using this approach. Chen et al. (2021) utilise CNNs with attention to detect HABs in Jiulong River, China. Shin et al. (2019) use LSTMs to detect HABs in South Korean Peninsula using the data between 1998 and 2018. Zheng et al. (2021) implement an LSTM-based approach to detect HABs along the BeiYun River, China. Kim et al. (2022) apply attention with two different levels: time and feature level, in combination with LSTMs to detect HABs in Nakdong River, South Korea.

The locations studied in this chapter differ from the majority since most of the focus is divided between Southeast Asia and the United States, whereas our study area is the North and Irish Sea (Sebastiá-Frasquet et al., 2020; Wang et al., 2022). The increased frequency of blooms results in more focus on these areas (Gu et al., 2021; Anderson et al., 2021). Most of the approaches use models like SVMs, RFs or use LSTMs to analyse the long/short term temporal patterns in the data. The approaches that classify the blooms use static values or expert knowledge to classify the responses, as in the cases of Mellios et al. (2020) and Yang et al. (2020). Our approach takes the context of the measurements into account as factors such as temperature affect cellular activity and oxygen solubility in water (Lepock, 2005). The detection time spans of the current approaches are usually short, ranging from 12 hours to 4 days. The proposed model predicts anomalous activity in monitored locations ranging from 1 day ahead to 7 days ahead, using only data from a single day, with a flexible labelling approach. Explanation models provide insight into how the input influences the model's output.

4.1.1 Challenges

Modelling algal blooms has several challenges. Algal blooms are extreme events; therefore, positive labelled samples are extremely low (3-5%) in the dataset, which needs to be addressed during training with methods such as SMOTE or label weighting and model evaluation with a weighted F1 score. Deep learning models require vast amounts of data for training which is solved with continuous and frequent monitoring. Algal blooms are inherently complex as the underlying mechanism is influenced by many factors such as nutrient intake of nitrates and phosphates through industrial pollutants or fertilisers, the water temperature and available light.

4.1.2 Model Explainability

Deep learning models are complex structures with parameters ranging from thousands to billions. A user cannot comprehend the decision-making structure of a neural network; therefore, the interpretability of a model is essential to understand how a model works. Interpretability in a machine learning context can be defined as the extraction of relevant information about a model's prediction mechanism that can be understood by end users. Interpretability is essential to eliminate bias, debug, and provide trustworthiness and information to the end user. Explainability can be model-specific or model-agnostic. Model-specific approaches are naturally interpretable such as linear regression, logistic regression and decision trees (Adadi and Berrada, 2018). Model agnostic explainability can be divided into two categories; local model agnostic and global model agnostic (Molnar, 2020). Model agnostic explanation separates the explanation from the prediction model. Examples include Partial Dependency Plots (Friedman, 2001) and Accumulated Local Effects (Apley and Zhu, 2020) for global model agnostic methods and Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) for local model agnostic methods. Therefore, this type of model is usable for any type of learning model and makes the comparison of explanations across different learning approaches easier (Molnar, 2020). The levels of explanation can also differ, such as input-output, layer and neuron explanation.

Example-Based Explainers Example-based methods provide explanations by selecting specific instances in a dataset to explain model behaviour (Molnar, 2020). In contrast to model agnostic methods, example-based explainers do not explain summaries of features (Molnar, 2020). These methods work well for images as the data given as input needs to be represented in an understandable way, such as images or text (Molnar, 2020). Examples of such methods include Counterfactual Explanations and Adversarial Examples (Molnar, 2020). Counterfactual explanations work by creating hypothetical conditions that are not observed to explain events such that the effect of the change of feature values is observed on the label. Used domains include loan models where small changes can affect the model's outcome. Adversarial examples use perturbations such as noise to deceive the model into observing changes in the output. This type of explanation model is used in image classifiers where the distance between the original and perturbed sample is kept to a minimum to observe class boundaries (Molnar, 2020).

4.1.3 Local Interpretable Model-Agnostic Explanations

The problem of interpreting the outputs of a model is a challenge in the field of deep learning. Interpretation enables the users to understand the internal mechanism of the model by removing the "black box" properties of the models. A method for explaining deep learning classifier model behaviour is LIME (Ribeiro et al., 2016). This type of interpretation is model agnostic, so it can be applied to any model and the explanations are done locally by observing the changes on features relative to the output from the model. The explanation is done by the objective function below (Ribeiro et al., 2016):

$$\varepsilon(x) = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \omega(g) \tag{4.1}$$

where $\varepsilon(x)$ is the explanation of the model, f is the explained model, g is the explanation model, π_x is used to define the locality of the sample, $\omega(g)$ is the measure of complexity of g and $\mathcal{G}$ is all the possible interpretation models. This process aims to find the optimal explanation model that maximises the local behaviour while keeping the complexity of the model as simple as possible. The disadvantage of LIME is that only linear models are used to explain behaviour, and the explanations are only done locally, which may be different from the global structure of the data. The results of the explanation are sample-based such that the influence of each feature is shown on the decision of the label for the specific sample.

LIME has been used for tasks such as detection of antisocial behaviour from tweets, predictive maintenance and natural disaster response (Zinovyeva et al., 2020; Usuga-Cadavid et al., 2021; Gao and Wang, 2022). However, no work about algal bloom explainability using LIME has been done.

4.1.4 SHapley Additive exPlanations

Another method for explaining deep learning classifier model behaviour is SHAP. It is a method that uses game theory and locality for the interpretability of deep learning models (Lundberg and Lee, 2017). The model removes a specific feature f from the input, sampling it from a baseline provided by the user. The model compares the difference between outputs in two cases, calculating the impact of feature f on the output. There can be various choices for the baselines, such as the mean value of the feature, zero, or it can be sampled from the training data. This can be formally defined as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{4.2}$$

where S is a subset of features, F is all features, $f_{S\cup\{i\}}$ is a model trained with feature i, $f_S(x_S)$ is the model trained without feature i. The results of SHAP show how the ranges of different features impact the model result.

Intrinsic dependencies might exist in the data. KernelSHAP has been developed to explain models that have dependent features as input (Lundberg and Lee, 2017). Due to the properties of the function, there is only a single solution to it, which can be approximated. The explanation is done disregarding the feature value. In return, this results in simplified inputs being used in the model such that $h_x(x') = x$, where x' is the simplified input h_x is the per sample simplification function. The following properties should hold for the values (Aas et al., 2019):

1. Local Accuracy: The explanation model should return the value of $f(x)$ when the original input x is given as input into the model.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i^{'} \tag{4.3}$$

2. Missingness: Missing features in the original data should not have an impact

$$x_i^{'} = 0 \Rightarrow \phi_i = 0 \tag{4.4}$$

3. Consistency: If feature i contributes more to the output of f' instead of f, the explanation model should have a higher coefficient for $\phi_i(f', x)$

$$f_x'(z') - f_x'(z' \backslash i) \geq f_x(z') - f_x(z' \backslash i) \tag{4.5}$$

Since the explanation model selects a subset of features and the model might not be able to deal with missing features, two assumptions are made on the data:

linearity and independence; both simplify the computation of expected values.

$$
\begin{aligned}
f(h_x(z')) &\approx E_{z_{\bar{S}}}[f(z)] && \text{feature independence} \\
&\approx f([z_s, E[z_{\bar{S}}]]]) && \text{linearity}
\end{aligned}
$$

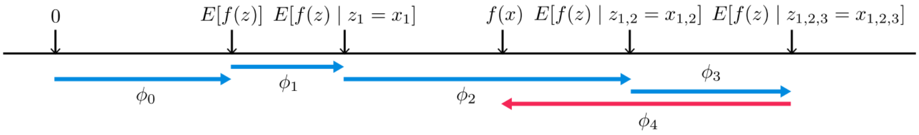

Figure 4.1: Sample SHAP values (Lundberg and Lee, 2017)

Figure 4.1 depicts how the addition of each feature changes the expected value of the model. The model starts with none of the features known and adds new features at each stage. The linearity and independence assumption makes the order of addition of features inconsequential. All possible orderings of ϕ_i are averaged while calculating the SHAP value.

SHAP has been used for various tasks such as text classification, ophthalmic conditions, and heavy metal detection in sewer water (Souza et al., 2021; Singh et al., 2021; Jiang et al., 2022). SHAP has been used to explain HABs in Geum River, South Korea (Park et al., 2022).

Gradient SHAP

This method alters the original SHAP by adding Gaussian noise to each input sample n times, selecting a random point between the baseline and the noisy input, and then computing the gradient of the outputs with respect to the randomly selected noisy points (Kokhlikyan et al., 2020). This method assumes the features of the model are independent, and the explanation model is linear between the inputs and the given baselines.

4.1.5 DeepLIFT

Developed by Shrikumar et al. (2017), DeepLIFT explains a model's behaviour by mapping its output to its input via backpropagation. The explanations are done by comparing the differences between baselines for the non-linear activations to find the neurons that deviate from the output (Kokhlikyan et al., 2020). The comparison is made by the concept of multipliers:

$$m_{\Delta x \Delta t} = \frac{C_{\Delta x \Delta t}}{\Delta x} \tag{4.6}$$

where Δx is the distance from the baseline, Δt is the distance from the expected output of neuron t, and $m_{\Delta x \Delta t}$ is the contribution of the neuron. The chain rule is applied and backpropagated through the network. The process done by DeepLIFT approximates Shapley values (Shrikumar et al., 2017).

DeepLIFT has been utilised for tasks such as question answering, traffic modelling and kidney cancer detection but not for early HAB detection (Arkhangelskaia and Dutta, 2019; Nascita et al., 2021; Zhou and Wei, 2022).

Works where explainability is used in the workflow for HAB detection are few. Hong et al. (2021) use Grad-CAM to explain model behaviour using the gradients in the final convolutional layer to create localisation maps for images.

4.2 Methodology

4.2.1 Problem Definition

Given the data for day i, the problem of algal bloom prediction could be modelled as:

$$f_n(S_i) = y_{i+n} \tag{4.7}$$

where S_i is the in-situ data and f is the model itself. y_{i+n} is a binary label for the dissolved oxygen threshold for n days ahead of observation day i.

4.2.2 The Data

The model in Figure 3.6, which consists of self-attention, LSTM and linear components, was used to fill in the missing values. The data used for this chapter is the same as the one in Section 3.2, the column that is used for labelling is *o2conc*. The following formula is used to calculate the maximum amount of dissolved oxygen concentration in the water given the temperature and salinity Garcia and Gordon (1992):

$$\begin{gathered} InC_O^* = A_0 + A_1T_+A_2T^2 + A_3T^2 + A_3T^3 + A_4T^4 + A_5T^5 + \\ S(B_0 + B_1T + B_2T^2 + B_3T^3) + C_0S^2 \end{gathered} \tag{4.8}$$

where $A_0, ..., A_5$, $B_0, ..., B_3$ and C are coefficients of the equation given in Table 4.1, S is the salinity, and T is $In[(298.15 - T_O)(273.15 + T_O)^{-1}]$, where T_O is the observed temperature value at time t. Algal bloom starts with the increased algal activity in a body of water which results in increased dissolved oxygen; therefore, thresholding was used, comparing the current dissolved oxygen to the maximum percentage of dissolved oxygen the water can hold at time t. If the percentage is 5% above the maximum threshold, the label is 1, else 0. The labelling process is done per day based on mean dissolved oxygen. The positive label percentages for each location are as follows: 1.44% for TH1, 3.89% for Dowsing, 3.98% for WestGab and 11.44% for LivBay. The relationship between temperature and salinity with dissolved oxygen is further proven by the Pearson correlation matrix in Figure 4.2. The baseline models for this chapter were chosen as the SVM and RF, as they were the most popular machine learning

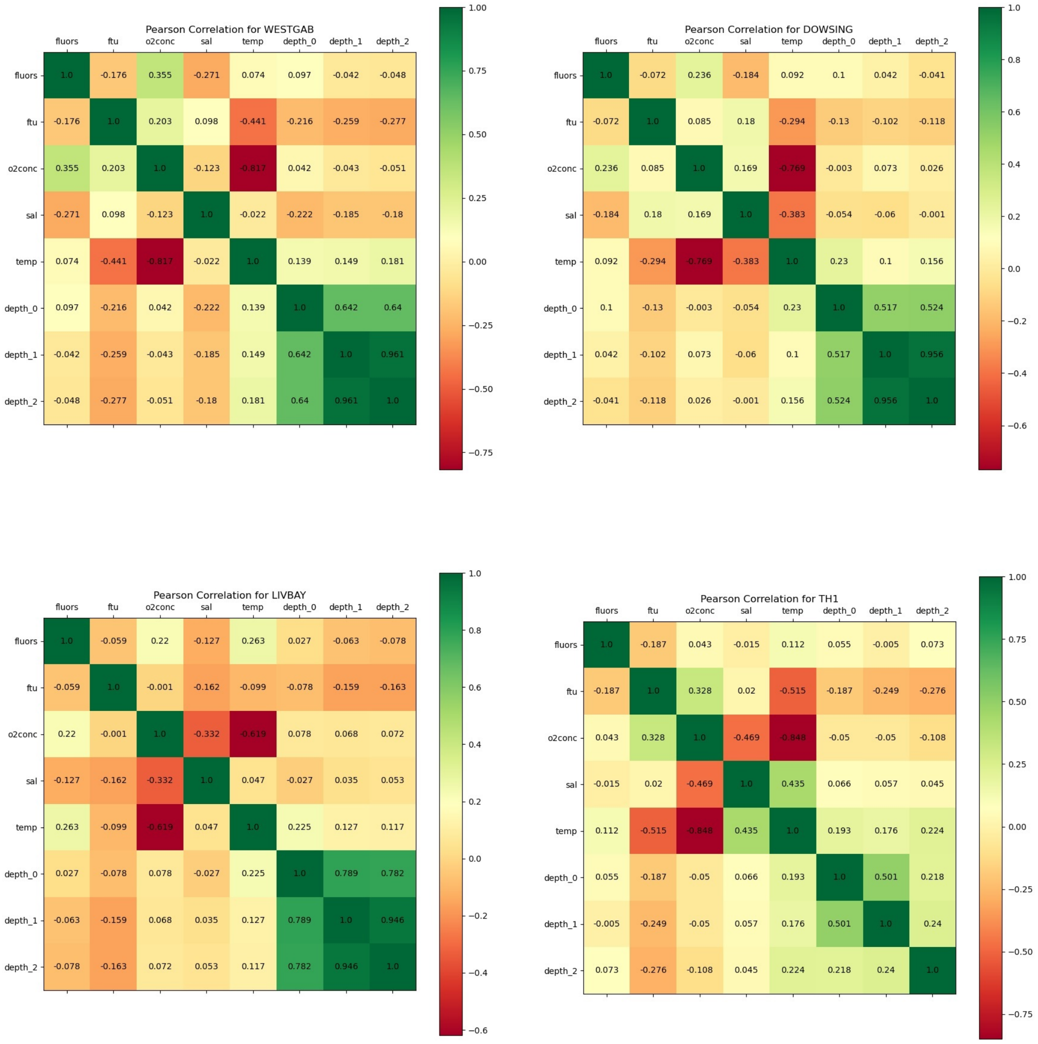

Figure 4.2: Pearson coefficient values

Coefficient	Value
A_0	2.00907
A_1	3.22014
A_2	4.05010
A_3	4.944457
A_4	$-2.56847 * 10^{-1}$
A_5	3.887674
B_0	$-6.24523 * 10^{-3}$
B_1	$-7.37614 * 10^{-3}$
B_2	$-1.03410 * 10^{-2}$
B_3	$-8.17083 * 10^{-3}$
C	$-4.88682 * 10^{-7}$

Table 4.1: Coefficients for Equation 4.8

models for this task, as outlined in Section 4.1. An IF model is included to observe if the abnormalities could be identified in an unsupervised fashion by identifying the differences between normal occurrences and abnormalities. A convolutional VAE is also included to see if relevant information could be extracted from a latent space regarding these abnormalities with varying filter sizes. A Luong attention model is also included to observe if any improvements could be made over LSTM models.

Time2vec

As in NLP tasks, embeddings can be used for time series data. In this case, Time2Vec is used (Kazemi et al., 2019). The embedding can be divided into two: time domain and frequency domain. The time domain is indicated by a single linear component, and the frequency domain is indicated by the periodic function F, such as sin or cos with $k-1$ components.

$$t2v(x_i) = \begin{cases} w_i^T x + m_i, & \text{if } i = 0 \\ F(w_i^T x + m_i), & \text{if } 1 \leq i \leq k \end{cases} \tag{4.9}$$

Time2vec is similar to the positional encoding of Vaswani et al. (2017). Unlike positional encoding, time2vec performs embedding in continuous time, so it is able to capture the periodicity of inputs.

4.2.3 Proposed Model

The proposed model, Transformer-Convolution (TF-Conv), consists of four components: a time embedding component (Time2Vec), a transformer, a convolutional layer and a linear layer with softmax (Vaswani et al., 2017; Kazemi et al., 2019). The embedding layer maps the input to two domains: time and frequency. The transformer is used to generate the sequence for i day(s) ahead, which ranges between 1-7. Separate embedding components are used for input and target sequences as they differ in their number of features. The input is the measurements of day x, and the target is the measurements of day $x + i$, where i is the number of days into the future ranging between 1 and 7. The input data is used to generate the target observations using the transformer network. The target variable is used during training to compute the loss between the generated sequence and the ground truth. Masking is used at the decoding stage of the transformer. During training, teacher forcing is used for the transformer. The ground truth is given as the target value during decoding. During testing, the previous output of the transformer is used as the target tensor, and initially, a tensor of zeros of shape $(1, batch\_size, num\_features)$ is given as the target. The convolutional layer is used for feature selection. The generated sequence does not include the dissolved oxygen so as not to overfit the convolution part of the model to only the dissolved oxygen. The generated sequence is taken through a 1-D convolution layer to serve as a feature selector. Lastly, the filtered observation is passed through a linear layer to classify the sequence. The labels were inversely weighted during training due to label imbalance in the dataset. The final output of the network is a binary variable which denotes whether or not the daily average dissolved oxygen is above the threshold or not. Figure 4.3 illustrates the proposed architecture. The training and testing procedures are provided in pseudocode format in Algorithm 2 and Algorithm 3.

Algorithm 2 TF-Conv training (single batch)

```
Ensure: $X_{src}$ = tensor of$(seq\_len, batch\_size, num\_features)$
Ensure: $X_{tgt}$ = tensor of$(seq\_len, batch\_size, num\_features - 1)$
  $X_{src} \leftarrow time2vec(X_{src})$
  $X_{tgt} \leftarrow time2vec(X_{tgt})$
  $X_{src} \leftarrow tf\_encode(X_{src})$
  $X_{src} \leftarrow tf\_decode(X_{src}, X_{tgt}, masks)$
  $X_{src} \leftarrow avg\_pool(GeLU(conv_1d(X_{src})))$
  $X_{src} \leftarrow softmax(linear(X_{src}))$
```

Algorithm 3 TF-Conv testing (single batch)

```
Ensure: $X_{src}$ = tensor of$(seq\_len, batch\_size, num\_features)$
Ensure: $X_{tgt}$ = tensor of zeros$(1, batch\_size, num\_features - 1)$
  $X_{src}, X_{tgt} \leftarrow time2vec(X_{src}), time2vec(X_{tgt})$
  $X_{src} \leftarrow tf\_encode(X_{src})$
  $outputs = [\,]$
  while $cur\_seq \neq seq\_len$ do
    $output \leftarrow tf\_decode(X_{src}[cur\_seq], X_{tgt}, masks)$
    $X_{tgt} \leftarrow output$
    $outputs.append(output)$
  end while
  $outputs \leftarrow avg\_pool(GeLU(conv\_1d(outputs)))$
  $outputs \leftarrow softmax(linear(outputs))$
```

GradientShap[1] was used as the explanation model. A tensor of zeroes is used as the baseline for the explanation model. The output of the explanation model is per sample and per time-step. To give an overall view, the explanations are aggregated per day, and an average is calculated per feature. The hyperparameters for this model are:

[1] https://captum.ai/api/gradient_shap.html

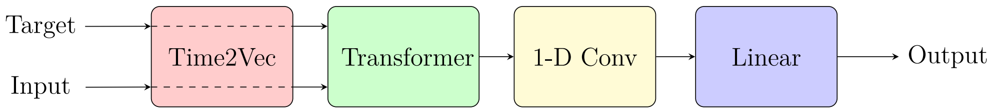

Figure 4.3: Proposed model for predicting oxygen thresholds. The input consists of all of the observed variables at day x, whereas the target consists of all variables except dissolved oxygen at day $x + i$. The transformer generates the target sequence for day $x + i$ except the dissolved oxygen. The output is a binary variable denoting if the average dissolved oxygen at day $x + i$ is below or above a threshold.

