applied sciences

MDPI

Article

Structure Estimation of Adversarial Distributions for Enhancing Model Robustness: A Clustering-Based Approach

Bader Rasheed [1,*], Adil Khan [1,2] and Asad Masood Khattak [3]

[1] Institute of Data Science and Artificial Intelligence, Innopolis University, Niversitetskaya Str, Innopolis 420500, Russia; a.m.khan@hull.ac.uk
[2] School of Computer Science, University of Hull, Hull HU6 7RX, UK
[3] College of Technological Innovation, Zayed University, Abu Dhabi Campus, Abu Dhabi 144534, United Arab Emirates; asad.khattak@zu.ac.ae
* Correspondence: b.rasheed@innopolis.university

Abstract: In this paper, we propose an advanced method for adversarial training that focuses on leveraging the underlying structure of adversarial perturbation distributions. Unlike conventional adversarial training techniques that consider adversarial examples in isolation, our approach employs clustering algorithms in conjunction with dimensionality reduction techniques to group adversarial perturbations, effectively constructing a more intricate and structured feature space for model training. Our method incorporates density and boundary-aware clustering mechanisms to capture the inherent spatial relationships among adversarial examples. Furthermore, we introduce a strategy for utilizing adversarial perturbations to enhance the delineation between clusters, leading to the formation of more robust and compact clusters. To substantiate the method's efficacy, we performed a comprehensive evaluation using well-established benchmarks, including MNIST and CIFAR-10 datasets. The performance metrics employed for the evaluation encompass the adversarial clean accuracy trade-off, demonstrating a significant improvement in both robust and standard test accuracy over traditional adversarial training methods. Through empirical experiments, we show that the proposed clustering-based adversarial training framework not only enhances the model's robustness against a range of adversarial attacks, such as FGSM and PGD, but also improves generalization in clean data domains.

Keywords: deep neural networks; robustness; adversarial attacks; adversarial training; clustering

Citation: Rasheed, B.; Khan, A.; Masood Khattak, A. Structure Estimation of Adversarial Distributions for Enhancing Model Robustness: A Clustering-Based Approach. *Appl. Sci.* **2023**, *13*, 10972. https://doi.org/10.3390/app131910972

Academic Editor: Mina Sheikhalishahi

Received: 18 August 2023
Revised: 13 September 2023
Accepted: 15 September 2023
Published: 5 October 2023

1. Introduction

With the growing popularity of deep learning (DL), DL-based systems are being applied in a wide variety of areas [1]. Despite their impressive performance, these models remain highly susceptible to adversarial attacks, where perturbed inputs are crafted to deceive the models. Such vulnerability poses significant risks, particularly in security-critical applications.

Adversarial examples, or inputs designed to cause incorrect predictions, not only are intriguing from a scientific standpoint but also have substantial practical implications [2–4]. Numerous techniques have been developed to mitigate the susceptibility of DL models to adversarial attacks [5], with adversarial training emerging as a prominent defensive strategy [2].

Adversarial training, which involves training models on adversarial examples, has emerged as a prominent method to increase resistance to adversarial attacks [2]. Properly trained models not only perform well on clean data but also withstand attacks that these models are trained to resist. Thus, adversarial training holds the key to a new frontier in machine learning where models are both high performing and secure [6].

However, current adversarial training methodologies predominantly suffer from many shortcomings, including model overfitting and a limited capacity, to generalize against

novel types of adversarial attacks [7,8]. Most prior works have either overlooked this shortcoming or treated it as an unavoidable consequence of the exhaustive approach [7,8].

Moreover, existing preprocessing defenses against adversarial attacks, such as input transformations and dimensionality reduction, frequently distort the adversarial examples to make them less effective but do so without gaining any deeper understanding of the adversarial structure [9]. These methods often achieve only partial robustness and are susceptible to adaptive attacks that anticipate such transformations [10].

Contrary to the above approaches, we introduce a novel yet elegantly simple strategy named adversarial structural clustering (ASC) that addresses these issues effectively. Although clustering is a well-understood technique in machine learning, its application in the domain of adversarial training is notably absent. Our approach stands apart in its ability to dissect the complex landscape of adversarial examples into more manageable and interpretable clusters, thereby allowing for a more targeted and effective training regimen.