- Baseline: tensor of zeros
- Number of samples: 100

In Chapter 2, it is mentioned that most of the study sites relate to Far East Asia in China or Hong Kong, Lake Erie or the Coast of Florida in the U.S or the Red Sea. Our study location is unique in this sense. Most of the approaches mostly use models like SVM or RF, or using LSTMs to analyse the long/short term temporal patterns in the data. Our approach comes up with a possible sequence for n days after observation using a sequence-to-sequence approach and filtering the necessary parts of the generated sequence to predict the correct label.

4.2.4 Additional Experiments

Effect of Transformer Pre-training

An additional experiment is conducted to observe the effect of transformer pre-training on prediction performance. The experiment is conducted for each day using the same hyperparameters obtained in the original experiments.

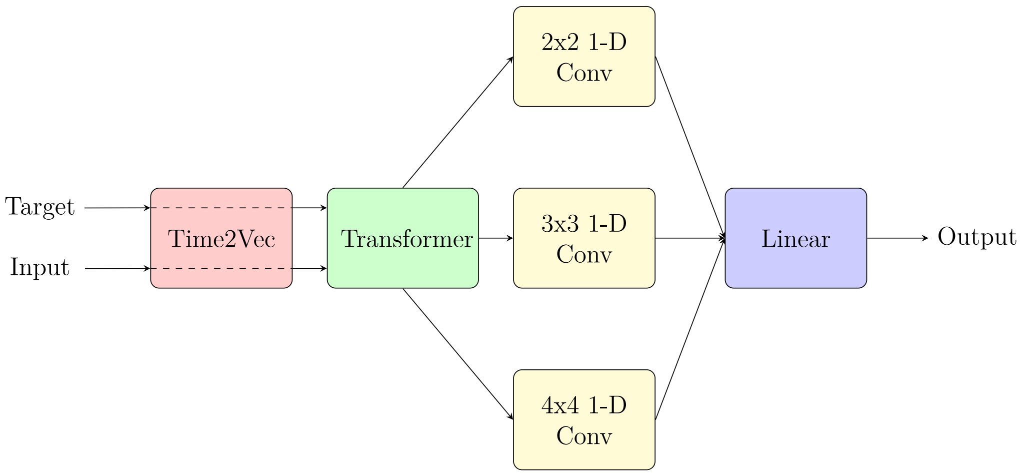

Figure 4.4: Second proposed model for predicting oxygen thresholds. The convolutional layer is modified to have several convolutions with different sizes.

Effect of Multiple Convolutions on Model Performance

Another experiment was set to observe if obtaining information from various sized filters would benefit the prediction performance, similar to the inception model He et al. (2016). Three filters were used with sizes $2x2$, $3x3$, and $4x4$, as seen in Figure 4.4. The generated sequence was forwarded through each filter, with the results concatenated and fed into the linear layer. Grid search was used for hyperparameter optimisation using the same set in the initial experimentation.

Differences Between Explanation Models

Different explanation models make different assumptions about model behaviour. By comparing different approaches, an explainability model's usability in various settings can be tested. In addition to SHAP, four additional explainers are used: LIME, kernel SHAP, DeepLift and DeepLift Shap. The hyperparameters for the models are:

- LIME:
  - Similarity function: Euclidean distance

  - Surrogate model: Ridge regression
  - Baseline: tensor of zeroes
  - Number of samples: 100

- DeepLIFT:
  - Baseline: tensor of zeroes

- DeepLIFT SHAP:
  - Baseline: tensor of zeroes

4.3 Results

The predictions are done i days into the future given the observation at day x. i ranges between 1 to 7. 70% of data of TH1 buoy was used for training, 30% for validation. This location was chosen due to nutrient flow from the River Thames. By modelling different nutrient concentrations, a more generalised model can be created. A single location was used for training to test the generalisability of the model and to assess the model performance with data gathered from various locations with different properties. The other three sites are used for testing.

The F1 scores of each day for each site are presented in Figure 4.5. The proposed model is able to generalise between locations with different properties. Other approaches such as RF need to be trained per location and per day to be usable. The mean F1 scores for all test locations are illustrated in Figure 4.6. With unseen data, the proposed approach outperforms all of the baselines. The F1 score was used as the performance metric due to the issue of label imbalance in the datasets. The weights of recall and precision were equal for the F1 score. An Adam optimiser was used for this task with 200 epochs and earlystopping with

Day	Batch Size	# of Encoder/Decoder Layers	# of Attention Heads	Transformer Network Dimensions	Learning Rate	Dropout Rate
1	16	2	2	32	0.001196	0.212
2	64	3	5	256	0.000606	0.512
3	6	1	2	32	0.002497	0.102
4	6	1	2	128	0.003346	0.136
5	4	3	2	128	0.003670	0.217
6	6	2	1	128	0.003635	0.115
7	6	2	1	32	0.003635	0.115

Table 4.2: Hyperparameters used for each model where the value of day is i days into the future.

a patience of 15 epochs Kingma and Ba (2014). The embedding size of time2vec was set to 10, and the convolution window size was set to 2 for all experiments. The rest of the hyperparameters are given in Table 4.2 based on the prediction day. The hyperparameter optimisation was done using grid search.

The results of the first additional experiment, which is the observation of pre-training on overall model performance, can be seen in Table 4.5. The pre-training was performed while hyperparameters were kept constant. Although the representation learning is improved through division of tasks, the overall performance is reduced. The results of the second additional experiment, which is the use of multiple convolutions, can be seen in Table 4.4. The use of multiple convolutions results in an intermediate state which captures information at different scales, resulting in a better performing model in the majority of the cases. The outputs of additional explainers are illustrated in Figures 4.7, 4.8, 4.9, and 4.10. Each explanation model makes different assumptions while generating the explanations, leading to different results.

4.4 Discussion

In terms of mean F-score, the proposed model TF-Conv is the most suitable model for the majority of the cases. The RF classifier had problems such as overfitting as it performs nearly perfectly in the training site, TH1, whereas it performs

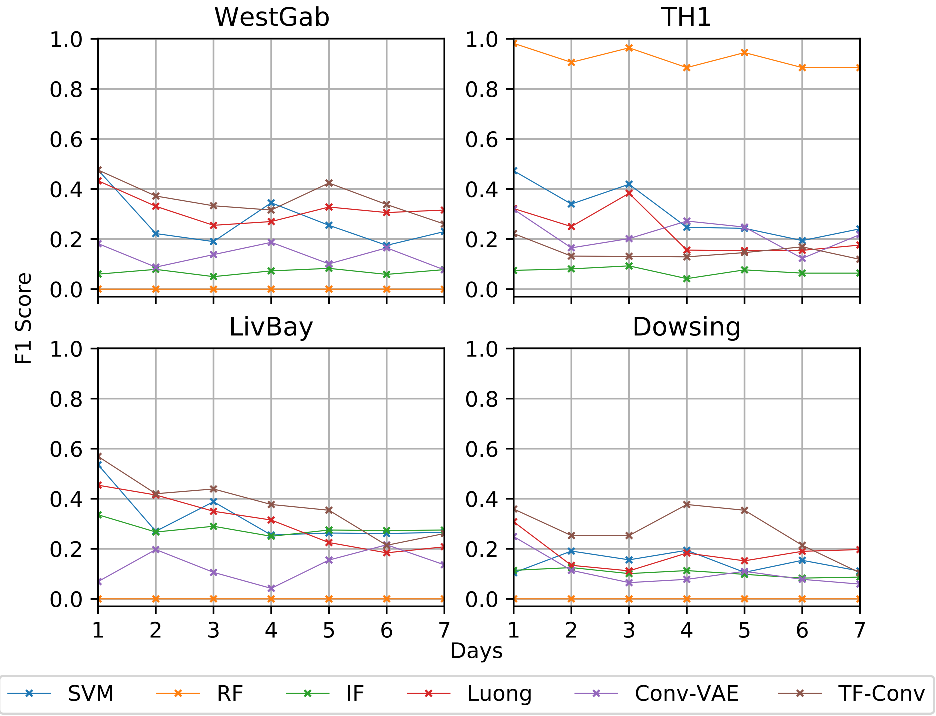

Figure 4.5: F1 scores for abnormality prediction for all 4 buoys

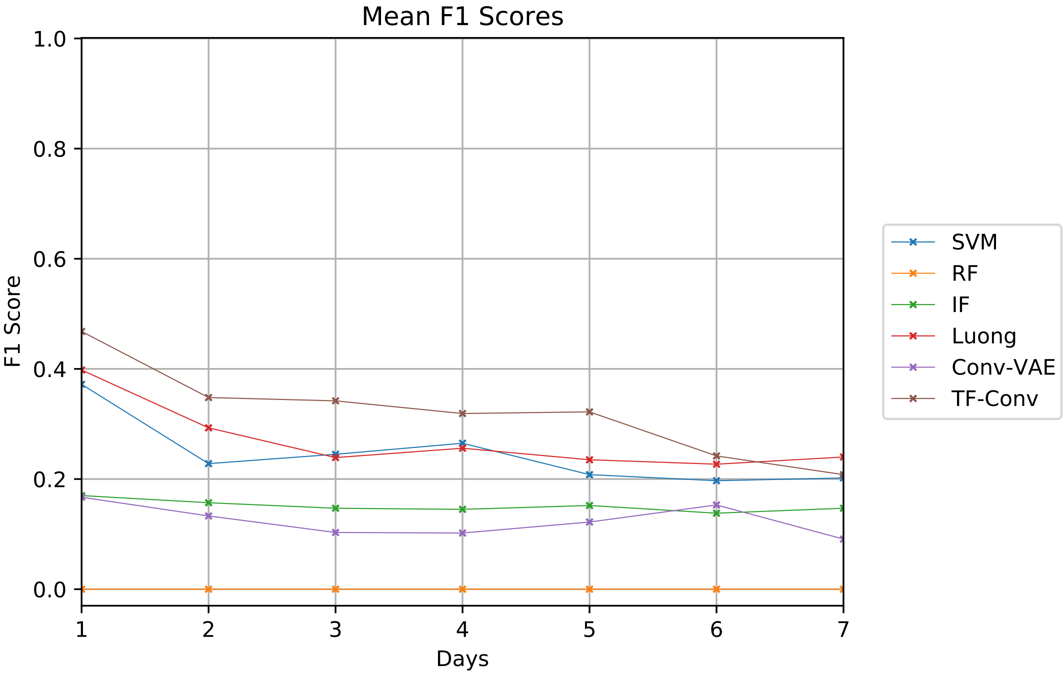

Figure 4.6: Mean F1 scores for abnormality prediction for testing buoys: WestGab, LivBay and Dowsing.

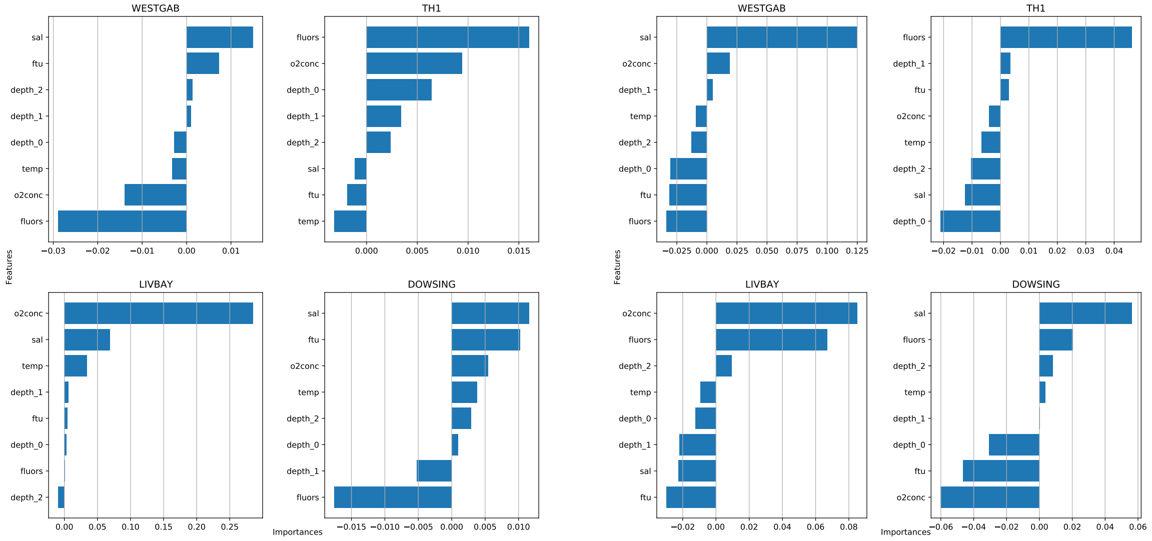

Figure 4.7: Left: Feature importances of SHAP for predictions 1-day ahead. Right: Feature importances of SHAP for predictions 7-days ahead.

	WestGab	**TH1**	**LivBay**	**Dowsing**
1	0.679	0.720	0.815	0.621
2	0.656	0.589	0.670	0.596
3	0.617	0.657	0.710	0.603
4	0.620	0.684	0.674	0.621
5	0.665	0.676	0.666	0.567
6	0.647	0.696	0.558	0.583
7	0.596	0.719	0.604	0.534

Table 4.3: AUC ROC for 1-7 days

poorly in other locations. The SVM classifier suffers from the same phenomenon for the Dowsing buoy. To obtain satisfactory results for the RF classifier, it could be trained on all four locations, which might cause memory issues and maintenance costs. IF assumes that the outliers in the data can be predicted due to their different properties and low occurrence rates. The results show that the increased activities in all sites were not outliers due to their properties, and the assumptions made by the IF do not hold.

The decreasing performance of the attention model between day 1 and day

Day	Single-Conv	Multi-Conv	Difference(%)
1	0.468	0.445	-4.92
2	0.348	0.406	16.66
3	0.342	0.360	5.26
4	0.319	0.330	3.44
5	0.322	0.315	-2.18
6	0.242	0.280	15.70
7	0.208	0.245	17.78

Table 4.4: Results for TF-Conv without and with multiple convolutions

Day	Classic	Pre-trained
1	0.468	0
2	0.348	0.113
3	0.342	0
4	0.319	0.014
5	0.322	0
6	0.242	0.11
7	0.208	0

Table 4.5: Results for TF-Conv without and with pretraining of the transformer

6 indicates that Luong attention is not suitable for predicting the near future blooms, but it may be suitable for prediction for days further into the future. The inputs for the deep learning models are aggregated based on observation day, whereas the machine learning models use averages of features based on observation day due to the model's limitation of not being able to model tensors with more than two dimensions. The use of aggregation aids the deep learning models' generalisability since these models are exposed to raw data rather than a summarized version. Even with a summarized version of the data, the RF classifier performs better in a singular site comparison, but the trade-off is made in generalisability.

The explanation model used was *GradientShap*, which works by adding random noise to data samples that were sampled between the baseline and the input

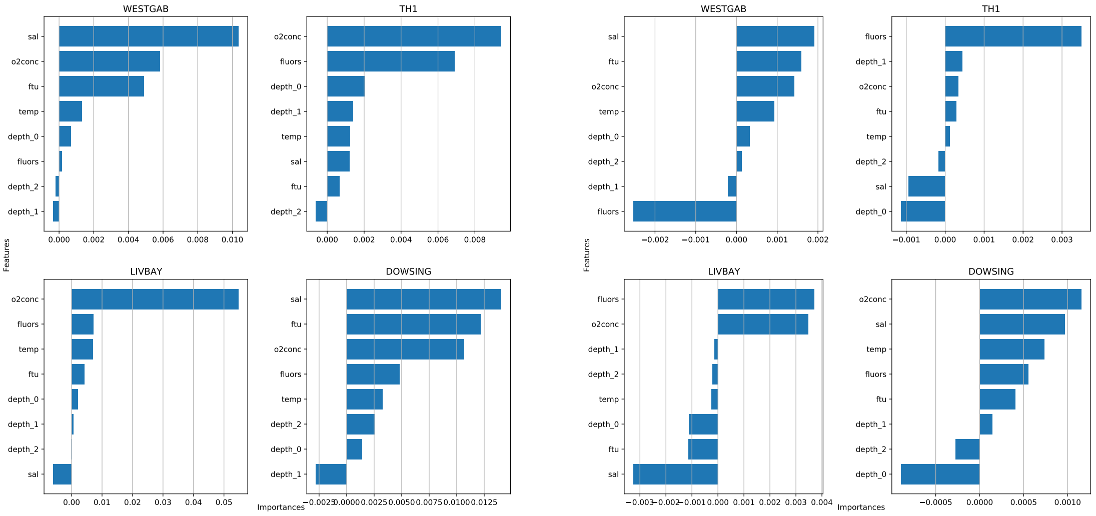

Figure 4.8: Left: Feature importances of LIME for predictions 1-day ahead. Right: Feature importances of LIME for predictions 7-days ahead.

and computing the gradients. The explanations differ from site to site, as seen in Figure 4.7. It also shows that the order and the magnitude of the importances change from day to day. The model used assumes feature independence, and the explanation model is linear. The explanation models show that each site has its own properties, and the site with the most positive labels (LivBay) and the best performance out of all sites has *o*2*conc* as the most important feature, which indicates that tracking the *o*2*conc* in the water might be useful where abnormalities frequently occur while using the TF-Conv model. The explanations also give insight into how input features differ from one another depending on the prediction day, empirically showing the requirement of training a model for each prediction day.

The use of multiple convolutions aids the model in filtering the sequence with different window sizes, resulting in varying information being passed to the next stage of the network. Using multiple convolutions aids the model in the majority of the cases, which can be observed in Table 4.4. Further hyperparameter tuning

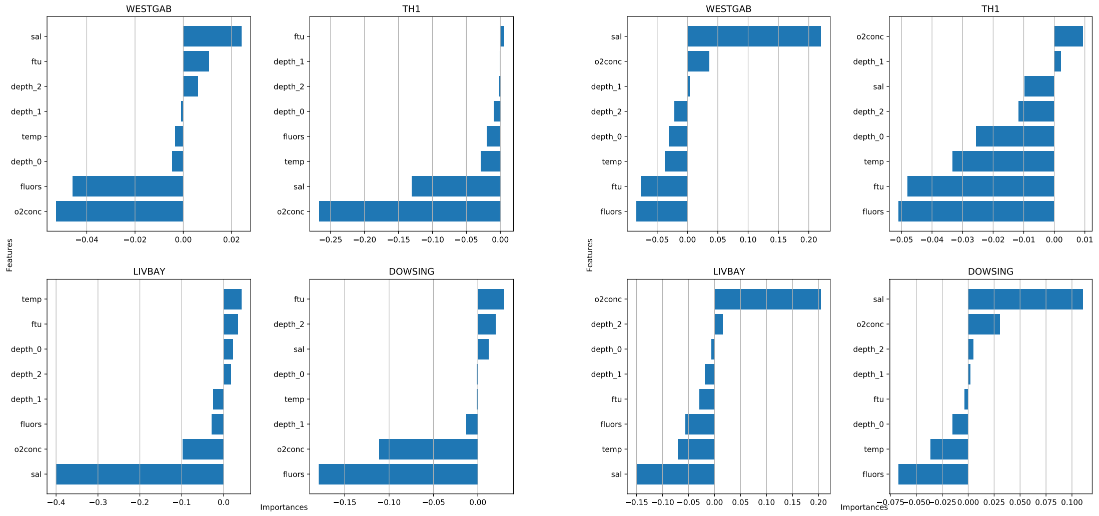

Figure 4.9: Left: Feature importances of DeepLift for predictions 1-day ahead. Right: Feature importances of DeepLift for predictions 7-days ahead.

could be done to increase model performance if seen as necessary.

To observe the effect of pre-training, every specification of the training was kept the same except the training process itself. Table 4.5 indicates that pre-training impacts the model negatively. Hyperparameter tuning may yield different results, which can be done for future work.

Different explanation models make different assumptions. All of the models that were chosen are local model agnostic. SHAP views the explainability of the model from a game theoretical approach. LIME uses explainable surrogate models such as linear regression. DeepLIFT explains model behaviour through backpropagation and activation function behaviour. DeepLIFT SHAP approximates SHAP values using DeepLIFT. As the constraints change, the values for each feature change, which can be observed in Figures 4.8, 4.9, 4.10, and 4.7. The approximations done by DeepLIFT SHAP may not be similar to SHAP values due to hyperparameter choices. For predicting a day ahead, all models detect that salinity is the most important feature that affects the model's output for

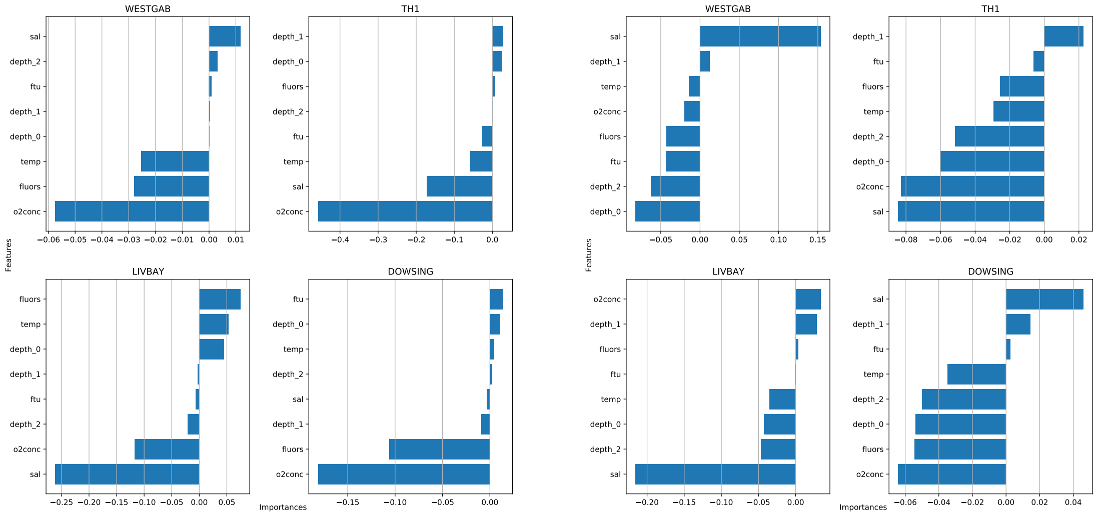

Figure 4.10: Left: Feature importances of DeepLift SHAP for predictions 1-day ahead. Right: Feature importances of DeepLift SHAP for predictions 7-days ahead.

WestGab data, and salinity or turbidity is either the most or the second most important feature for the Dowsing buoy. A similar pattern is observed for predicting seven days ahead for sites WestGab, TH1 and LivBay. DeepLift and DeepLift SHAP tend to focus on negative features, as seen from the plots, which might not reflect the detection capabilities of the models.

The Area Under the Receiver Operating Characteristic Curve (AUC ROC) scores for the TF-Conv model, as observed in Table 4.3, and the confusion matrices, Figure 4.11, allow us to investigate the performance of the model further. The model is better than random guessing for this task as the AUC ROC score is > 0.5; however, the increase and decrease in scores are not as expected. The scores do not decrease as the predictions move further into the future; rather, they change randomly. This may mean the model is unstable. The confusion matrix reveals that the model misses most cases in the WestGab and Dowsing site, having high FN and FP rates. This entails that predicting HABs using classification for these four sites may not be suitable. An alternative approach is

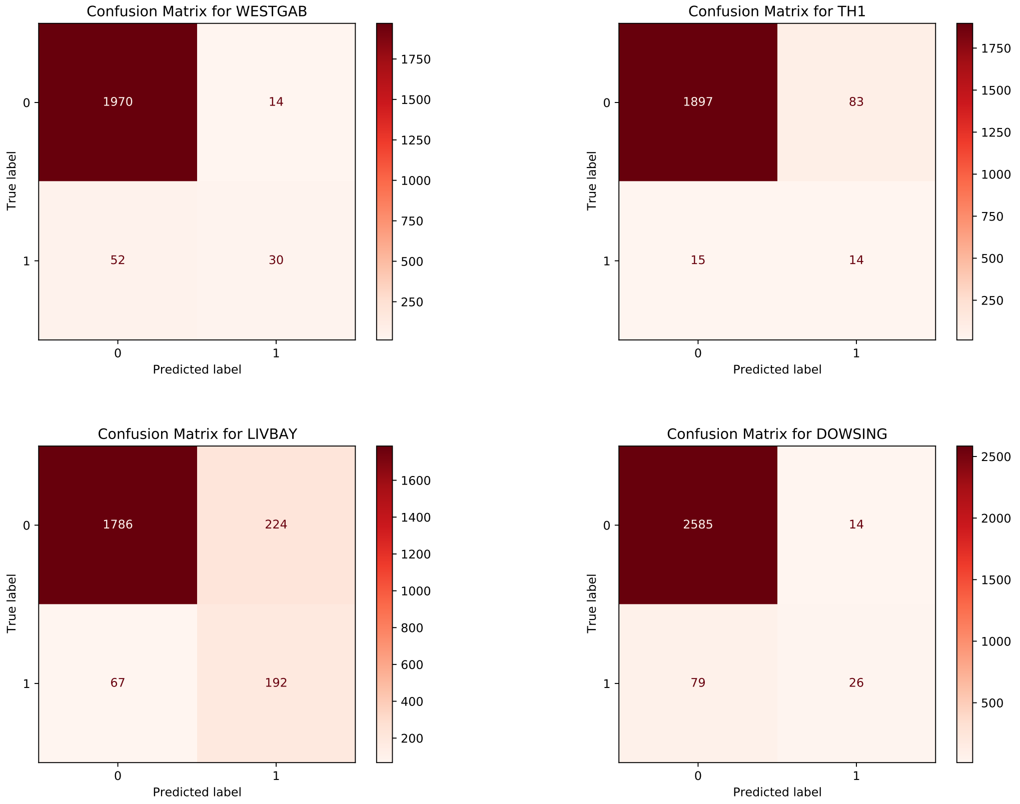

Figure 4.11: Confusion matrices for all sites for predicting a day ahead

explored in Chapter 5, applying regression with a multimodal learning approach.

4.5 Conclusion

In this chapter, a novel model is proposed for detecting algal blooms by predicting dissolved oxygen concentration 1 to 7 days ahead using time embeddings, a transformer network and a convolutional layer. The proposed model increases the prediction performance by 5% in terms of F-score on average, ranging from 1 to 7 days ahead of occurrence. The importance of each feature is provided with SHAP values per day, increasing the interpretability of the model. It has been observed that the most important feature changes based on the monitoring site and prediction day. Analysing the results obtained from AUC ROC, it is noticed that the models perform better than random guessing as the score is always higher than 0.5. On the other hand, the sample of confusion matrices shows that the model is not able to capture the majority of the HAB incidences, strongly suggesting that classification might not be the optimal solution for detection.

Three additional experiments are done to observe various training and evaluation settings. It has been observed that pre-training the transformer is not efficient in obtaining better results and using multiple convolutions to filter a sequence benefits the performance of the model in the majority of the cases. It has been noticed that using different explanation models results in different outcomes depending on a model's assumptions and might not reflect the properties of the monitoring site.

Data with different frequencies, such as ship-based data or data with different modalities could be used to improve the detection process. This work could be extended to closed bodies of water. The current results indicate that models could be tested for different day ranges than they were trained on to test the model's generalisability. The stability of the model could be checked by predicting bloom events further than seven days. The performance of the model could be assessed by training it per location. Generalisability among different locations was not included in the scope of this chapter, and transfer learning methods could be used in the future to test the efficiency of this architecture.

Chapter 5

Multimodal Learning Approach to Algal Bloom Prediction

In the event of HABs, colour changes occur in water. The blooms may cause health hazards to humans and livestock through the ingestion of such water sources (Falconer, 1999). To ensure public health and safety, these blooms must be detected and controlled before the condition exacerbates.

Different forms of data could be used for detection. In-situ data such as buoys or water samples analysed in labs, satellite data such as MODIS for detecting colour or nutrient changes from various bands or text from social media such as Twitter when the monitored sites are close to populated areas could be used for detection purposes. To improve the detection process, the information from different data modalities could be analysed simultaneously.

As seen in Chapter 4, HAB detection can be done via in-situ sensors. This chapter focuses on detecting HABs in another data format, satellite imagery, in combination with in-situ data. Satellite imagery has been used for tasks such as land cover classification and disaster mitigation (Rakhlin et al., 2018; Fayne et al., 2017). Deep learning can be used to analyse these images in cases such as population estimation and volcano deformation (Robinson et al., 2017; Anantrasirichai et al., 2019). Images of water bodies obtained via satellite can also be analysed using deep learning. Using various deep learning layers, detection of algal blooms could be done by segmentation and convolutional neural network elements. Using the images of the same location through time, another type of early warning system could be developed or a system that predicts the next stages of the bloom. Using multiple modalities reduces dependencies on using a single mode of data as alternate modalities could be used as replacements in situations where data collection issues arise such as cloud cover for satellite data and biofouling for insitu data.

There are several previous works that have focused on the prediction of HABs from satellite images. Methods include HABNet, which uses a mix of CNN, LSTM and machine learning components focusing on the Arabian Gulf and the Gulf of Mexico, linear neural networks focusing on West Florida, CNN architectures focusing on lakes in China and neural and fuzzy neural networks focusing on Cefni Reservoir of Anglesey, the U.K. (Hill et al., 2020; El-Habashi et al., 2016; Pu et al., 2019; Silva and Panella, 2018). Most of the focus falls on East Asia, the U.S and the Baltic Sea (Sebastiá-Frasquet et al., 2020).

5.1 Related Work

The current approaches use each data type separately. Shehhi and Kaya (2020) use MODIS data to predict chl-a, sea surface temperature and fluorescence line height with SARIMA, regression and ANN. Hill et al. (2020) classify HAB events using twelve different MODIS channels with CNN, LSTM and ML methods. Hu et al. (2005) use MODIS data to detect and trace red tides in Florida Bay and use in-situ data to compare the predicted chl-a values from the MODIS data. Vannah and Chang (2013) combine Medium Resolution Imaging Spectrometer (MERIS), MODIS and in-situ data before training a genetic programming model to measure phycocyanin. Cao et al. (2020) use XGBoost to predict chl-a levels in several lakes in China and use in-situ data for validation.

Satellite data could be used to detect HABs using infrared, near-infrared and blue to green ratio based approaches (Binding et al., 2013; Clark et al., 2017). El-Habashi et al. (2016) analyse MODIS and Visible Infrared Imaging Radiometer Suite (VIIRS) satellite data with ANNs to predict the chl-a concentration. Hill et al. (2020) use a mix of CNN, LSTM and ML methods to detect HABs using a number of modalities of satellite data in West Florida. The approach used for this model consists of splitting all of the data in a given date window of bloom occurrence with varying days into the future. Lin et al. (2018) use Landsat data to predict the chl-a concentration in Lake Erie, the USA, using multiple linear regression, non-linear general additive models and boosted regression trees. Hu et al. (2020) use NOMAD 2.0 and SeaBASS satellite data in combination with SVRs to predict the chl-a concentration. This approach aimed to create a more generalised model by testing the SVR model with SeaWiFS data which is sampled globally. Kim et al. (2019a) utilise the U-Net architecture around the Korean Peninsula to detect red tides between 2011 and 2018. Tian and Huang (2019) predict HABs up to 5 days using satellite data. ANNs were used in this work, with the chl-a data classified into four levels. Gokul et al. (2019) use MODIS-Aqua data with a second-order derivative approach to detect and monitor HABs in the Red Sea. Mehrabian and Pahlevan (2019) use IF to predict algal blooms using Landsat-8 and Sentinel-2 images. However, they use 12 different IF, one for each month leading to unnecessary complexity. Pahlevan et al. (2020) use Mixture

Density Networks with satellite data to predict chl-a concentrations in various locations. Gokaraju et al. (2011) use kernel SVMs and Kernel Principal Component Analysis to detect algal blooms in the Gulf of Mexico with MODIS and SeaWiFS data. Ananias and Negri (2021) use SVM to analyse algal blooms in Lake Erie, U.S and Lake Taihu, China, with satellite data. Izadi et al. (2021) forecast algal blooms in a 5-9 day range using satellite in the Gulf of Florida with SVM, RF and XGBoost. Song et al. (2015) use MODIS and MERIS to detect algal blooms using thresholding and use in-situ data for validation in Monterey Bay, U.S. Ghatkar et al. (2019) use XGBoost to detect algal blooms in the Arabian Sea and the Bay of Bengal. Yussof et al. (2021) use LSTMs and CNNs in conjunction with level 3 MODIS AQUA data to predict chl-a in Sabah, Indonesia. The advantages and drawbacks of the models used can be found in Section 2.5.1 (ANN), Section 2.5.2 (CNN), Section 2.5.3 (LSTM), Section 2.4.4(XGBoost), Section 2.4.2 (RF) and Section 2.4.1 (SVR).

The in-situ approaches were studied in Section 4.1 of this thesis.

It can be noticed that for this task, either satellite data or sensory data is used for analysis but not both. In some cases where satellite data is used, in-situ data is used for verification purposes. Using only satellite data reduces the temporal prediction capabilities of models as the data is infrequent. Using only in-situ data reduces the spatial extendibility of the predictions as the observations are location specific. The span of data used for early HAB detection is usually shorter than a year, reducing the generalisability of the model (Sebastiá-Frasquet et al., 2020). The use of multiple modalities reduces the effect of problems that occur during data collection such as biofouling of in-situ sensors or clouds covering the observation area as by learning multiple modalities at the same time enables the model to substitute one modality for the other one when needed.

In this chapter, various multimodal approaches are proposed that use in-situ and satellite data simultaneously, exposing both the temporal and spatial information for the observation sites. Detecting only a single variable, such as chl-a or dissolved oxygen, has no applicability for the end-user, and other contextual information is needed. The proposed multimodal fusion model predicts additional variables, temperature and salinity, which affect the maximum amount of oxygen the water can contain, as stated in Equation 4.8. Using the predicted variables, the oxygen saturation at time t is calculated, providing more information to the end-users. Additional contributions are made for the problem of missing data through coordinated representation learning by creating a single representation space for both satellite and in-situ data during learning and using only a single modality during prediction.

5.1.1 Challenges

Temporal Frequency Difference The temporal frequency of the in-situ data differs from satellite data as in-situ data is collected multiple times per day, whereas there is only a single corrected image per day for each location for satellite data. The API used for collecting MODIS data selects the best observation per day from a 16-day period on several criteria, such as cloud coverage and view angles.

Spatial Resolution Difference The in-situ data collects information close to the buoy, whereas the satellite data used covers a 6x6 km area, upsampled to 256x256 pixels, which can provide information about the area surrounding the monitoring site. The upsampling is done due to the model input specifications of CNNs.

Generalisability Different algal species are found in different locations. Each species produces toxins with various levels of toxicity. Due to this variation, it becomes harder to pinpoint which blooms are harmful or not. The exact species can be determined by lab-tested samples, which require transportation and time. Therefore, nearly all of the studies focus on a single water body. To make the model more general, it becomes essential to train it with data from various locations. A non-general model cannot be used for different locations due to the water bodies and algal species' properties. With sufficient data, utilising the properties of various locations results in a generalised model.

5.2 Multimodal Learning

Multimodal learning is defined as using different data sources as input to train a model, such as using video and audio for text transcription. It consists of five approaches: representation, translation, alignment, fusion and co-learning (Baltrušaitis et al., 2018). A sample of multimodal learning settings is given in Table 5.1 for in-situ and satellite data.

5.2.1 Representation

A multimodal representation of data is achieved by using information from multiple entities. There are two main ways of representing multimodal data: joint and coordinated representation (Baltrušaitis et al., 2018).

Joint Representations This type of representation is used in tasks where all modes of data are present in both training and testing and it is stated as Baltrušaitis et al. (2018):

$$x_m = f(x_1, x_2, ...x_n) \tag{5.1}$$

	Feature Learning	Supervised Training	Testing
Classic Deep Learning	In-situ Satellite Imagery	In-situ Satellite Imagery	In-situ Satellite Imagery
Multimodal Fusion	In-situ + Satellite Imagery	In-situ + Satellite Imagery	In-situ + Satellite Imagery
Cross Modality Learning	In-situ + Satellite Imagery	In-situ	In-situ
	In-situ + Satellite Imagery	Satellite Imagery	Satellite Imagery
Shared Representation Learning	In-situ + Satellite Imagery	In-situ	Satellite Imagery
	In-situ + Satellite Imagery	Satellite Imagery	In-situ

Table 5.1: Sample multimodal settings adapted from Guo et al. (2019)

Typically used models include CNN and RNN. Using deep learning models results in an intersection between multimodal representation learning and multimodal fusion as a fusion strategy is required at one stage to concatenate the information from different modalities (Baltrušaitis et al., 2018).

Coordinated Representations In coordinated representations, different modalities are learned separately but with additional constraints. Examples of this approach include DeVise embedding (Frome et al., 2013), deep cross-modal hashing (Jiang and Li, 2017), and kernel canonical correlation analysis (Lai and Fyfe, 2000).

In the context of HAB detection, multiple modalities such as in-situ and satellite data could be fused with a joint representation approach or used in place of one another using a coordinated representation approach.

5.2.2 Translation

Multimodal approaches include translating from one modality into another, such as audio signals to text or text to images. Translation is done via two approaches: example-based or generative approaches.

Example-Based Translation Example-based translation is divided into two categories:

- Retrieval-based translation: This translation is done by finding the closest sample to the input in unimodal or semantic space. This approach has been used for speech synthesis (Bregler et al., 1997) and text-to-speech systems (Hunt and Black, 1996).
- Combination-based translation: This translation approach builds upon retrieval-based translation and combines samples to return more meaningful translations. Some examples are image description generation with Linear Programming and hand-crafted rules (Kuznetsova et al., 2012) and CNNs (Lebret et al., 2015).

Generative Translation This approach performs multiple translations given the source modality. It is divided into three categories:

- Grammar-based translation: This approach depends on a pre-defined grammar for generating another modality. It is used in creating video descriptions (Barbu et al., 2012) and image descriptions (Yao et al., 2010).
- Encoder-decoder translation: This approach is achieved via neural networks for tasks such as machine translation (Kalchbrenner and Blunsom, 2013), image generation (Mansimov et al., 2015) and speech generation (Owens et al., 2016).

- Continuous generation translation: This approach is used for sequence generation, where an output is given for each time step. Taylor et al. (2012) use Hidden Markov Models for visual speech generation, and Deena and Galata (2009) use Gaussian Process for audio-based visual speech synthesis.

Translation approaches cannot be used for HAB detection as the task is either a classification or a regression one. An alternate task in this domain could be the translation of in-situ data to create aerial imagery around the observation site to populate artificial datasets and vice versa.

5.2.3 Alignment

Alignment is defined as finding commonalities between different modalities. It is applied with two approaches:

Explicit Alignment In this approach, alignment is done using sub-components in different modalities using a distance metric. It is done in both an unsupervised and a supervised manner. Noulas et al. (2011) use Bayesian Networks to align speakers to videos. Yu and Ballard (2004) use generative graphical models to align objects in images with audio input. Supervised approaches include deep learning models like CNN for measuring similarities between image and text (Mao et al., 2016) and LSTM for finding similarities between images and their descriptions (Hu et al., 2016).

Implicit Alignment This approach is used as an intermediate for another task. Two methods are used for this approach: graphical models and neural networks. Graphical models have been used for language translation (Vogel et al., 1996). Attention models have been used for image captioning (Xu et al., 2015).

Alignment could be used utilised by using a common representation for multiple modalities. During training, the distance among different modalities could be minimised based on the observation day, enabling the model to expose commonalities and use modalities in place of one other.

5.2.4 Fusion

The definition of multimodal fusion is to combine different modalities to predict a single outcome; a class or a numerical value. There are two approaches for fusion:

- Model-agnostic fusion: This approach works by fusing data at different stages: early, mid (hybrid) or late fusion. Early fusion is done by fusing the input data before feeding it into the model. It aims to exploit low-level features of data. Late fusion is done after each modality is processed and fused without further steps and predictions are done through averaging (Shutova et al., 2016) or voting mechanisms (Morvant et al., 2014). Middle fusion is done after each modality is processed, and the intermediate structure is fused for further analysis. It has been used for tasks such as multimodal speaker identification (Wu et al., 2006).

- Model-based fusion: This approach has three variations: Multiple kernel learning, graphical models and neural networks. Multiple kernel learning is an extension of the kernel SVM that has been used for tasks such as multimodal sentiment analysis (Poria et al., 2015) and multimodal affect recognition (Jaques et al., 2015). Graphical models used include Hidden Markov Model (Gurban et al., 2008) and Conditional Random Fields (CRF) (Fidler et al., 2013). In neural networks, RNN has been used extensively, ranging from question answering (Gao et al., 2015) to video description generation (Jin and Liang, 2016).

Both model-based and model-agnostic fusion could be utilised for HAB detection with multiple modalities such as lab analysed samples, in-situ data and aerial imagery.

5.2.5 Co-learning

Co-learning is defined as aiding a resource-poor modality with a resource-rich one. The resource-rich data is used in training but not in testing. Co-learning can be divided into three approaches: parallel, non-parallel and hybrid (Baltrušaitis et al., 2018).