The simplicity of ASC lies in its utilization of existing clustering algorithms to categorize adversarial examples based on their structural characteristics, yet this simplicity belies its effectiveness [11]. ASC creates a more structured space for training, enabling the model to better capture the commonalities among different types of adversarial input perturbations. To accomplish this, we introduce a density and boundary-aware clustering algorithm. This allows for effectively capturing and exploiting the salient structure of adversarial distributions. This happens by utilizing adversarial perturbations to enhance the boundary delineation between clusters, forming denser, more robust clusters. The formation of these clusters is guided by the geometric proximity of samples to the decision boundary, which typically indicates a higher susceptibility to adversarial perturbations. The ASC algorithm is inspired by the principles of active learning, a paradigm in machine learning where the learning algorithm selectively queries the most informative instances from the data for training while disregarding noisy instances. In the context of ASC, the most informative instances are those that represent different adversarial structures in the input space. We hypothesize that by training on a structured space that captures the diversity of adversarial structures and excludes outliers, we can enhance the robustness of deep learning models against a broad range of adversarial input perturbations. In the course of our research, we expanded the investigation to explore multiple dimensions. In addition to the specific clustering techniques tailored for adversarial example grouping, we studied strategies for selecting samples from these clusters for adversarial training, where each strategy emphasizes different aspects of the adversarial example's importance. Furthermore, we delved into the role of dimensionality reduction in enhancing the efficiency of our proposed approach. Understanding the interplay between clustering and information content is essential in our method. For this purpose, we studied a central aspect of this relationship, which is how the number of clusters affects the entropy of the clustered adversarial examples. This research contributes to the broader endeavor of creating more reliable and secure machine learning models, enabling their deployment in critical real-world applications.

In summary, this paper aims to answer a pivotal research question: "Can clustering techniques be effectively adapted to uncover the structure of adversarial example space and thereby substantially improve the robustness of deep learning models?" In exploring this question, we delineate novel methodologies, scrutinize their impact through rigorous evaluations, and lay the groundwork for a more robust generation of deep learning algorithms.

The paper is organized as follows: In Section 2, we give the necessary background on potential clustering algorithms and adversarial attacks to be used. Section 3 provides a detailed description of our methodology and presents the specific details of the implementation of the method. Section 4 studies how the number of clusters affects the entropy of the clustered adversarial example. Section 5 shows different experiments and analysis for the clustering-based adversarial training. Finally, Section 6 concludes this paper.

2. Background

In this section, we first delve into the details of generating adversarial examples and existing defense mechanisms, particularly focusing on adversarial training, discussing its limitations and the need for our research. Subsequently, we review clustering methods, emphasizing K-means, agglomerative clustering, and DBSCAN.

2.1. Adversarial Example Generation

Adversarial examples are maliciously crafted input samples designed to confuse machine learning models. These examples look virtually indistinguishable from normal samples to the human eye but can lead to erroneous outputs by the machine learning model. Several adversarial attacks can create such examples; in our study, we will use the fast gradient sign method (FGSM) [2], and projected gradient descent (PGD) [12]. These attacks perturb the input sample in a way that maximizes the model's loss function.

2.1.1. Fast Gradient Sign Method (FGSM)

The fast gradient sign method (FGSM) is an elementary yet effective adversarial attack introduced by Goodfellow et al. [2]. Its computational efficiency stems from its reliance on a first-order approximation of the loss function. For a neural network model characterized by its parameters θ, let $L(\theta, x, y)$ denote the loss function, where x is the input data and y is the true label. The adversarial example x' can be generated as follows:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)) \tag{1}$$

The term $\nabla_x L(\theta, x, y)$ is the gradient of the loss function L with respect to the input x. It is a vector in $\mathbb{R}^d$, where d is the dimensionality of the input. Mathematically, this gradient can be expressed as

$$\nabla_x L(\theta, x, y) = \left[\frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \ldots, \frac{\partial L}{\partial x_d} \right] \tag{2}$$

Here, ϵ is a hyperparameter that controls the magnitude of the perturbation introduced. In essence, FGSM perturbs each element x_i of the input x in the direction that maximizes the increase in loss L.

2.1.2. Projected Gradient Descent (PGD)

The projected gradient descent (PGD) is an extension of FGSM that employs multiple iterations to generate more effective adversarial examples. It is considered a robust first-order adversary due to its efficacy in defeating defensive techniques. An adversarial example x'_i at the ith iteration is computed as

$$x'_i = \text{Clip}_{x,\epsilon}(x'_{i-1} + \alpha \cdot \text{sign}(\nabla_x L(\theta, x'_{i-1}, y))) \tag{3}$$

The gradient term $\nabla_x L(\theta, x'_{i-1}, y)$ is computed similarly to Equation (1), but at each iteration i, the gradient is computed at the perturbed point x'_{i-1} instead of the original point x.

The term $\text{Clip}_{x,\epsilon}$ is an operation that projects x'_i into an ϵ-ball around x. Formally, $\text{Clip}_{x,\epsilon}(z)$ can be defined as

$$\text{Clip}_{x,\epsilon}(z) = x + \text{clamp}(z - x, -\epsilon, \epsilon) \tag{4}$$

Here, α is the step size controlling how much the adversarial example changes in each iteration. Note that α may differ from ϵ used in Equation (1), and its optimal value can be data and model dependent.

2.2. Defenses against Adversarial Attacks

There exist other defense strategies in the literature, which range from adversarial training and model architecture modifications to data preprocessing defenses.

2.2.1. Model Architecture Modifications

These defenses alter the model architecture to introduce intrinsic robustness against adversarial perturbations. Methods such as defensive distillation [13] and feature squeezing [14] fall into this category. However, these approaches often do not generalize well against diverse attack types and can introduce additional computational overhead [10].

2.2.2. Data Preprocessing Defenses

Data preprocessing defenses, such as input transformations [15] and randomization [16], work by transforming the adversarial examples before feeding them into the model. They aim to either remove or distort the adversarial perturbations so that the model can classify the altered input correctly. These methods are partially effective but are often circumvented by adaptive attacks that anticipate such transformations [10].

While data preprocessing defenses have previously been employed, none have leveraged clustering to understand the inherent structure of adversarial examples, thus missing out on a more strategic, targeted, and effective training regimen.

2.2.3. Adversarial Training Defense

Adversarial training (AT) is a defense mechanism designed to improve the model's resilience to these adversarial attacks [2]. It incorporates adversarial examples (generated from any of the aforementioned attacks) into the training process. The intention is to enable the model to learn from these malicious examples and improve its robustness, equipping it to better classify similar adversarial examples in the future.

However, despite its effectiveness, AT has limitations. One primary challenge is that the process is computationally expensive. Moreover, it has been observed that using all adversarial examples for training may not necessarily improve model robustness, as some adversarial examples may only introduce noise into the learning process [17]. This is where our research comes in: we propose a method that judiciously selects adversarial examples for training, with the aim of improving model robustness.

2.3. Clustering Methods

Clustering is the task of dividing data into groups (clusters) so that items within a group are more similar to each other than to items in other groups [18]. The simplicity of our approach lies in its ability to extend conventional clustering algorithms by modifying the distance metric to incorporate the adversarial degradation potentials. This shift in distance metric captures the susceptibility of data instances to adversarial perturbations, allowing for a more robust partitioning in the presence of adversarial noise. In this research, we focus on K-means, agglomerative clustering, and density-based spatial clustering of applications with noise (DBSCAN).

K-means is a centroid-based method that partitions data into K nonoverlapping subsets (clusters) such that each data point belongs to the cluster with the nearest mean. The algorithm iteratively assigns points to the closest centroid and recalculates the centroid until a stopping criterion is met.

Agglomerative clustering is a type of hierarchical clustering method that builds a cluster hierarchy. This method starts with each data point as an individual cluster and merges the closest pair of clusters until only one cluster (or a specified number of clusters) remains.

DBSCAN is a density-based clustering method that is particularly effective for discovering clusters of arbitrary shape and removing noise. It operates by defining a neighborhood around a data point, and if the number of points within this neighborhood exceeds a certain threshold, a new cluster is formed. If the neighborhood contains fewer points than the

threshold but lies within the neighborhood of another cluster, it is added to that cluster. If it does not meet either criterion, it is classified as noise.

Unlike K-means and agglomerative clustering, which require the number of clusters as input, DBSCAN infers the number of clusters based on the data, making it more versatile in handling complex datasets. Moreover, DBSCAN's ability to detect noise can be beneficial in the context of adversarial examples, as it allows for the separation of potent adversarial examples from less impactful ones.

While traditional clustering algorithms provide a strong foundational base, other more advanced clustering methods, such as biclustering and triclustering, could also be adapted in a similar fashion.