- Parallel Co-learning: In this approach, both modalities share instances such as images and their descriptions. Co-training is done using several weak classifiers that are trained on each modality to label unlabelled data. Co-training has been used for audio-visual speech recognition (Christoudias et al., 2006). Transfer learning is another strategy for parallel co-learning with models such as multimodal autoencoders (Ngiam et al., 2011) and multimodal DBMs (Srivastava et al., 2012).

- Non-parallel Co-learning: Non-parallel data share categories or concepts but not samples. Strategies used include transfer learning and zero-shot learning. Transfer learning enables the transfer of information of different modalities (Baltrušaitis et al., 2018). Zero-shot learning is training a model in such a way that it is able to detect classes it has not seen in the training data. Popular application areas include object classification. In multimodal learning, zero-shot learning is achieved by acquiring information from one modality that is not present in another.

- Hybrid Co-learning: In this approach, two non-parallel modalities share a modality. Examples such as multilingual image captioning, where the image is the shared modality belong to this category.

Transfer learning approaches could be used to utilise non-parallel co-learning for the task of HAB detection. Images and in-situ data where incidents of HAB are observed could be used in conjunction with parallel co-learning approaches.

In the context of HAB detection, different modalities are used for exposing different kinds of information regarding incidents. With satellite imagery, colour changes could be observed using various bands, and with in-situ data, nutrient monitoring could be applied pre-emptively to detect incidents. Different modalities could be combined to improve detection models by fusing/replacing different kinds of information.

5.3 Methodology

5.3.1 Problem Definition

Given the data for day i, the problem of algal bloom prediction could be modelled as:

$$f(S_i, M_i, C_i) = (y^1_{i+n}, y^2_{i+n}, y^3_{i+n}) \tag{5.2}$$

where S_i is the in-situ data, M_i is the MODIS data, C_i is the nutrient/algal data gathered from various satellites, and f is the model itself. $(y^1_{i+n}, y^2_{i+n}, y^3_{i+n})$ is the generated values for dissolved oxygen, salinity and temperature for n days ahead of observation day i.

5.3.2 The Data

In addition to the data from the previous chapter, two new data sources are used; one from MODIS observations and another from Copernicus Marine Service (CMS). chl-a's absorption peaks are between 450 nm and 650 nm, whereas phycocyanin, a toxin released by algae, peaks around 615 nm, overlapping with chl-a (Simis et al., 2012). Therefore, the data gathered by satellite could be used for detecting algal blooms.

MODIS data used in this project is collected from Sentinel Hub [1]. MODIS Satellite gathers data from 36 different bands with varying resolutions (250 m for bands 1-2, 500 meters for bands 3-7 and 1 km for bands 8-36). Algal blooms can be detected using True Colour Bands (RGB) 1, 4 and 3. Additional bands such as False Colour Bands 2, 1 and 4 and Normalized Difference Water Index (NDWI) Bands $(B4 - B2)/(B4 + B2)$. The differences in data are depicted in Figure 5.1. The data is collected with a resolution of 500m 6x6 km around each monitoring site, upsampled to 256x256 using bicubic interpolation. Each pixel contains the best information from a 16-day period depending on a number of factors such as observation coverage, cloud coverage, view angles etc. The data covered by MODIS ranges from the 24th of February 2000 to December 2019.

CMS data used is titled *OCEANCOLOUR_ATL_CHL_L4_REP_OBSER VATIONS_009_098* [2]. The data is gathered by several satellites, SeaWiFS, MODIS, MERIS, VIIRS etc. The data is collected as daily-mean with a resolution of 1 km. A region of 6x6 km is gathered around each monitoring site, upsampled to 256x256 with bicubic interpolation. The data gathered is Level 4 data which went through the process of interpolation. Each pixel contains the daily mean value for chl-a. The data covered by CMS ranges from the 4th of September 1997

[1] https://www.sentinel-hub.com/

[2] now renamed *OCEANCOLOUR_ATL_BGC_L4_MY_009_118*

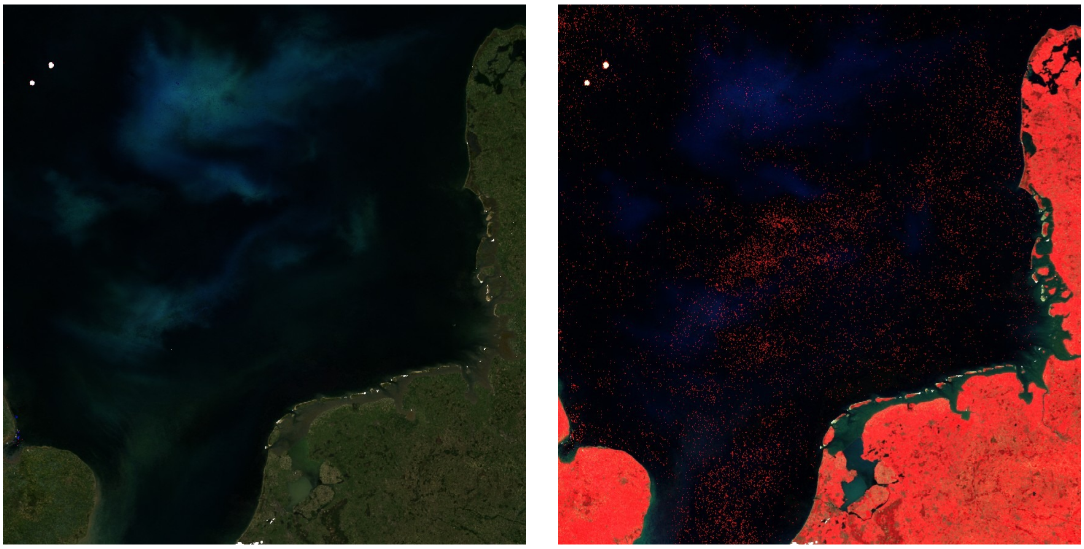

Figure 5.1: Left: RGB image of a bloom in the North Sea on June 2015 Right: False Colour image of a bloom in the North Sea on June 2015

to December 2019.

5.3.3 Proposed Models

A multimodal fusion approach with joint representation is proposed for the task of early algal bloom detection. The proposed model is outlined in Figure 5.2. The used CNN architectures are outlined in Section 5.3.4. The hyperparameters for the transformer component are transferred from Chapter 4 to reduce the search space. The model's outputs are three values; temperature, salinity and oxygen. After concatenating the outputs of hidden layers, a linear layer or an XGBoost is used to predict the aforementioned three values. Using these three predicted values and Equation 4.8, the continuous variables are transformed into percentage values and compared with the ground truth using Mean Absolute Error (MAE). The pseudocode is presented in Algorithm 4 and Algorithm 5. Before the percentage calculation, the multimodal results are compared to unimodal approaches using RMSE: SVM, k-NN, MLP and Luong attention LSTM.

A number of hyperparameters were chosen for XGBoost tuning. These are: eta, max_depth, min_child_weight, subsample, colsample_bytree, n_rounds, target_var. eta is the learning rate, max_depth is the maximum depth each tree can have, min_child_weight is the minimum number of instances for each node in the tree, subsample is the ratio of the training samples used for that tree, colsample_bytree is the ratio of feature columns selected for each tree, n_rounds is the maximum number trees the model can have.

An approach similar to Chen et al. (2018) is followed for training the XGBoost model. The individual components of modalities are trained using a linear layer after the concatenation. The linear layer is removed, and the concatenated individual outputs are given as input to the XGBoost for training, transferring the learned individual representations.

The thresholding process differs from the one in Chapter 4. In Chapter 4, thresholding was done by applying Equation 4.8 to each data point in the time series, the percentage was calculated per data point, and the label was given based on the average percentage per day. In this chapter, the average daily temperature, salinity and dissolved oxygen values per day are given as input to Equation 4.8 and a percentage is calculated.

5.3.4 CNN Models

Various CNN architectures were used for experimentation. The models were chosen based on variety and different structures.

ResNet Developed by He et al. (2016), this architecture mimics the structure of pyramidal cells, which includes skip connections between convolutional components. Two versions of ResNet were used for experimentation: ResNet18 &

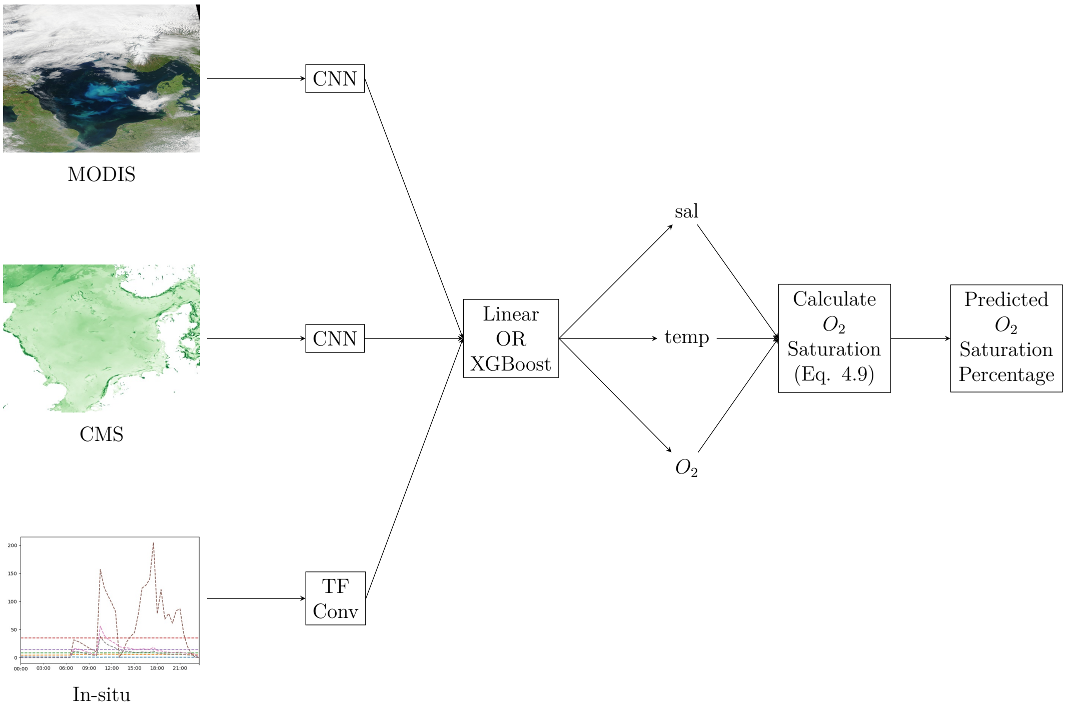
MODIS
CMS
In-situ
CNN
CNN
TF Conv
Linear OR XGBoost
sal
temp
O_2
Calculate O_2 Saturation (Eq. 4.9)
Predicted O_2 Saturation Percentage

Figure 5.2: The proposed multimodal fusion approach. The model takes in three different tensors as input; two satellite data modalities with $(batch\_size, num\_channels = 3, 256, 256)$ for MODIS and $(batch\_size, num\_channels = 2, 256, 256)$ for CMS and one for in-situ data $(seq\_len = 75, batch\_size, num\_feautres = 8)$

ResNet152. The differences between these models come from the number of layers included in these models, 18 and 152, respectively. The ResNet152 model also uses bottleneck blocks, whereas the ResNet18 uses basic blocks, both depicted in Figure 5.3.

MobileNet This CNN was designed for mobile vision applications (Howard et al., 2017). For experimentation, MobileNetv2 is used, which uses inverted residual blocks with a linear bottleneck (Sandler et al., 2018).

AlexNet Developed by Krizhevsky et al. (2012), AlexNet introduced non-linear activation functions and overlapped pooling. The implementation of multi-GPU training was introduced in this work as well.

Algorithm 4 Multimodal approach training (single batch)

```
Ensure: $X_{src}$ = tensor of$(seq\_len, batch\_size, num\_features)$
Ensure: $X_{tgt}$ = tensor of$(seq\_len, batch\_size, num\_features - 1)$
Ensure: $X_{modis}$ = tensor of$(batch\_size, in\_channels = 3, height = 256, width = 256)$
Ensure: $X_{cms}$ = tensor of$(batch\_size, in\_channels = 3, height = 256, width = 256)$
  $X_{src} \leftarrow time2vec(X_{src})$
  $X_{tgt} \leftarrow time2vec(X_{tgt})$
  $X_{src} \leftarrow transformer\_encode(X_{src})$
  $X_{src} \leftarrow transformer\_decode(X_{src}, X_{tgt}, masks)$
  $X_{src} \leftarrow avg\_pool(GeLU(conv\_1d(X_{src})))$
  $X_{modis} \leftarrow modis\_cnn(X_{modis})$
  $X_{cms} \leftarrow cms\_cnn(X_{cms})$
  $X = torch.concat(X_{src}, X_{modis}, X_{cms})$
  if $linear$ then
      $Y \leftarrow softmax(linear(X))$
  else
      for $ninoutput\_variables$ do
          $Y_n \leftarrow XGBoost_n(X)$
      end for
  end if
```

VGG Two versions of VGG were used for experimentation: VGG19 & VGG19 with batch norm (Simonyan and Zisserman, 2014). The architecture follows a standard CNN consisting of convolutional layers followed by hidden and fully connected layers.

GoogleNet GoogleNet architecture uses Inception modules that use multiple convolution sizes such as 1x1, 3x3 and 5x5 (Szegedy et al., 2015). The inception model is illustrated in Figure 5.4.

Algorithm 5 Multimodal approach testing (single batch))

```
Ensure: $X_{src}$ = tensor of$(seq\_len, batch\_size, num\_features)$
Ensure: $X_{tgt}$ = tensor of$(1, batch\_size, num\_features - 1)$
Ensure: $X_{modis}$ = tensor of$(batch\_size, in\_channels = 3, height = 256, width = 256)$
Ensure: $X_{cms}$ = tensor of$(batch\_size, in\_channels = 2, height = 256, width = 256)$
  $X_{src}, X_{tgt} \leftarrow time2vec(X_{src}), time2vec(X_{tgt})$
  $X_{src} \leftarrow tf\_encode(X_{src})$
  $outputs = [\,]$
  while $cur\_seq \neq seq\_len$ do
      $output \leftarrow tf\_decode(X_{src}[cur\_seq], X_{tgt}, masks)$
      $X_{tgt} \leftarrow output$
      $outputs.append(output)$
  end while
  $X_{modis} \leftarrow modis\_cnn(X_{modis})$
  $X_{cms} \leftarrow cms\_cnn(X_{cms})$
  $X = torch.concat(outputs, X_{modis}, X_{cms})$
  $outputs \leftarrow avg\_pool(GeLU(conv\_1d(outputs)))$
  $outputs \leftarrow softmax(linear(outputs))$
```

5.3.5 Additional Experiments

Coordinated Representation Approach for Detection

A second model is proposed for algal bloom detection with multimodal data that uses a coordinated representations approach with CMS data. The architecture is visualised in Figure 5.5. Using this model enables a user to detect algal blooms with an alternate data modality. Cases where in-situ data is corrupted and low in number enable the user to analyse satellite data for the detection and vice versa. The pseudocode is given in Algorithm 6.

Using Different MODIS Sensors for Detection

As mentioned previously in Section 5.3.2, satellites gather data using many sensors. Non-RGB related data could be used for further analysis. This experiment

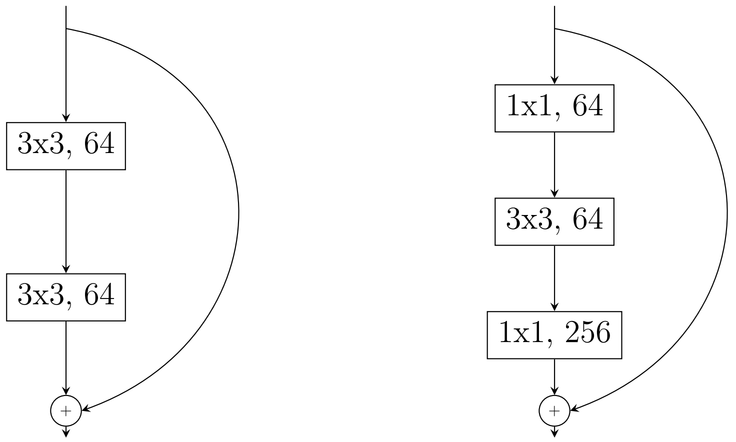

Figure 5.3: Left: ResNet Basic Block Right:ResNet Bottleneck Block (He et al., 2016). After each convolutional block ReLU is used as the activation function.

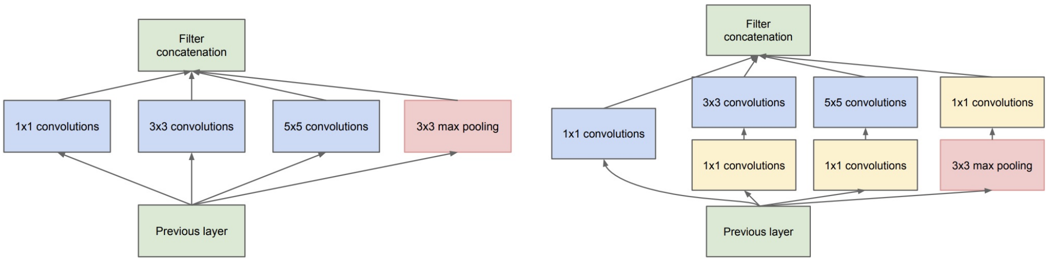

Figure 5.4: Left: Inception Basic Block Right:Inception Dimensionality Reduction Block (He et al., 2016)

compares RGB, False Color and NDWI for the task of early algal bloom detection.

5.4 Results

The predictions are made one to seven days into the future, given the observations on day x. 70% of data of TH1 buoy was used for training, 30% for validation. This location was chosen due to nutrient flow as it is located near the delta of the River Thames. The reason behind the location choice is to create a more

Algorithm 6 Representation training (single batch)

```
**Ensure:** $X_{sensor}$ = tensor of $(75, batch\_size, 8)$
**Ensure:** $X_{satellite}$ = tensor of $(batch\_size, 256, 256, 2)$
  $X_{sensor} \leftarrow linear(X_{sensor}.flatten())$
  $X_{satellite} \leftarrow conv\_model(X_{satellite})$
  $X_{satellite} \leftarrow linear(X_{satellite})$
  $loss \leftarrow euc\_dist(X_{sensor}, X_{satellite})$
  $loss.backward()$
```

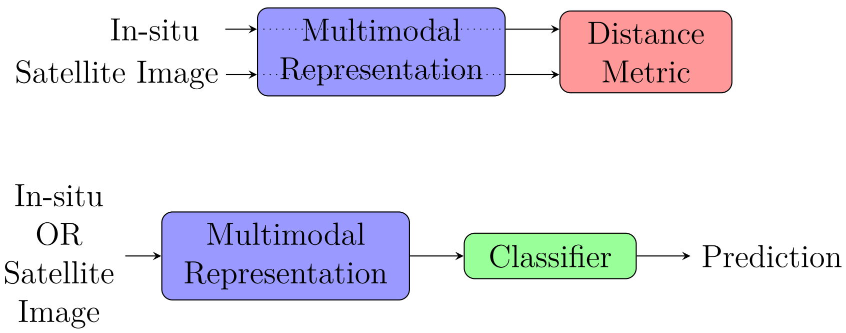
In-situ
Satellite Image
Multimodal Representation
Distance Metric
In-situ OR Satellite Image
Multimodal Representation
Classifier
Prediction

Figure 5.5: Proposed Multimodal Joint Representation

generalized model using the nutrient flows. A single location is used to observe if the model would be able to perform satisfactorily for locations with different properties, therefore testing the generalisability of the model. The other three sites are used for testing.

Figure 5.6 illustrates the MSE values for each model based on the number of days into the future. For all models, a hyperparameter search was done based on the prediction day. For the SVR model, a model was created for each predicted variable. For deep learning models, an Adam optimizer was used for this task with 200 epochs and earlystopping with a patience of 15 epochs (Kingma and Ba, 2014). The TF-Conv model in Chapter 4 is used for the in-situ data. The CNN models tested for MODIS and CMS data are: ResNet18, ResNet152, MobileNet_v2, VGG19, VGG19_bn, and AlexNet. Two comparisons are made, one with MSE for the three predicted variables and one with MAE to compare oxygen saturation percentages. The hyperparameters for the CNN models for the fusion approaches are given in Table 5.3. The results show that a single CNN architecture is not suitable for each prediction day. The hyperparameters for the XGBoost models for the fusion approaches are shown in Table 5.5 and Table 5.4. The same deduction about CNN hyperparameters could be made for the parameters of the XGBoost models.

The results of the first additional experiment, the coordinated representation approach, are presented in Figure 5.8 and Figure 5.9. The hyperparameters for the coordinated representations approach are given in Table 5.6. As the in-situ data is used as the ground truth for the representation that model learns the in-situ representation better than satellite data. In certain locations such as Dowsing, the replacement of data could be done and would result in a better performance than in-situ or multimodal approaches. The hyperparameters for the XGBoost model for each data modality are given in Table 5.7 and Table 5.8.

The results of the second additional experiment, using different MODIS sensors for detection, are visualised in Figure 5.10 and Figure 5.11. These figures show that in majority of the cases using RGB data is more suitable to detect HABs in these locations. The hyperparameters for the False Color and NDWI are shown in Table 5.9 and Table 5.10. The shift from the original hyperparameters are clear such that models like alexnet and resnet18 perform better than vgg19 in a number of cases, leading to an implication that the NDWI and False Color data contain different information than RGB data.

5.5 Discussion

Using only satellite data for tracking blooms results in differences in terms of explained variance based on year (Brivio et al., 2001). Therefore, using only a single modality might affect our prediction capability. From Figure 3.8, it can

Day	Model Type	MAE
1	Luong	5.182
2	KNR	7.954
3	KNR	7.952
4	XGB-Late	8.41
5	Fusion-Late	7.855
6	XGB-Late	8.300
7	XGB-Late	8.115

Table 5.2: MAE results for each day with the best performing model

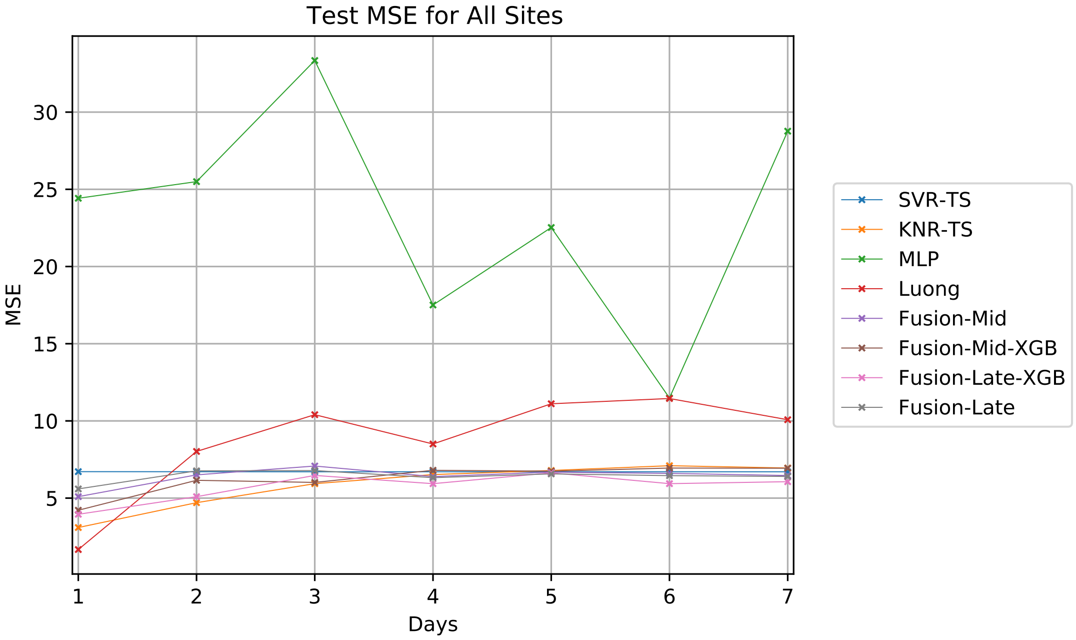

Figure 5.6: Mean MSE for test locations

be deduced that the Luong attention model is suitable for predicting the next day and k-NN is suitable for predicting two and three days ahead, using only in-situ observations. For the rest of the days, the most suitable model is one of the multimodal approaches proposed, either using the late fusion approach or transferring the learned representations from the late fusion approach and using XGBoost as the final classifier. Table 5.2 indicates an error rate between 7.855-8.3% for multimodal approaches for prediction days 4-7, which is on par with unimodal approaches for days 2 & 3. Using a feature representation approach

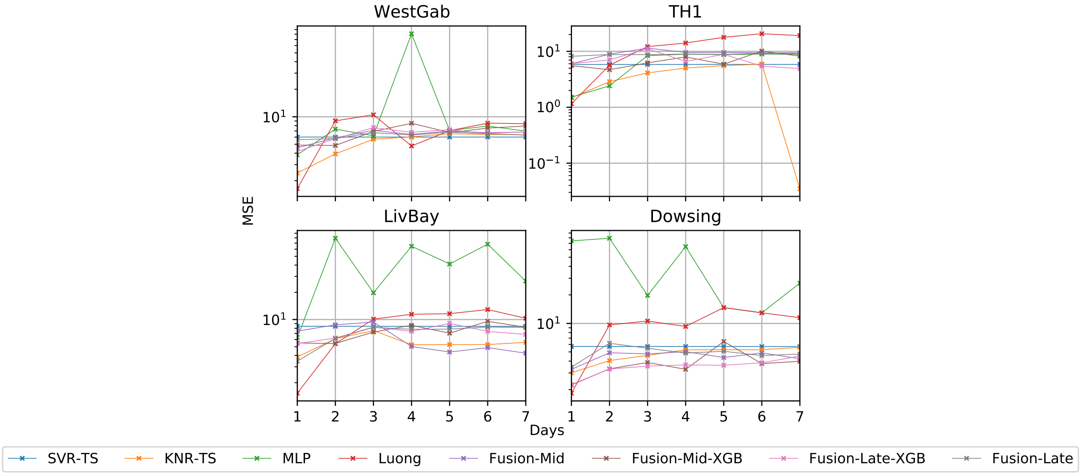

Figure 5.7: Mean MSE for each monitoring location

combined with XGBoost benefits the predictions.

Transferring learned parameters from different fusion approaches results in XGBoost models with slightly different hyperparameters. The XGBoost parameters in Tables 5.5 and 5.4 show that the learning rate hyperparameter *eta* is constant for the *o2conc* variable in the majority of the experiments and the generated trees are not as deep as other variables. The majority of the other hyperparameters are mostly constant in mid fusion. Hyperparameters such as *min_child_weight* and *subsample* vary depending on prediction day for both late and mid fusion XGBoost.

The proposed coordinated representation approach enables the use of multiple modalities. This approach can be used when the data quality of one modality is low. As the in-situ data is used as the ground truth, it was expected that the lowest error would be obtained from it, which can be observed in Figure 5.8. However, the in-situ model can be replaced by other approaches depending on the prediction day and the location, as illustrated in Figure 5.9. In the majority of the cases, VGG-19 with batch norm is the best performing CNN, which shows that batch norm enables the model to increase its generalisation capabilities. For

Day	batch_size	modis_model	copernicus_model	fusion_type	lr
1	16	mobilenet_v2	mobilenet_v2	mid	0.000238
	16	vgg19	vgg19	late	0.000343
2	32	vgg19	vgg19_bn	mid	0.003272
	32	resnet18	alexnet	late	0.004592
3	32	vgg19	mobilenet_v2	mid	0.000251
	16	resnet152	vgg19	late	0.000191
4	32	resnet18	alexnet	mid	0.001584
	16	resnet152	mobilenet_v2	late	0.0016
5	16	alexnet	vgg19	mid	0.002756
	32	vgg19_bn	mobilenet_v2	late	0.000138
6	32	alexnet	vgg19	mid	0.002278
	32	vgg19	vgg19	late	0.000932
7	32	vgg19	vgg19	mid	0.003165
	32	vgg19	vgg19	late	0.001475

Table 5.3: Hyperparameters for fusion models

in-situ data, the learning rate, *eta*, is small in the majority of the cases with a varying number of epochs. The rest of the parameters vary, resulting in different models for each prediction day. For satellite data, the generated trees are not deep and stay constant with varying *eta* values.

HABs can be detected using different data bands. On average, using RGB bands results in the best performance, followed by NDWI as indicated in Figure 5.10. This experiment entails that different modalities could be used for detection purposes. The results show that one modality is not best in all cases; therefore, various other options must be explored. Illustrated in Figure 5.11, WestGab's performance indicates that all three data band sets could be used for different days depending on the detection day. Using NDWI for detection at two and three days ahead at the Dowsing site shows comparable results to RGB and False Colour. The CNN types change depending on the type of data used and detection day.

Day	**fusion_type**	**eta**	**max_depth**	**min_child_weight**	**subsample**	**colsample_bytree**	**n_rounds**	**target_var**
1	late	1	3	5	1	0.7	500	o2conc
		0.01	3	0	0.5	0.5	500	sal
		0.01	10	0	0.5	0.5	1000	temp
2	late	1	5	3	1	0.5	500	o2conc
		1	3	3	1	0.7	500	sal
		0.1	10	0	1	0.5	500	temp
3	late	1	3	0	1	0.7	500	o2conc
		0.1	3	0	0.5	0.5	500	sal
		0.01	10	0	0.5	0.5	1000	temp
4	late	1	3	0	1	0.5	500	o2conc
		0.01	3	5	1	0.7	500	sal
		0.01	10	0	0.5	0.7	1000	temp
5	late	1	3	0	0.5	0.5	500	o2conc
		0.1	5	0	1	0.5	500	sal
		0.01	10	3	1	0.5	1000	temp
6	late	1	3	0	0.5	0.5	500	o2conc
		0.1	3	5	0.5	0.7	500	sal
		0.01	10	0	1	0.5	1000	temp
7	late	1	3	0	0.5	0.5	500	o2conc
		0.1	3	3	1	0.5	500	sal
		0.01	10	0	1	0.5	1000	temp

Table 5.4: Hyperparameters for the XGBoost model for the late fusion approach

Day	**fusion_type**	**eta**	**max_depth**	**min_child_weight**	**subsample**	**colsample_bytree**	**n_rounds**	**target_var**
1	mid	1	3	0	1	0.7	500	o2conc
		0.01	3	0	0.5	0.7	500	sal
		0.01	10	0	1	0.5	500	temp
2	mid	0.1	3	0	0.5	0.7	1000	o2conc
		0.01	3	3	1	0.5	500	sal
		0.01	10	3	1	0.5	1000	temp
3	mid	0.1	3	5	0.5	0.7	500	o2conc
		0.01	3	0	0.5	0.7	500	sal
		0.01	10	0	1	0.5	1000	temp
4	mid	1	3	0	0.5	0.5	500	o2conc
		0.1	3	5	0.5	0.7	500	sal
		0.01	10	0	1	0.5	500	temp
5	mid	1	3	0	0.5	0.5	500	o2conc
		0.01	3	3	0.5	0.5	500	sal
		0.01	10	0	0.5	0.7	500	temp
6	mid	1	3	0	0.5	0.5	500	o2conc
		0.1	3	3	0.5	0.5	500	sal
		0.01	10	0	0.5	0.5	1000	temp
7	mid	1	3	5	0.5	0.7	500	o2conc
		0.1	3	3	1	0.5	500	sal
		0.01	10	3	1	0.5	1000	temp

Table 5.5: Hyperparameters for the XGBoost model for the middle fusion approach

Day	batch_size	cnn_model	dims	lr
1	16	vgg19_bn	100	0.003509
2	16	resnet152	100	0.004236
3	16	vgg19_bn	100	0.003509
4	16	vgg19_bn	100	0.003509
5	16	vgg19_bn	100	0.003509
6	16	vgg19_bn	100	0.003509
7	32	vgg19	100	0.003507

Table 5.6: Hyperparameters for coordinated representation models

Day	data_type	eta	max_depth	min_child_weight	subsample	colsample_bytree	n_rounds	target_var
1	sensor	0.01	3	3	0.5	0.7	10000	o2conc
		0.01	10	5	0.5	0.7	500	sal
		0.01	5	3	1	0.5	10000	temp
2	sensor	0.01	5	5	0.5	0.5	1000	o2conc
		0.1	3	5	1	0.7	500	sal
		0.01	10	5	1	0.5	500	temp
3	sensor	0.01	3	5	1	0.5	10000	o2conc
		0.01	3	5	1	0.7	500	sal
		0.01	10	5	1	0.5	10000	temp
4	sensor	1	3	0	0.5	0.5	500	o2conc
		0.01	5	5	0.5	0.5	500	sal
		0.01	10	5	1	0.5	10000	temp
5	sensor	1	3	0	0.5	0.5	500	o2conc
		0.01	3	3	0.5	0.5	500	sal
		0.01	10	5	1	0.5	10000	temp
6	sensor	1	3	0	0.5	0.5	500	o2conc
		0.01	3	5	0.5	0.7	500	sal
		0.01	10	3	1	0.5	1000	temp
7	sensor	1	3	0	0.5	0.5	500	o2conc
		0.01	3	5	0.5	0.5	500	sal
		0.01	10	3	0.5	0.7	10000	temp

Table 5.7: Hyperparameters for the XGBoost model for the coordinated representation approach using sensory data

Day	**data_type**	**eta**	**max_depth**	**min_child_weight**	**subsample**	**colsample_bytree**	**n_rounds**	**target_var**
1	satellite	0.01	3	0	0.5	0.5	500	o2conc
		0.1	3	0	1	0.5	500	sal
		1	3	0	0.5	0.5	10000	temp
2	satellite	1	3	0	1	0.5	500	o2conc
		0.01	3	0	0.5	0.5	500	sal
		1	3	0	1	0.7	500	temp
3	satellite	1	3	0	1	0.5	500	o2conc
		0.1	3	0	1	0.5	500	sal
		1	3	3	0.5	0.5	10000	temp
4	satellite	1	3	0	1	0.5	500	o2conc
		0.1	3	0	1	0.5	500	sal
		1	3	0	0.5	0.5	10000	temp
5	satellite	1	3	0	0.5	0.5	500	o2conc
		0.1	3	0	1	0.7	500	sal
		1	3	3	0.5	0.5	500	temp
6	satellite	1	3	0	0.5	0.7	1000	o2conc
		0.01	3	3	0.5	0.5	500	sal
		1	3	3	0.5	0.5	1000	temp
7	satellite	1	3	0	0.5	0.5	500	o2conc
		0.1	3	0	1	0.5	500	sal
		1	3	0	0.5	0.5	1000	temp

Table 5.8: Hyperparameters for the XGBoost model for the coordinated representation approach using satellite data

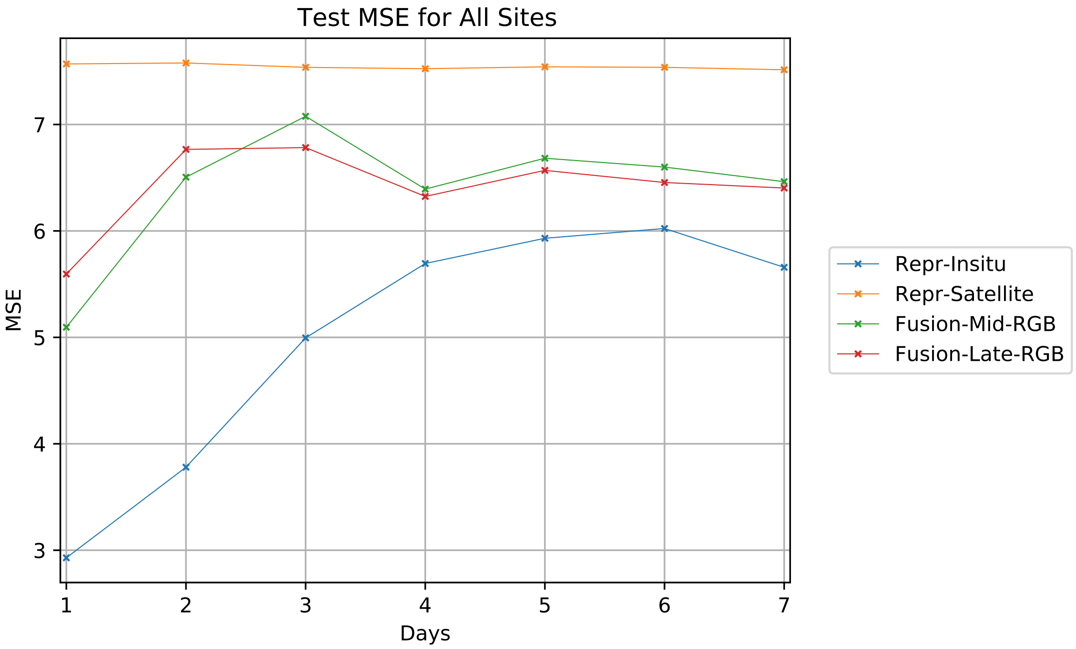

Figure 5.8: Mean MSE for test locations for coordinated representations approach

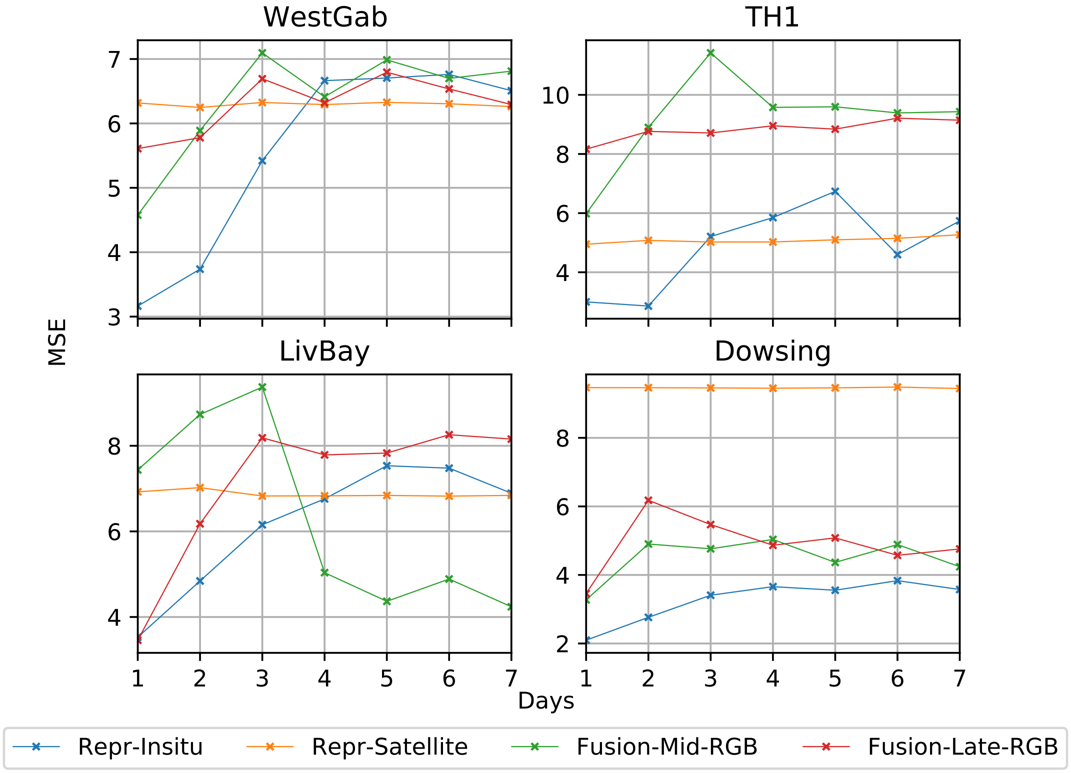

Figure 5.9: Mean MSE for each monitoring location using coordinated representations