Biclustering involves clustering both the rows and columns of a data matrix simultaneously. This allows for the simultaneous clustering of instances and features, which can lead to more precise and potentially more robust clustering results.

Triclustering further extends this by clustering in three dimensions, adding another layer of complexity but also more opportunities for capturing intricate relationships within the data.

These advanced clustering methods could be incorporated into our model in the same manner as the traditional clustering methods, by extending their respective distance metrics with the adversarial degradation potentials. The advantage would be more nuanced cluster formations, which could potentially be more robust to adversarial perturbations. However, it is worth noting that these advanced methods often entail higher computational costs, particularly when dealing with large datasets. The complexity can scale rapidly with the increase in dimensions and the number of data points. Therefore, while biclustering and triclustering methods may yield more accurate and robust clusters, they may not be feasible for larger-scale applications. Our current model aims to strike a balance between accuracy and computational efficiency, providing a robust yet scalable solution.

In the following sections, we will elaborate on the integration of these clustering methods into our adversarial example generation and selection process, with a focus on addressing the limitations of traditional adversarial training.

3. Methodology

This section provides a detailed description of our methodology, which involves generating adversarial examples, clustering these examples, and selecting specific samples from the clusters for adversarial training (AT). We also present the specific details of the implementation of our idea, including the clustering algorithms used, the criteria for sample selection, and the process of dimensionality reduction. Figure 1 provides an overview of our proposed methodology.

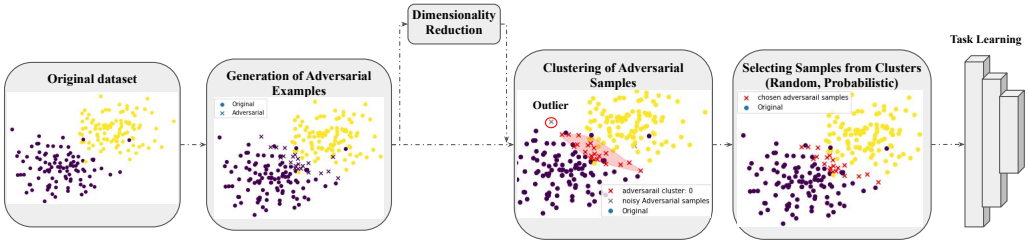

Figure 1. Overview of the proposed methodology. The blue and purple dots represent the original samples from the two classes. The (×) samples represent adversarial samples generated from both classes. Further, the adversarial (×) samples are clustered, and the red samples represent samples chosen by the cluster, while the other off black (×) samples are not inside the cluster.

3.1. Adversarial Example Generation

Our methodology commences with the generation of adversarial examples. Employing the fast gradient sign method (FGSM) and projected gradient descent (PGD), we produce

adversarial samples that span the input space. This facilitates the creation of a diverse set of samples for subsequent operations.

3.2. Dimensionality Reduction

Given the high dimensionality of image data, we also employ dimensionality reduction techniques as an optional step before clustering to speed up the clustering process and potentially improve its effectiveness. In this study, we consider principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) due to their prevalence and performance in reducing the dimensionality of data [19].

PCA is a linear method that identifies the directions (principal components) in which the data vary the most and projects the data onto these directions, discarding less informative ones. On the other hand, t-SNE is a nonlinear method that constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects are assigned a higher probability, and dissimilar ones a lower probability. The same is done for the low-dimensional counterparts. t-SNE then minimizes the divergence between the two distributions, ensuring that similar objects in high-dimensional space remain close in the reduced space.

3.3. Clustering of Adversarial Examples

Following the generation of adversarial examples, the next step involves clustering these examples. The rationale behind this is that similar adversarial examples, which lead to similar misclassifications, are likely to be grouped together. By clustering adversarial examples, we can identify these groups and tailor our adversarial training process accordingly.

Our approach introduces an extension for the previously described clustering algorithms to effectively capture the structure of adversarial distributions. This makes the clustering a boundary-aware clustering method that makes use of adversarial perturbations to enhance the boundary delineation between clusters.

The core idea is to use adversarial perturbations as a guide to shape the formation of clusters, creating denser and more robust clusters that capture the salient structure of adversarial distributions. The algorithm works by iteratively adjusting cluster boundaries based on adversarial feedback, creating a flexible, dynamic clustering algorithm that adapts to the complex structure of adversarial data.