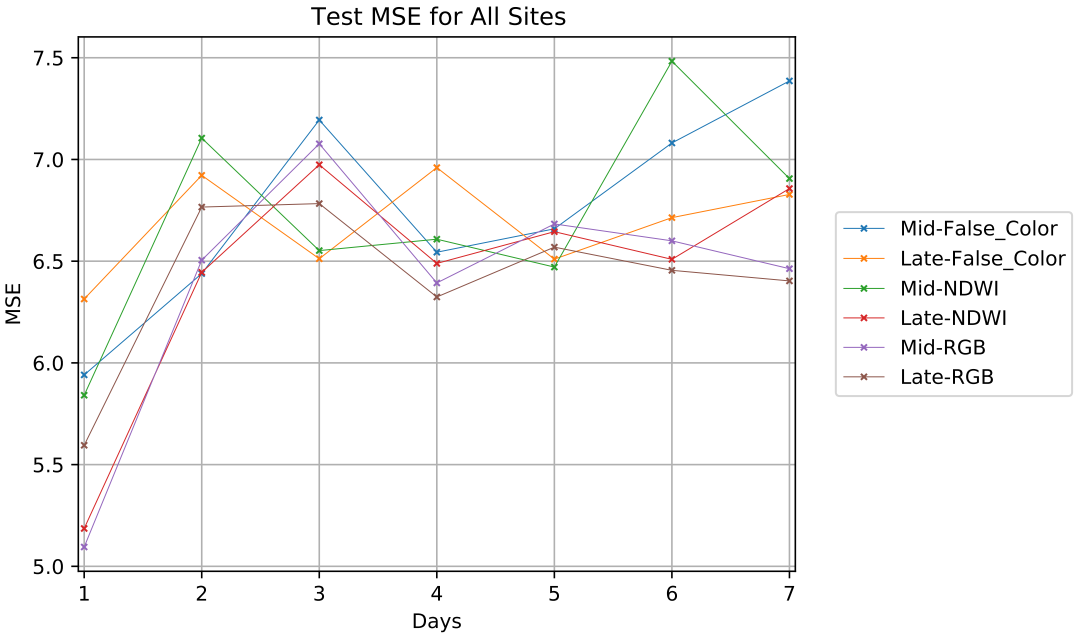

Figure 5.10: Mean MSE for test locations using different bands

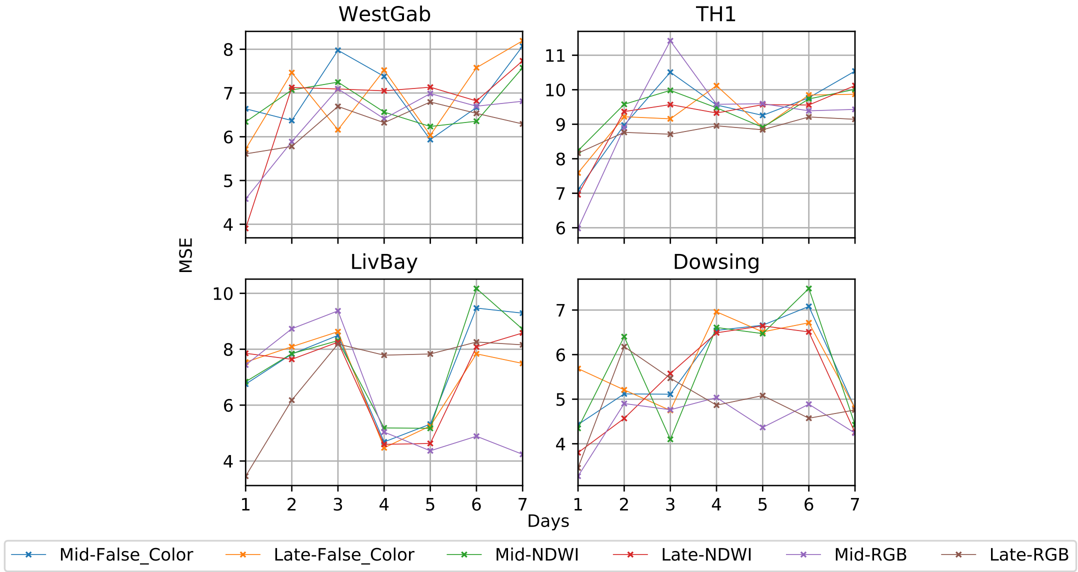

Figure 5.11: Mean MSE for each monitoring location using RGB, False Color and NDWI

Day	batch_size		modis_model	fusion_type	lr
1	32	vgg19	vgg19	mid	0.001185
	32	resnet152	alexnet	late	0.000158
2	32	vgg19_bn	vgg19	mid	0.003227
	16	vgg19_bn	mobilenet_v2	late	0.000212
3	16	resnet18	vgg19_bn	mid	0.003336
	16	resnet18	vgg19	late	0.001844
4	16	vgg19_bn	alexnet	mid	0.003438
	32	resnet152	vgg19	late	0.000974
5	32	alexnet	vgg19_bn	mid	0.003319
	16	alexnet	alexnet	late	0.00159
6	16	vgg19_bn	vgg19	mid	0.000894
	16	mobilenet_v2	alexnet	late	0.002594
7	32	alexnet	vgg19_bn	mid	0.002932
	32	resnet18	alexnet	late	0.004592

Table 5.9: Hyperparameters for the fusion model that uses False Color data as input

Day	**batch_size**		**modis_model**	**fusion_type**	**lr**
1	16	resnet18	alexnet	mid	0.00024
	32	vgg19	vgg19	late	0.000932
2	16	resnet18	alexnet	mid	0.001026
	16	alexnet	vgg19	late	0.003549
3	32	resnet18	alexnet	mid	0.001584
	16	mobilenet_v2	mobilenet_v2	late	0.000191
4	32	mobilenet_v2	vgg19	mid	0.001835
	32	resnet152	mobilenet_v2	late	0.000774
5	32	vgg19	alexnet	mid	0.002691
	16	vgg19_bn	mobilenet_v2	late	0.000212
6	32	alexnet	vgg19_bn	mid	0.003295
	32	mobilenet_v2	vgg19_bn	late	0.00185
7	16	vgg19	vgg19_bn	mid	0.002434
	32	mobilenet_v2	vgg19_bn	late	0.002416

Table 5.10: Hyperparameters for the fusion model that uses NDWI data as input

5.6 Conclusion

In this chapter, an additional approach regarding the detection of HABs is proposed. This approach uses the TF-Conv approach from Chapter 4 and uses two additional modalities gathered from satellite data. The main approach uses each modality separately in the initial step, fuses the intermediate states and calculates an output using three different approaches. These different approaches use a linear layer, an XGBoost regressor, or a weighted mean to calculate these values. The aim is to use each modality to extract essential information to make a more reliable model.

Two additional experiments were done; (i) testing the effect of different data types on model performance and (ii) replacing modalities using a coordinated approach in cases where data quality might be insufficient for predictions. It was observed that different data types perform well for different prediction days and observation sites. Using alternate modalities benefits predictions, it can be deduced that in-situ and satellite can be swapped depending on the day range and location. Other than prediction, model explainability and interpretability for multimodal approaches could be explored in the future.

Chapter 6

Conclusion

This chapter summarises the research undertaken for this project, outlines the limitations and work that can be undertaken for further research.

6.1 Summary

It was stated in Hypothesis 1 that the missingness of water quality data was MAR, and observed variables could be used to improve data quality via imputation. In Chapter 3, imputation for partially observed water quality data was discussed, using baselines that make different assumptions about the data distribution, aiming to answer Research Question 1. A novel approach to data imputation is introduced using self-attention and LSTMs to overcome the shortcoming of explainability in this process. An increase in performance was observed and each timestep's importance was was derived using the self-attention component. The use of self-attention aids in satisfying MAR properties as only the input is considered when attention weights are calculated.

In Hypothesis 2, it was stated that improving the labelling procedure using a contextual approach would benefit the model performance. In Chapter 4, it was discussed that the current approaches to algal bloom lack in areas such as generalisability for different locations, explainability and labelling of data, aiming to answer Research Question 2. A new solution for the labelling problem was proposed by including the contexts of observations. The generalisability issue was tackled using a representation learning approach to generate data and filter the generated data and test the trained model on locations with different properties. SHAP is used to tackle the explainability issue and aid domain experts using such models.

In Hypothesis 3, it was stated that various data modalities could be used simultaneously, leading to better performance. In Chapter 5, the possibility of using multiple modalities for HAB detection is discussed, aiming to answer Research Question 3. The current methods only utilise unimodal approaches and introduce multimodality to validate the model. An approach that simultaneously utilises multiple modalities for analysis using in-situ and satellite data is introduced. Further approaches are experimented with using coordinated approaches, aiming to replace the low quality data modality with a better one. The applicability of multimodal analysis is further explored by training models using various bands of satellite data.

6.2 Limitations

The Nature of Missingness As stated by Rubin (1978), it is impossible to obtain an exact value for a datum. If it were so, then the value would not be missing. If the nature of the missingness pattern were known, the imputed value would have no error, and the datum would not be missing. Therefore, the nature of the problem forces the user to make assumptions about the data to minimise the error in the imputation process.

Use of Proxy Variables The approach used for this work used dissolved oxygen as the target variable, which is not only created by phytoplankton but other organisms that contain chlorophyll in their cells. The same condition applies to the use of the chl-a variable. The sure way of determining harmful algal blooms is through measuring toxins in the water body or analysing algae species in samples in a lab which was collected from the water body. In the latter solution, the data gathered is not as frequent as in-situ monitoring and takes time; therefore, it may not be applicable for detection in short time windows.

Monitoring Locations The monitoring locations used for this thesis are not near populated areas; therefore, only in-situ or satellite observations could be used for this task. Water bodies closer to populated areas create opportunities for analysis, as new types of data sources, such as tweets, blogs and close-up images of bloom incidents can be gathered.

Generalisability of Models Due to ranging the variety of sensors at in-situ sites, it is a challenge to model the different behaviour observed at monitoring sites. The challenge is also apparent in models that use satellite data, as different works use data gathered by different sensors.

6.3 Future Work

Analysis of Collected Water Samples In-situ data can be further supported by analysing water samples gathered from monitoring locations. This analysis can give insight into the species found at the monitoring site at the collection date. The disadvantage of this type of analysis is that the frequency is low, and it creates the issue of fusing of time series datasets with different frequencies.

Social Media Data Twitter data could be incorporated into the model depending on the geolocation of the tweet and the possible study area. Tweets with image attachments which contain certain keywords could be included in a future iteration of the proposed model to improve its performance. However, social media data is limited to the HAB event happening at the time of posting. Therefore social media data cannot be used for early detection but for mitigation of a current event.

Citizen Science Programs Citizen programs are beneficial for monitoring water bodies around certain locations and gathering information related to blooms, harmful or not. By obtaining data from citizen science programs, more precise observations could be made, and models could be improved. This data source could be used in conjunction with social media data for populated areas.

Ship Traffic Data The LivBay site used for this thesis is near active ship routes, which may start/end their voyage at this destination. It is known that ballast water carried by ships can induce algal blooms if the newly introduced species can survive in this environment. Therefore it might be useful to include ship traffic data for similar locations that see heavy ship traffic.

Bibliography

Aas, K., Jullum, M., and Løland, A. (2019). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*.

Abdal, R., Qin, Y., and Wonka, P. (2019). Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4432–4441.

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

Aissia, M.-A. B., Chebana, F., and Ouarda, T. B. (2017). Multivariate missing data in hydrology–review and applications. *Advances in Water Resources*, 110:299–309.

Al Shehhi, M. R. and Kaya, A. (2021). Time series and neural network to forecast water quality parameters using satellite data. *Continental Shelf Research*, 231:104612.

Allan, F. and Wishart, J. (1930). A method of estimating the yield of a missing plot in field experimental work. *The Journal of Agricultural Science*, 20(3):399–406.

Almuhtaram, H., Zamyadi, A., and Hofmann, R. (2021). Machine learning for anomaly detection in cyanobacterial fluorescence signals. *Water Research*, page 117073.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

ALRashdi, R. and O'Keefe, S. (2019). Deep learning and word embeddings for tweet classification for crisis response. *arXiv preprint arXiv:1903.11024*.

Ames, A., Steiner, V., Liebold, E., Milz, S. A., and Eitniear, S. (2019). Perceptions of water-related environmental concerns in northwest ohio one year after a lake erie harmful algal bloom. *Environmental management*, 64(6):689–700.

Ananias, P. H. M. and Negri, R. G. (2021). Anomalous behaviour detection using one-class support vector machine and remote sensing images: a case study of algal bloom occurrence in inland waters. *International Journal of Digital Earth*, 14(7):921–942.

Anantrasirichai, N., Biggs, J., Albino, F., and Bull, D. (2019). A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. *Remote Sensing of Environment*, 230:111179.

Anderson, D. M., Fensin, E., Gobler, C. J., Hoeglund, A. E., Hubbard, K. A., Kulis, D. M., Landsberg, J. H., Lefebvre, K. A., Provoost, P., Richlen, M. L., et al. (2021). Marine harmful algal blooms (habs) in the united states: history, current status and future trends. *Harmful Algae*, 102:101975.

Anderson, D. M., Glibert, P. M., and Burkholder, J. M. (2002). Harmful algal blooms and eutrophication: nutrient sources, composition, and consequences. *Estuaries*, 25(4):704–726.

Apley, D. W. and Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086.

Arkhangelskaia, E. and Dutta, S. (2019). Whatcha lookin'at? deeplifting bert's attention in question answering. *arXiv preprint arXiv:1910.06431*.

Assembly, U. N. G. (2015). Transforming our world: The 2030 agenda for sustainable development.

Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.

Bansal, P., Deshpande, P., and Sarawagi, S. (2021). Missing value imputation on multidimensional time series. *arXiv preprint arXiv:2103.01600*.

Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., et al. (2012). Video in sentences out. *arXiv preprint arXiv:1204.2742*.

Beaulieu-Jones, B. K., Moore, J. H., and CONSORTIUM, P. R. O.-A. A. C. T. (2017). Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific symposium on biocomputing 2017*, pages 207–218. World Scientific.

Bechard, A. (2020). The economic impacts of harmful algal blooms on tourism: an examination of southwest florida using a spline regression approach. *Natural Hazards*, 104(1):593–609.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Billah, M., Waheed, S., and Hanifa, A. (2016). Stock market prediction using an improved training algorithm of neural network. In *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, pages 1–4. IEEE.

Bilogur, A. (2018). Missingno: a missing data visualization suite. *Journal of Open Source Software*, 3(22):547.

Binding, C., Greenberg, T., and Bukata, R. (2013). The meris maximum chlorophyll index; its merits and limitations for inland water algal bloom monitoring. *Journal of Great Lakes Research*, 39:100–107.

Binding, C., Greenberg, T., McCullough, G., Watson, S., and Page, E. (2018). An analysis of satellite-derived chlorophyll and algal bloom indices on lake winnipeg. *Journal of Great Lakes Research*, 44(3):436–446.

Blauw, A. N., Beninca, E., Laane, R. W., Greenwood, N., and Huisman, J. (2012). Dancing with the tides: fluctuations of coastal phytoplankton orchestrated by different oscillatory modes of the tidal cycle. *PLoS One*, 7(11).

Blauw, A. N., Benincà, E., Laane, R. W., Greenwood, N., and Huisman, J. (2018). Predictability and environmental drivers of chlorophyll fluctuations vary across different time scales and regions of the north sea. *Progress in Oceanography*, 161:1–18.

Boquet, G., Vicario, J. L., Morell, A., and Serrano, J. (2019). Missing data in traffic estimation: A variational autoencoder imputation method. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2882–2886. IEEE.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

Boyd, P. W. and Doney, S. C. (2003). The impact of climate change and feedback processes on the ocean carbon cycle. In *Ocean biogeochemistry*, pages 157–193. Springer.

Bregler, C., Covell, M., and Slaney, M. (1997). Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Bridle, J. (1989). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2.

Brivio, P., Giardino, C., and Zilioli, E. (2001). Determination of chlorophyll concentration changes in lake garda using an image-based radiative transfer code for landsat tm images. *International Journal of Remote Sensing*, 22(2-3):487–502.

Buczak, A. L. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176.

Burkholder, J., Libra, B., Weyer, P., Heathcote, S., Kolpin, D., Thorne, P. S., and Wichman, M. (2007a). Impacts of waste from concentrated animal feeding operations on water quality. *Environmental health perspectives*, 115(2):308–312.

Burkholder, J. M., Hallegraeff, G. M., Melia, G., Cohen, A., Bowers, H. A., Oldach, D. W., Parrow, M. W., Sullivan, M. J., Zimba, P. V., Allen, E. H., et al. (2007b). Phytoplankton and bacterial assemblages in ballast water of us military ships as a function of port of origin, voyage time, and ocean exchange practices. *Harmful Algae*, 6(4):486–518.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982.

Caillault, É. P., Lefebvre, A., Bigand, A., et al. (2020). Dynamic time warping-based imputation for univariate time series data. *Pattern Recognition Letters*, 139:139–147.

Cannizzaro, J. P., Hu, C., English, D. C., Carder, K. L., Heil, C. A., and Müller-Karger, F. E. (2009). Detection of karenia brevis blooms on the west florida shelf using in situ backscattering and fluorescence data. *Harmful Algae*, 8(6):898–909.

Cao, H., Han, L., and Li, L. (2022). A deep learning method for cyanobacterial harmful algae blooms prediction in taihu lake, china. *Harmful Algae*, 113:102189.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. (2018). Brits: bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, pages 6775–6785.

Cao, Z., Ma, R., Duan, H., Pahlevan, N., Melack, J., Shen, M., and Xue, K. (2020). A machine learning approach to estimate chlorophyll-a from landsat-8 measurements in inland lakes. *Remote Sensing of Environment*, 248:111974.

Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Chau, H. (2022). A red tide on march 13, 2022 on the southern side of lamma island, hong kong. [Online; accessed July 12, 2022].

Chellapilla, K., Puri, S., and Simard, P. (2006). High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft.

Chen, H., Lundberg, S., and Lee, S.-I. (2018). Hybrid gradient boosting trees and neural networks for forecasting operating room data. *arXiv preprint arXiv:1801.07384*.

Chen, K., Zhou, Y., and Dai, F. (2015a). A lstm-based method for stock returns prediction: A case study of china stock market. In *2015 IEEE international conference on big data (big data)*, pages 2823–2824. IEEE.

Chen, L.-J., Ho, Y.-H., Hsieh, H.-H., Huang, S.-T., Lee, H.-C., and Mahajan, S. (2017). Adf: An anomaly detection framework for large-scale pm2. 5 sensing systems. *IEEE Internet of Things Journal*, 5(2):559–570.

Chen, Q., Guan, T., Yun, L., Li, R., and Recknagel, F. (2015b). Online forecasting chlorophyll a concentrations by an auto-regressive integrated moving average model: Feasibilities and potentials. *Harmful Algae*, 43:58–65.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chen, X., Fu, Y., and Zhou, H. (2021). Conv-attention model based on multivariate time series prediction: The cyanobacteria bloom case. In *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, pages 1018–1023.

Chen, Y., Song, L., Liu, Y., Yang, L., and Li, D. (2020). A review of the artificial neural network models for water quality prediction. *Applied Sciences*, 10(17):5776.

Chia, W. Y., Tang, D. Y. Y., Khoo, K. S., Lup, A. N. K., and Chew, K. W. (2020). Nature's fight against plastic pollution: Algae for plastic biodegradation and bioplastics production. *Environmental Science and Ecotechnology*, page 100065.

Chislock, M. F., Doster, E., Zitomer, R. A., and Wilson, A. E. (2013). Eutrophication: causes, consequences, and controls in aquatic ecosystems. *Nature Education Knowledge*, 4(4):10.

Chivers, B. D., Wallbank, J., Cole, S. J., Sebek, O., Stanley, S., Fry, M., and Leontidis, G. (2020). Imputation of missing sub-hourly precipitation data in a large sensor network: A machine learning approach. *Journal of Hydrology*, 588:125126.

Cho, H., Choi, U., and Park, H. (2018). Deep learning application to time-series prediction of daily chlorophyll-a concentration. *WIT Trans. Ecol. Environ.*, 215:157–63.

Cho, H. and Park, H. (2019). Merged-lstm and multistep prediction of daily chlorophyll-a concentration for algal bloom forecast. In *IOP Conference Series: Earth and Environmental Science*, volume 351, page 012020. IOP Publishing.

Cho, S., Lim, B., Jung, J., Kim, S., Chae, H., Park, J., Park, S., and Park, J. K. (2014). Factors affecting algal blooms in a man-made lake and prediction using an artificial neural network. *Measurement*, 53:224–233.

Choudhury, A. and Kosorok, M. R. (2020). Missing data imputation for classification problems. *arXiv preprint arXiv:2002.10709*.

Christoudias, C. M., Saenko, K., Morency, L.-P., and Darrell, T. (2006). Co-adaptation of audio-visual speech and gesture classifiers. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 84–91.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Clark, J. M., Schaeffer, B. A., Darling, J. A., Urquhart, E. A., Johnston, J. M., Ignatius, A. R., Myer, M. H., Loftin, K. A., Werdell, P. J., and Stumpf, R. P. (2017). Satellite monitoring of cyanobacterial harmful algal bloom frequency in recreational waters and drinking water sources. *Ecological indicators*, 80:84–95.

Colas, F. and Brazdil, P. (2006). Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Cruz, R. C., Reis Costa, P., Vinga, S., Krippahl, L., and Lopes, M. B. (2021). A review of recent machine learning advances for forecasting harmful algal blooms and shellfish contamination. *Journal of Marine Science and Engineering*, 9(3):283.

Davis, T. W., Berry, D. L., Boyer, G. L., and Gobler, C. J. (2009). The effects of temperature and nutrients on the growth and dynamics of toxic and non-toxic strains of microcystis during cyanobacteria blooms. *Harmful algae*, 8(5):715–725.

De Waele, G., Clauwaert, J., Menschaert, G., and Waegeman, W. (2022). Cpg transformer for imputation of single-cell methylomes. *Bioinformatics*, 38(3):597–603.

Deena, S. and Galata, A. (2009). Speech-driven facial animation using a shared gaussian process latent variable model. In *International Symposium on Visual Computing*, pages 89–100. Springer.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Derot, J., Yajima, H., and Jacquet, S. (2020). Advances in forecasting harmful algal blooms using machine learning models: A case study with planktothrix rubescens in lake geneva. *Harmful Algae*, 99:101906.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dino, H. I. and Abdulrazzaq, M. B. (2019). Facial expression classification based on svm, knn and mlp classifiers. In *2019 International Conference on Advanced Science and Engineering (ICOASE)*, pages 70–75. IEEE.

Directive, S. F. (2000). Directive 2000/60/ec of the european parliament and of the council of 23 october 2000 establishing a framework for community action in the field of water policy" or, in short, the eu water framework directive.

Directive, S. F. (2008). Directive 2008/56/ec of the european parliament and of the council.

Dodds, W. K., Bouska, W. W., Eitzmann, J. L., Pilger, T. J., Pitts, K. L., Riley, A. J., Schloesser, J. T., and Thornbrugh, D. J. (2009). Eutrophication of us freshwaters: analysis of potential economic damages.

Dogru, N. and Subasi, A. (2018). Traffic accident detection using random forest classifier. In *2018 15th learning and technology conference (L&T)*, pages 40–45. IEEE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dyson, K. and Huppert, D. D. (2010). Regional economic impacts of razor clam beach closures due to harmful algal blooms (habs) on the pacific coast of washington. *Harmful Algae*, 9(3):264–271.

El-Habashi, A., Ioannou, I., Tomlinson, M. C., Stumpf, R. P., and Ahmed, S. (2016). Satellite retrievals of karenia brevis harmful algal blooms in the west florida shelf using neural networks and comparisons with other techniques. *Remote Sensing*, 8(5):377.

Enders, C. K. (2010). *Applied missing data analysis*. Guilford press.

Falconer, I. R. (1999). An overview of problems caused by toxic blue–green algae (cyanobacteria) in drinking and recreational water. *Environmental Toxicology: An International Journal*, 14(1):5–12.

Falconer, I. R., Burch, M. D., Steffensen, D. A., Choice, M., and Coverdale, O. R. (1994). Toxicity of the blue-green alga (cyanobacterium) microcystis aeruginosa in drinking water to growing pigs, as an animal model for human injury and risk assessment. *Environmental toxicology and Water quality*, 9(2):131–139.

Fayne, J., Bolten, J., Lakshmi, V., and Ahamed, A. (2017). Optical and physical methods for mapping flooding with satellite imagery. In *Remote Sensing of Hydrological Extremes*, pages 83–103. Springer.

Fidler, S., Sharma, A., and Urtasun, R. (2013). A sentence is worth a thousand pixels. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1995–2002.

Flörke, M., Kynast, E., Bärlund, I., Eisner, S., Wimmer, F., and Alcamo, J. (2013). Domestic and industrial water uses of the past 60 years as a mirror of socio-economic development: A global simulation study. *Global Environmental Change*, 23(1):144–156.

Folguera, L., Zupan, J., Cicerone, D., and Magallanes, J. F. (2015). Self-organizing maps for imputation of missing data in incomplete data matrices. *Chemometrics and Intelligent Laboratory Systems*, 143:146–151.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model.

Fthenakis, V. and Kim, H. C. (2010). Life-cycle uses of water in us electricity generation. *Renewable and Sustainable Energy Reviews*, 14(7):2039–2048.

Fu, F. X., Tatters, A. O., and Hutchins, D. A. (2012). Global change and the future of harmful algal blooms in the ocean. *Marine Ecology Progress Series*, 470:207–233.

Gao, H., Mao, J., Zhou, J., Huang, Z., Wang, L., and Xu, W. (2015). Are you talking to a machine? dataset and methods for multilingual image question answering. *arXiv preprint arXiv:1505.05612*.

Gao, S. and Wang, Y. (2022). Explainable deep learning powered building risk assessment model for proactive hurricane response. *Risk analysis*.

Garcia, H. E. and Gordon, L. I. (1992). Oxygen solubility in seawater: Better fitting equations. *Limnology and oceanography*, 37(6):1307–1312.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J. (2002). Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143.

Gerssen, A., van Olst, E. H., Mulder, P. P., and de Boer, J. (2010). In-house validation of a liquid chromatography tandem mass spectrometry method for the analysis of lipophilic marine toxins in shellfish using matrix-matched calibration. *Analytical and bioanalytical chemistry*, 397(7):3079–3088.

Ghatkar, J. G., Singh, R. K., and Shanmugam, P. (2019). Classification of algal bloom species from remote sensing data using an extreme gradient boosted decision tree model. *International Journal of Remote Sensing*, 40(24):9412–9438.

Glibert, P. M. and Burkholder, J. M. (2011). Harmful algal blooms and eutrophication:“strategies” for nutrient uptake and growth outside the redfield comfort zone. *Chinese Journal of Oceanology and Limnology*, 29(4):724–738.

Gokaraju, B., Durbha, S. S., King, R. L., and Younan, N. H. (2011). A machine learning based spatio-temporal data mining approach for detection of harmful algal blooms in the gulf of mexico. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 4(3):710–720.

Gokul, E. A., Raitsos, D. E., Gittings, J. A., Alkawri, A., and Hoteit, I. (2019). Remotely sensing harmful algal blooms in the red sea. *PloS one*, 14(4):e0215463.

Gondara, L. and Wang, K. (2018). Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 260–272. Springer.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gu, H., Wu, Y., Lü, S., Lu, D., Tang, Y. Z., and Qi, Y. (2021). Emerging harmful algal bloom species over the last four decades in china. *Harmful Algae*, page 102059.

Guallar, C., Delgado, M., Diogene, J., and Fernandez-Tejedor, M. (2016). Artificial neural network approach to population dynamics of harmful algal blooms in alfacs bay (nw mediterranean): Case studies of karlodinium and pseudo-nitzschia. *Ecological modelling*, 338:37–50.

Güler, C., Thyne, G. D., McCray, J. E., and Turner, K. A. (2002). Evaluation of graphical and multivariate statistical methods for classification of water chemistry data. *Hydrogeology journal*, 10(4):455–474.

Guo, J., Dong, Y., and Lee, J. H. (2020). A real time data driven algal bloom risk forecast system for mariculture management. *Marine Pollution Bulletin*, 161:111731.

Guo, W., Wang, J., and Wang, S. (2019). Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394.

Gurban, M., Thiran, J.-P., Drugman, T., and Dutoit, T. (2008). Dynamic modality weighting for multi-stream hmms inaudio-visual speech recognition. In *Proceedings of the 10th international conference on Multimodal interfaces*, pages 237–240.

Hallegraeff, G. M. (1998). Transport of toxic dinoflagellates via ships ballast water: bioeconomic risk assessment and efficacy of possible ballast water management strategies. *Marine Ecology Progress Series*, 168:297–309.

Hallegraeff, G. M. (2010). Ocean climate change, phytoplankton community responses, and harmful algal blooms: a formidable predictive challenge 1. *Journal of phycology*, 46(2):220–235.

Han, J. and Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*, pages 195–201. Springer.

Hara, Y., Fukuyama, Y., Murakami, K., Iizaka, T., and Matsui, T. (2020). Fault detection of hydroelectric generators using isolation forest. In *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 864–869. IEEE.

Harper, D. (1992). What is eutrophication? In *Eutrophication of Freshwaters*, pages 1–28. Springer.

Harrou, F., Dairi, A., Sun, Y., and Kadri, F. (2018). Detecting abnormal ozone measurements with a deep learning-based strategy. *IEEE Sensors Journal*, 18(17):7222–7232.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Heaton, J. B., Polson, N. G., and Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12.

Heffernan, J., Barry, J., Devlin, M., and Fryer, R. (2010). A simulation tool for designing nutrient monitoring programmes for eutrophication assessments. *Environmetrics: The official journal of the International Environmetrics Society*, 21(1):3–20.

Hill, P. R., Kumar, A., Temimi, M., and Bull, D. R. (2020). Habnet: machine learning, remote sensing-based detection of harmful algal blooms. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:3229–3239.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hong, S. M., Baek, S.-S., Yun, D., Kwon, Y.-H., Duan, H., Pyo, J., and Cho, K. H. (2021). Monitoring the vertical distribution of habs using hyperspectral imagery and deep learning models. *Science of The Total Environment*, 794:148592.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, C., Feng, L., and Guan, Q. (2020). A machine learning approach to estimate surface chlorophyll a concentrations in global oceans from satellite measurements. *IEEE Transactions on Geoscience and Remote Sensing*, 59(6):4590–4607.

Hu, C., Muller-Karger, F. E., Taylor, C. J., Carder, K. L., Kelble, C., Johns, E., and Heil, C. A. (2005). Red tide detection and tracing using modis fluorescence data: A regional example in sw florida coastal waters. *Remote Sensing of Environment*, 97(3):311–321.

Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., and Darrell, T. (2016). Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564.

Huang, J., Gao, J., and Zhang, Y. (2015). Combination of artificial neural network and clustering techniques for predicting phytoplankton biomass of lake poyang, china. *Limnology*, 16(3):179–191.

Huang, X., Tan, H., Lin, G., and Tian, Y. (2018). A lstm-based bidirectional translation model for optimizing rare words and terminologies. In *2018 international conference on artificial intelligence and big data (ICAIBD)*, pages 185–189. IEEE.

Hunt, A. J. and Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

Huynh, N. H., Böer, G., and Schramm, H. (2022). Self-attention and generative adversarial networks for algae monitoring. *European Journal of Remote Sensing*, 55(1):10–22.

Ismail, N. and Malik, O. A. (2021). Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. *Information Processing in Agriculture.*

Izadi, M., Sultan, M., Kadiri, R. E., Ghannadi, A., and Abdelmohsen, K. (2021).

A remote sensing and machine learning-based approach to forecast the onset of harmful algal bloom. *Remote Sensing*, 13(19):3863.

Jabreel, M. and Moreno, A. (2019). A deep learning-based approach for multi-label emotion classification in tweets. *Applied Sciences*, 9(6):1123.

Jadhav, A., Pramod, D., and Ramanathan, K. (2019). Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, 33(10):913–933.

Jaeger, M. (2006). On testing the missing at random assumption. In *European Conference on Machine Learning*, pages 671–678. Springer.

Jaques, N., Taylor, S., Sano, A., and Picard, R. (2015). Multi-task, multi-kernel learning for estimating individual wellbeing. In *Proc. NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*, volume 898, page 3.

Jiang, H., He, Z., Ye, G., and Zhang, H. (2020). Network intrusion detection based on pso-xgboost model. *IEEE Access*, 8:58392–58401.

Jiang, P., Liu, X., Zhang, J., and Yuan, X. (2016). A framework based on hidden markov model with adaptive weighting for microcystin forecasting and early-warning. *Decision Support Systems*, 84:89–103.

Jiang, Q.-Y. and Li, W.-J. (2017). Deep cross-modal hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3232–3240.

Jiang, Y., Li, C., Song, H., and Wang, W. (2022). Deep learning model based on urban multi-source data for predicting heavy metals (cu, zn, ni, cr) in industrial sewer networks. *Journal of Hazardous Materials*, 432:128732.

Jin, Q. and Liang, J. (2016). Video description generation using audio and visual cues. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 239–242.

Jin, X., Yu, X., Wang, X., Bai, Y., Su, T., and Kong, J. (2020). Prediction for time series with cnn and lstm. In *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)*, pages 631–641. Springer.

John, H. and Naaz, S. (2019). Credit card fraud detection using local outlier factor and isolation forest. *Int. J. Comput. Sci. Eng.*, 7(4):1060–1064.

Jun, E., Mulyadi, A. W., and Suk, H.-I. (2019). Stochastic imputation and uncertainty-aware attention to ehr for mortality prediction. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Justice, C. O., Vermote, E., Townshend, J. R., Defries, R., Roy, D. P., Hall, D. K., Salomonson, V. V., Privette, J. L., Riggs, G., Strahler, A., et al. (1998). The moderate resolution imaging spectroradiometer (modis): Land remote sensing for global change research. *IEEE transactions on geoscience and remote sensing*, 36(4):1228–1249.

Kahru, M. and Mitchell, B. G. (2008). Ocean color reveals increased blooms in various parts of the world. *Eos, Transactions American Geophysical Union*, 89(18):170–170.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709.

Kamilaris, A. and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90.

Kang, L., Gong, Y., Yang, C., Luo, J., Luo, Q., and Gao, Y. (2010). Marine phytoplankton recognition using hybrid classification methods. In *2010 4th International Conference on Bioinformatics and Biomedical Engineering*, pages 1–5. IEEE.

Karlson, B., Andersen, P., Arneborg, L., Cembella, A., Eikrem, W., John, U., West, J. J., Klemm, K., Kobos, J., Lehtinen, S., et al. (2021). Harmful algal blooms and their effects in coastal seas of northern europe. *Harmful Algae*, page 101989.

Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., and Brubaker, M. (2019). Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*.

Kelly, A. and Linda, G. (1996). Algae in aquatic ecosystems. *Natural Resources Facts Fact Sheet*, (96-4).

Khalifeloo, M. H., Mohammad, M., and Heydari, M. (2015). Multiple imputation for hydrological missing data by using a regression method (klang river basin). *International Journal of Researchin Engineering and Technology*, 4(06).

Khan, R. M., Salehi, B., Mahdianpari, M., Mohammadimanesh, F., Mountrakis, G., and Quackenbush, L. J. (2021). A meta-analysis on harmful algal bloom (hab) detection and monitoring: A remote sensing perspective. *Remote Sensing*, 13(21):4347.

Kim, H.-G., Jang, G.-J., Choi, H.-J., Kim, M., Kim, Y.-W., and Choi, J. (2017). Recurrent neural networks with missing information imputation for medical examination data prediction. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 317–323. IEEE.

Kim, M., Baek, S., Ligaray, M., Pyo, J., Park, M., and Cho, K. H. (2015). Comparative studies of different imputation methods for recovering streamflow observation. *Water*, 7(12):6847–6860.

Kim, S. (2016). A multiple process univariate model for the prediction of chlorophyll-a concentration in river systems. In *Annales de Limnologie-International Journal of Limnology*, volume 52, pages 137–150. EDP Sciences.

Kim, S. M., Shin, J., Baek, S., and Ryu, J.-H. (2019a). U-net convolutional neural network model for deep red tide learning using goci. *Journal of Coastal Research*, 90(SI):302–309.

Kim, T., Shin, J., Lee, D., Kim, Y., Na, E., Park, J.-h., Lim, C., and Cha, Y. (2022). Simultaneous feature engineering and interpretation: Forecasting harmful algal blooms using a deep learning approach. *Water Research*, 215:118289.

Kim, W., Cho, W., Choi, J., Kim, J., Park, C., and Choo, J. (2019b). A comparison of the effects of data imputation methods on model performance. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 592–599. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson,

O. (2020). Captum: A unified and generic model interpretability library for pytorch.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Kuznetsova, P., Ordonez, V., Berg, A., Berg, T., and Choi, Y. (2012). Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 359–368.

Lai, P. L. and Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377.

Le, X.-H., Ho, H. V., Lee, G., and Jung, S. (2019). Application of long short-term memory (lstm) neural network for flood forecasting. *Water*, 11(7):1387.

Lebret, R., Pinheiro, P., and Collobert, R. (2015). Phrase-based image captioning. In *International Conference on Machine Learning*, pages 2085–2094. PMLR.

Lee, D., Kim, J., Moon, W.-J., and Ye, J. C. (2019). Collagan: Collaborative gan for missing image data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2487–2496.

Lee, K., Lee, I., and Lee, S. (2018). Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135.

Lee, S. and Lee, D. (2018). Improved prediction of harmful algal blooms in four major south korea's rivers using deep learning models. *International journal of environmental research and public health*, 15(7):1322.

Lehman, P., Boyer, G., Hall, C., Waller, S., and Gehrts, K. (2005). Distribution and toxicity of a new colonial microcystis aeruginosa bloom in the san francisco bay estuary, california. *Hydrobiologia*, 541(1):87–99.

Leonardo, P., Denga, Y., Tronczynski, J., Bignert, A., Mehtonen, J., Hylland, K., Angelidis, M., Davies, I., Velikova, V., Law, R., Herat, B., Piha, H., Hanke, G., Batty, J., Dachs, J., Duffek, A., Lepom, P., Vethaak, D., and Roose, P. (2011). *Marine Strategy Framework Directive : task group 8 report : contaminants and pollution effects, April 2010.* Publications Office.

Lepock, J. R. (2005). How do cells respond to their thermal environment? *International journal of hyperthermia*, 21(8):681–687.

Li, N. and Chen, Z. (2018). Image captioning with visual-semantic lstm. In *IJCAI*, pages 793–799.

Li, X., Yu, J., Jia, Z., and Song, J. (2014). Harmful algal blooms prediction with machine learning models in tolo harbour. In *2014 International Conference on Smart Computing*, pages 245–250. IEEE.

Lin, S. (2017). *Climate change and algal blooms.* Michigan State University.

Lin, S., Novitski, L. N., Qi, J., and Stevenson, R. J. (2018). Landsat tm/etm+ and machine-learning algorithms for limnological studies and algal bloom management of inland lakes. *Journal of Applied Remote Sensing*, 12(2):026003.

Little, R. J. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American statistical Association*, 83(404):1198–1202.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE.

Liu, J., Lin, H., Liu, X., Xu, B., Ren, Y., Diao, Y., and Yang, L. (2019a). Transformer-based capsule network for stock movement prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 66–73.

Liu, Q., Zhou, F., Hang, R., and Yuan, X. (2017). Bidirectional-convolutional lstm based spectral-spatial feature learning for hyperspectral image classification. *Remote Sensing*, 9(12):1330.

Liu, S., Liao, G., and Ding, Y. (2018). Stock transaction prediction modeling and analysis based on lstm. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 2787–2790. IEEE.

Liu, Y., Yu, R., Zheng, S., Zhan, E., and Yue, Y. (2019b). Naomi: Non-autoregressive multiresolution sequence imputation. *arXiv preprint arXiv:1901.10946*.

Lui, G. C., Li, W. K., Leung, K. M., Lee, J. H., and Jayawardena, A. W. (2007). Modelling algal blooms using vector autoregressive model with exogenous variables and long memory filter. *Ecological modelling*, 200(1-2):130–138.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Luo, Y., Cai, X., Zhang, Y., Xu, J., et al. (2018). Multivariate time series imputation with generative adversarial networks. *Advances in Neural Information Processing Systems*, 31.

Luo, Y., Yang, K., Yu, Z., Chen, J., Xu, Y., Zhou, X., and Yang, Y. (2017). Dynamic monitoring and prediction of dianchi lake cyanobacteria outbreaks in the context of rapid urbanization. *Environmental Science and Pollution Research*, 24(6):5335–5348.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Mallin, M. A., McIver, M. R., Robuck, A. R., and Dickens, A. K. (2015). Industrial swine and poultry production causes chronic nutrient and fecal microbial stream pollution. *Water, Air, & Soil Pollution*, 226(12):1–13.

Mansimov, E., Parisotto, E., Ba, J. L., and Salakhutdinov, R. (2015). Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Mayr, A., Klambauer, G., Unterthiner, T., and Hochreiter, S. (2016). Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322.

McClain, C. R., Feldman, G. C., and Hooker, S. B. (2004). An overview of the seawifs project and strategies for producing a climate research quality global ocean bio-optical time series. *Deep Sea Research Part II: Topical Studies in Oceanography*, 51(1-3):5–42.

McCoy, J. T., Kroon, S., and Auret, L. (2018). Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, 51(21):141–146.

McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. (2000). Tracking groups of people. *Computer vision and image understanding*, 80(1):42–56.

Mehrabian, A. and Pahlevan, N. (2019). Identifying anomalies in surface water quality using isolation forest. In *AGU Fall Meeting Abstracts*, volume 2019, pages EP54C–09.

Mehrotra, K. G., Mohan, C. K., and Huang, H. (2017). *Anomaly detection principles and algorithms*, volume 1. Springer.

Mellios, N., Moe, S. J., and Laspidou, C. (2020). Machine learning approaches for predicting health risk of cyanobacterial blooms in northern european lakes. *Water*, 12(4):1191.

Mikolov, T. et al. (2012). Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 80:26.

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Morvant, E., Habrard, A., and Ayache, S. (2014). Majority vote of diverse classifiers for late fusion. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 153–162. Springer.

Mulia, I. E., Asano, T., and Tkalich, P. (2015). Retrieval of missing values in water temperature series using a data-driven model. *Earth Science Informatics*, 8(4):787–798.

Muttil, N. and Chau, K.-w. (2006). Neural network and genetic programming for modelling coastal algal blooms. *International Journal of Environment and Pollution*, 28(3-4):223–238.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21.

Narvekar, M. and Fargose, P. (2015). Daily weather forecasting using artificial neural network.

Nascita, A., Montieri, A., Aceto, G., Ciuonzo, D., Persico, V., and Pescapé, A. (2021). Xai meets mobile traffic classification: Understanding and improving multimodal deep learning architectures. *IEEE Transactions on Network and Service Management*, 18(4):4225–4246.

Nations, U. (2021). United nations world water development report 2021.

Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *ICML*.

Nieh, C., Dorevitch, S., Liu, L. C., and Jones, R. M. (2014). Evaluation of imputation methods for microbial surface water quality studies. *Environmental Science: Processes & Impacts*, 16(5):1145–1153.

Noulas, A., Englebienne, G., and Krose, B. J. (2011). Multimodal speaker diarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):79–93.

Olenina, I., Wasmund, N., Hajdu, S., Jurgensone, I., Gromisz, S., Kownacka, J., Toming, K., Vaiciūtė, D., and Olenin, S. (2010). Assessing impacts of invasive phytoplankton: the baltic sea case. *Marine Pollution Bulletin*, 60(10):1691–1700.

Osman, M. S., Abu-Mahfouz, A. M., and Page, P. R. (2018). A survey on data imputation techniques: Water distribution system as a use case. *IEEE Access*, 6:63279–63291.

Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E. H., and Freeman, W. T. (2016). Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413.

O'Neil, J. M., Davis, T. W., Burford, M. A., and Gobler, C. J. (2012). The rise of harmful cyanobacteria blooms: the potential roles of eutrophication and climate change. *Harmful algae*, 14:313–334.

Paerl, H. W. and Huisman, J. (2008). Blooms like it hot. *Science*, 320(5872):57–58.

Pahlevan, N., Smith, B., Schalles, J., Binding, C., Cao, Z., Ma, R., Alikas, K., Kangro, K., Gurlin, D., Hà, N., et al. (2020). Seamless retrievals of chlorophyll-a from sentinel-2 (msi) and sentinel-3 (olci) in inland and coastal waters: A machine-learning approach. *Remote Sensing of Environment*, 240:111604.

Papadimitriou, S., Sun, J., Faloutos, C., and Philip, S. Y. (2013). Dimensionality reduction and filtering on time series sensor streams. *Managing and Mining Sensor Data*, pages 103–141.

Park, J., Lee, H., Park, C. Y., Hasan, S., Heo, T.-Y., and Lee, W. H. (2019). Algal morphological identification in watersheds for drinking water supply using neural architecture search for convolutional neural network. *Water*, 11(7):1338.

Park, J., Lee, W. H., Kim, K. T., Park, C. Y., Lee, S., and Heo, T.-Y. (2022). Interpretation of ensemble learning to predict water quality using explainable artificial intelligence. *Science of The Total Environment*, 832:155070.

Park, Y., Cho, K. H., Park, J., Cha, S. M., and Kim, J. H. (2015). Development of early-warning protocol for predicting chlorophyll-a concentration using machine learning models in freshwater and estuarine reservoirs, korea. *Science of the Total Environment*, 502:31–41.

Park, Y., Lee, H. K., Shin, J.-K., Chon, K., Kim, S., Cho, K. H., Kim, J. H., and Baek, S.-S. (2021). A machine learning approach for early warning of cyanobacterial bloom outbreaks in a freshwater reservoir. *Journal of Environmental Management*, 288:112415.

Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S., and Mohammadian, A. K. (2020). Toward safer highways, application of xgboost and shap for real-time accident detection and feature analysis. *Accident Analysis & Prevention*, 136:105405.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

Pereira, L. S. (2017). Water, agriculture and food: challenges and issues. *Water Resources Management*, 31(10):2985–2999.

Pham, H.-H., Khoudour, L., Crouzil, A., Zegers, P., and Velastin, S. A. (2018). Exploiting deep residual networks for human action recognition from skeletal data. *Computer Vision and Image Understanding*, 170:51–66.

Poria, S., Cambria, E., and Gelbukh, A. (2015). Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2539–2544.

Pu, F., Ding, C., Chao, Z., Yu, Y., and Xu, X. (2019). Water-quality classification of inland lakes using landsat8 images by convolutional neural networks. *Remote Sensing*, 11(14):1674.

Pyo, J., Park, L. J., Pachepsky, Y., Baek, S.-S., Kim, K., and Cho, K. H. (2020). Using convolutional neural network for predicting cyanobacteria concentrations in river water. *Water Research*, 186:116349.

Qin, M., Li, Z., and Du, Z. (2017). Red tide time series forecasting by combining arima and deep belief network. *Knowledge-Based Systems*, 125:39–52.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rakhlin, A., Davydow, A., and Nikolenko, S. I. (2018). Land cover classification from satellite imagery with u-net and lovasz-softmax loss. In *CVPR Workshops*, pages 262–266.

Ratolojanahary, R., Ngouna, R. H., Medjaher, K., Junca-Bourié, J., Dauriac, F., and Sebilo, M. (2019). Model selection to improve multiple imputation for handling high rate missingness in a water quality dataset. *Expert Systems with Applications*, 131:299–307.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " 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*, pages 1135–1144.

Rigby, G. and Hallegraeff, G. (1996). Ballast water controls to minimize the translocation and establishment of toxic phytoplankton: What progress have we made and where are we going? In *Seventh International Conference on Toxic Phytoplankton*, pages 201–204.

Robinson, C., Hohman, F., and Dilkina, B. (2017). A deep learning approach for population estimation from satellite imagery. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, pages 47–54.

Rodríguez, R., Pastorini, M., Etcheverry, L., Chreties, C., Fossati, M., Castro, A., and Gorgoglione, A. (2021). Water-quality data imputation with a high percentage of missing values: A machine learning approach. *Sustainability*, 13(11):6318.

Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., and Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS journal of photogrammetry and remote sensing*, 67:93–104.

Rubin, D. B. (1978). Multiple imputations in sample surveys-a phenomenological bayesian approach to nonresponse. In *Proceedings of the survey research methods section of the American Statistical Association*, volume 1, pages 20–34. American Statistical Association.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Russell, S. and Norvig, P. (2002). Artificial intelligence: a modern approach.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Santos, M. S., Abreu, P. H., Wilk, S., and Santos, J. (2020). How distance metrics influence missing data imputation with k-nearest neighbours. *Pattern Recognition Letters*, 136:111–119.

Sarkar, A. and Pandey, P. (2015). River water quality modelling using artificial neural network technique. *Aquatic procedia*, 4:1070–1077.

Sarkar, S. K. (2018). *Marine Algal Bloom: Characteristics, Causes and Climate Change Impacts*. Springer.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Sebastiá-Frasquet, M.-T., Aguilar-Maldonado, J.-A., Herrero-Durá, I., Santamaría-del Ángel, E., Morell-Monzó, S., and Estornell, J. (2020). Advances in the monitoring of algal blooms by remote sensing: A bibliometric analysis. *Applied Sciences*, 10(21):7877.

Semeniuta, S., Severyn, A., and Barth, E. (2017). A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*.

Shamshirband, S., Jafari Nodoushan, E., Adolf, J. E., Abdul Manaf, A., Mosavi, A., and Chau, K.-w. (2019). Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters. *Engineering Applications of Computational Fluid Mechanics*, 13(1):91–101.

Shan, K., Ouyang, T., Wang, X., Yang, H., Zhou, B., Wu, Z., and Shang, M. (2022). Temporal prediction of algal parameters in three gorges reservoir based on highly time-resolved monitoring and long short-term memory network. *Journal of Hydrology*, 605:127304.

Shehhi, M. R. A. and Kaya, A. (2020). Time series and machine learning to forecast the water quality from satellite data. *arXiv preprint arXiv:2003.11923*.

Shin, J., Kim, S. M., Son, Y. B., Kim, K., and Ryu, J.-H. (2019). Early prediction of margalefidinium polykrikoides bloom using a lstm neural network model in the south sea of korea. *Journal of Coastal Research*, 90(SI):236–242.

Shin, Y., Kim, T., Hong, S., Lee, S., Lee, E., Hong, S., Lee, C., Kim, T., Park, M. S., Park, J., et al. (2020). Prediction of chlorophyll-a concentrations in the nakdong river using machine learning methods. *Water*, 12(6):1822.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR.

Shu, W., Li, J., Lan, Z., Xu, G., Nie, J., and Liu, S. (2021). Missing value imputation and prediction of river water quality based on gru–autoencoder with input-decay. In *2021 40th Chinese Control Conference (CCC)*, pages 8169–8174. IEEE.

Shu, X., Porikli, F., and Ahuja, N. (2014). Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3874–3881.

Shukla, J., Misra, A., and Chandra, P. (2008). Modeling and analysis of the algal bloom in a lake caused by discharge of nutrients. *Applied Mathematics and Computation*, 196(2):782–790.

Shumway, S. E., Burkholder, J. M., and Morton, S. L. (2018). *Harmful algal blooms: a compendium desk reference.* John Wiley & Sons.

Shutler, J., Grant, M., Miller, P., Rushton, E., and Anderson, K. (2010). Coccolithophore bloom detection in the north east atlantic using seawifs: Algorithm description, application and sensitivity analysis. *Remote Sensing of Environment*, 114(5):1008–1016.

Shutler, J. D., Davidson, K., Miller, P. I., Swan, S. C., Grant, M. G., and Bresnan, E. (2012). An adaptive approach to detect high-biomass algal blooms from eo chlorophyll-a data in support of harmful algal bloom monitoring. *Remote Sensing Letters*, 3(2):101–110.

Shutova, E., Kiela, D., and Maillard, J. (2016). Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170.

Silva, H. N. and Panella, M. (2018). Eutrophication analysis of water reservoirs by remote sensing and neural networks. In *2018 Progress in Electromagnetics Research Symposium (PIERS-Toyama)*, pages 458–463. IEEE.

Simis, S. G., Huot, Y., Babin, M., Seppälä, J., and Metsamaa, L. (2012). Optimization of variable fluorescence measurements of phytoplankton communities with cyanobacteria. *Photosynthesis research*, 112(1):13–30.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, A., Balaji, J. J., Rasheed, M. A., Jayakumar, V., Raman, R., and Lakshminarayanan, V. (2021). Evaluation of explainable deep learning methods for ophthalmic diagnosis. *Clinical Ophthalmology (Auckland, NZ)*, 15:2573.

Singha, S., Pasupuleti, S., Singha, S. S., and Kumar, S. (2020). Effectiveness of groundwater heavy metal pollution indices studies by deep-learning. *Journal of Contaminant Hydrology*, 235:103718.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, 2016.

Smayda, T. J. (2002). Turbulence, watermass stratification and harmful algal blooms: an alternative view and frontal zones as “pelagic seed banks”. *Harmful Algae*, 1(1):95–112.

Smil, V. (2004). *Enriching the earth: Fritz Haber, Carl Bosch, and the transformation of world food production.* MIT press.

Smith, G. J. and Daniels, V. (2018). Algal blooms of the 18th and 19th centuries. *Toxicon*, 142:42–44.

Son, J., Baek, M., Cho, M., and Han, B. (2017). Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5620–5629.

Song, W., Dolan, J. M., Cline, D., and Xiong, G. (2015). Learning-based algal bloom event recognition for oceanographic decision support system using remote sensing data. *Remote Sensing*, 7(10):13564–13585.

Souza, V., Nobre, J., and Becker, K. (2021). A deep learning ensemble to classify anxiety, depression, and their comorbidity from texts of social networks.

Srivastava, N., Salakhutdinov, R., et al. (2012). Multimodal learning with deep boltzmann machines. In *NIPS*, volume 1, page 2. Citeseer.

Steinkraus, D., Buck, I., and Simard, P. (2005). Using gpus for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1115–1120. IEEE.

Stekhoven, D. J. (2015). missforest: Nonparametric missing value imputation using random forest. *Astrophysics Source Code Library.*

Sucholutsky, I., Narayan, A., Schonlau, M., and Fischmeister, S. (2019). Pay attention and you won't lose it: a deep learning approach to sequence imputation. *PeerJ Computer Science*, 5:e210.

Sunda, W. G., Graneli, E., and Gobler, C. J. (2006). Positive feedback and the development and persistence of ecosystem disruptive algal blooms 1. *Journal of Phycology*, 42(5):963–974.

Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. *Advances in NIPS*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Tabari, H. and Hosseinzadeh Talaee, P. (2015). Reconstruction of river water quality missing data using artificial neural networks. *Water Quality Research Journal of Canada*, 50(4):326–335.

Tao, X., Peng, Y., Zhao, F., Zhao, P., and Wang, Y. (2018). A parallel algorithm for network traffic anomaly detection based on isolation forest. *International Journal of Distributed Sensor Networks*, 14(11):1550147718814471.

Taylor, S. L., Mahler, M., Theobald, B.-J., and Matthews, I. (2012). Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pages 275–284.

Ti, C. and Yan, X. (2013). Spatial and temporal variations of river nitrogen exports from major basins in china. *Environmental Science and Pollution Research*, 20(9):6509–6520.

Tian, Y. and Huang, M. (2019). An integrated web-based system for the monitoring and forecasting of coastal harmful algae blooms: Application to shenzhen city, china. *Journal of Marine Science and Engineering*, 7(9):314.

Tong, W., Li, L., Zhou, X., Hamilton, A., and Zhang, K. (2019). Deep learning pm 2.5 concentrations with bidirectional lstm rnn. *Air Quality, Atmosphere & Health*, 12(4):411–423.

Tran, L., Liu, X., Zhou, J., and Jin, R. (2017). Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1405–1414.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525.

Usuga-Cadavid, J. P., Lamouri, S., Grabot, B., and Fortin, A. (2021). Using deep learning to value free-form text data for predictive maintenance. *International Journal of Production Research*, pages 1–28.

Uzkent, B., Barkana, B. D., and Cevikalp, H. (2012). Non-speech environmental sound classification using svms with a new set of features. *International Journal of Innovative Computing, Information and Control*, 8(5):3511–3524.

Van Buuren, S. (2018). *Flexible imputation of missing data*. CRC press.

Vannah, B. and Chang, N.-B. (2013). Fusion of hyperspectral remote sensing data for near real-time monitoring of microcystin distribution in lake erie. In *Satellite Data Compression, Communications, and Processing IX*, volume 8871, page 88710M. International Society for Optics and Photonics.

Vapnik, V. (1963). Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vilas, L. G., Spyrakos, E., Palenzuela, J. M. T., and Pazos, Y. (2014). Support vector machine-based method for predicting pseudo-nitzschia spp. blooms in coastal waters (galician rias, nw spain). *Progress in Oceanography*, 124:66–77.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408.

Vogel, S., Ney, H., and Tillmann, C. (1996). Hmm-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Vörösmarty, C. J., McIntyre, P. B., Gessner, M. O., Dudgeon, D., Prusevich, A., Green, P., Glidden, S., Bunn, S. E., Sullivan, C. A., Liermann, C. R., et al. (2010). Global threats to human water security and river biodiversity. *nature*, 467(7315):555–561.

Wang, J., Zhang, S., Mu, X., Hu, X., and Ma, Y. (2022). Research characteristics on cyanotoxins in inland water: Insights from bibliometrics. *Water*, 14(4):667.

Wang, L., Zhang, T., Jin, X., Xu, J., Wang, X., Zhang, H., Yu, J., Sun, Q., Zhao, Z., and Xie, Y. (2020). An approach of recursive timing deep belief network for algal bloom forecasting. *Neural Computing and Applications*, 32(1):163–171.

Wang, X. and Xu, L. (2020). Unsteady multi-element time series analysis and prediction based on spatial-temporal attention and error forecast fusion. *Future Internet*, 12(2):34.

Wells, M. L., Trainer, V. L., Smayda, T. J., Karlson, B. S., Trick, C. G., Kudela, R. M., Ishikawa, A., Bernard, S., Wulff, A., Anderson, D. M., et al. (2015). Harmful algal blooms and climate change: Learning from the past and present to forecast the future. *Harmful algae*, 49:68–93.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Wu, J., Ma, D., and Wang, W. (2022). Leakage identification in water distribution networks based on xgboost algorithm. *Journal of Water Resources Planning and Management*, 148(3):04021107.

Wu, Z., Cai, L., and Meng, H. (2006). Multi-level fusion of audio and visual features for speaker identification. In *International Conference on Biometrics*, pages 493–499. Springer.

Xiao, X., He, J., Huang, H., Miller, T. R., Christakos, G., Reichwaldt, E. S., Ghadouani, A., Lin, S., Xu, X., and Shi, J. (2017). A novel single-parameter approach for forecasting algal blooms. *Water research*, 108:222–231.

Xu, G., Meng, Y., Qiu, X., Yu, Z., and Wu, X. (2019). Sentiment analysis of comment texts based on bilstm. *Ieee Access*, 7:51522–51532.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

Xu, W., Sun, H., Deng, C., and Tan, Y. (2017). Variational autoencoder for semi-supervised text classification. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Yajima, H. and Derot, J. (2018). Application of the random forest model for chlorophyll-a forecasts in fresh and brackish water bodies in japan, using multivariate long-term databases. *Journal of Hydroinformatics*, 20(1):206–220.

Yang, G. and Moyer, D. L. (2020). Estimation of nonlinear water-quality trends in high-frequency monitoring data. *Science of the Total Environment*, 715:136686.

Yang, Y., Bai, Y., Wang, X., Wang, L., Jin, X., and Sun, Q. (2020). Group decision-making support for sustainable governance of algal bloom in urban lakes. *Sustainability*, 12(4):1494.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (2010). I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508.

Yates, F. (1933). The analysis of replicated experiments when the field results are incomplete. *Empire Journal of Experimental Agriculture*, 1(2):129–142.

Ye, L., Cai, Q., Zhang, M., and Tan, L. (2014). Real-time observation, early warning and forecasting phytoplankton blooms by integrating in situ automated online sondes and hybrid evolutionary algorithms. *Ecological informatics*, 22:44–51.

Yi, H.-S., Park, S., An, K.-G., and Kwak, K.-C. (2018). Algal bloom prediction using extreme learning machine models at artificial weirs in the nakdong river, korea. *International journal of environmental research and public health*, 15(10):2078.

Yi, X., Zheng, Y., Zhang, J., and Li, T. (2016). St-mvl: filling missing values in geo-sensory time series data.

Yilmaz, E. and Aydin, D. (2019). Estimation of right censored nonparametric regression solved by knn imputation: A comparative study. *Turkiye Klinikleri Journal of Biostatistics*, 11(2).

Yim, I., Shin, J., Lee, H., Park, S., Nam, G., Kang, T., Cho, K. H., and Cha, Y. (2020). Deep learning-based retrieval of cyanobacteria pigment in inland water for in-situ and airborne hyperspectral data. *Ecological Indicators*, 110:105879.

Yoon, J., Jordon, J., and Van Der Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*.

Yu, C. and Ballard, D. H. (2004). On the integration of grounding language and learning objects. In *AAAI*, volume 4, page 2.

Yu, Z., Yang, K., Luo, Y., and Shang, C. (2020). Spatial-temporal process simulation and prediction of chlorophyll-a concentration in dianchi lake based on wavelet analysis and long-short term memory network. *Journal of Hydrology*, 582:124488.

Yussof, F. N., Maan, N., and Md Reba, M. N. (2021). Lstm networks to improve the prediction of harmful algal blooms in the west coast of sabah. *International Journal of Environmental Research and Public Health*, 18(14):7650.

Zachary, H. (2017). A harmful algal bloom in lake erie in september 2017. [Online; accessed July 12, 2022].

Zhang, A., Song, S., Sun, Y., and Wang, J. (2019a). Learning individual models for imputation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 160–171. IEEE.

Zhang, F., Wang, Y., Cao, M., Sun, X., Du, Z., Liu, R., and Ye, X. (2016). Deep-learning-based approach for prediction of algal blooms. *Sustainability*, 8(10):1060.

Zhang, Y. and Thorburn, P. J. (2021). A dual-head attention model for time series data imputation. *Computers and Electronics in Agriculture*, 189:106377.

Zhang, Y.-F., Thorburn, P., Xiang, W., and Fitch, P. (2019b). Ssim-a deep learning approach for recovering missing time series sensor data. *IEEE Internet of Things Journal*.

Zheng, L., Wang, H., Liu, C., Zhang, S., Ding, A., Xie, E., Li, J., and Wang, S. (2021). Prediction of harmful algal blooms in large water bodies using the combined efdc and lstm models. *Journal of Environmental Management*, 295:113060.

Zhou, K. and Wei, L. (2022). Grading prediction of kidney renal clear cell carcinoma by deep learning. In *2022 2nd International Conference on Bioinformatics and Intelligent Computing*, pages 37–42.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Zingone, A. and Enevoldsen, H. O. (2000). The diversity of harmful algal blooms: a challenge for science and management. *Ocean & coastal management*, 43(8-9):725–748.

Zinovyeva, E., Härdle, W. K., and Lessmann, S. (2020). Antisocial online behavior detection using deep learning. *Decision Support Systems*, 138:113362.