Our proposed adversarial clustering algorithm can be viewed as a specific case of clustering algorithms, where the neighborhood definition is based on a novel proposed distance metric, which incorporates the degradation potential of adversarial examples, thereby enabling our clustering algorithm to group adversarial examples based on their potential threat to the model.

3.4. Clustering of Adversarial Samples: Distance Metric and Degradation Potential

We introduce a novel distance metric sensitive to the degradation potential of adversarial examples. This allows our clustering algorithm to group adversarial examples based on the degree of threat they present to the model, rather than solely on their proximity in the input space.

First, we define the degradation potential of an adversarial example. Given a neural network model f with parameters θ, an input x, and its corresponding adversarial input x_{adv}, we define the degradation potential of x_{adv} on f as follows:

$$\Delta_{adv}(x_{adv}, f) = \nabla L(f(x_{adv}; \theta), y) \tag{5}$$

where L denotes the loss function between the model's prediction $f(x_{adv})$ and the true label y. ∇L is the gradient of the loss function L with respect to the input. The term $\nabla L(f(x_{adv}; \theta), y)$ signifies the change in the model's loss caused by the adversarial example. Calculating this degradation potential is already performed when generating the adversarial samples, and all that should be done is to save it.

Next, we define the distance metric D_{adv} between two adversarial examples x_{adv1} and x_{adv2} in a space governed by the l_p norm and the degradation potential Δ_{adv}:

$$D_{adv}(x_{adv1}, x_{adv2}) = ||x_{adv1} - x_{adv2}||_p + \lambda|\Delta_{adv}(x_{adv1}, f) - \Delta_{adv}(x_{adv2}, f)| \quad (6)$$

The $||||_p$ captures the geometric distance between the two adversarial samples. λ is a trade-off parameter that controls the relative importance of the input space distance and the degradation potential difference.

This metric ensures that adversarial examples causing a similar degree of degradation to the model performance are grouped together. Conversely, adversarial examples causing distinct levels of performance degradation might be separated, even if they are close in the input space.

Thus, our distance metric allows the clustering algorithm to capture the nuanced characteristics of adversarial examples, differentiating clusters according to the nature and threat level of the adversarial samples. This, in turn, provides a more informed basis for adversarial training, resulting in more robust models.

We are interested in a more nuanced way of characterizing the "difference" between degradation potentials of two adversarial samples. Rather than simply taking the absolute difference of degradation potentials, we could quantify the relationship between the degradation potentials of two adversarial samples. There are various methods to measure the correlation or similarity between two sets of data, such as Pearson correlation coefficient (PCC) [20], Spearman's rank correlation coefficient [21], Kullback–Leibler (KL) divergence [22], and mutual information (MI) [23].

Among these, KL divergence and mutual information might be the most appropriate for our case since they are capable of capturing complex relationships between probability distributions, which could be useful given the highly nonlinear nature of deep neural networks.

We could incorporate KL divergence or mutual information into the distance metric D_{adv} as follows:

$$D_{adv}(x_{adv1}, x_{adv2}) = ||x_{adv1} - x_{adv2}||_p + \lambda D_{KL}(\Delta_{adv}(x_{adv1}, M)||\Delta_{adv}(x_{adv2}, M)) \quad (7)$$

where λ is trade-off parameters controlling the importance of the input space distance, the KL divergence, and the mutual information.

$D_{KL}(P||Q)$ is the KL divergence between the degradation potentials of x_{adv1} and x_{adv2}. Let us say that P and Q are two probability distributions. The KL divergence of Q from P is defined as follows:

$$D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \quad (8)$$

where i ranges over all possible events.

In our case, P and Q are the probability distributions representing the degradation potentials of x_{adv1} and x_{adv2}, respectively. The degradation potential of an adversarial example could be represented as a vector of probabilities over the neural network's outputs, possibly obtained by applying a softmax function to the output logits of the network when fed with the adversarial example.

A crucial thing to note is that KL divergence is not symmetric, i.e., $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. This means that the order in which the adversarial examples are compared will affect the result. This could be leveraged to encode some form of directionality into the clustering process, for example, by always placing the adversarial example that is deemed more harmful (causes a larger increase in the loss) as P and the less harmful one as Q. This way, clusters will be more influenced by the more harmful examples, which could potentially enhance the robustness of the resulting model.

One practical consideration when computing KL divergence is that it is not defined when $P(i) > 0$ and $Q(i) = 0$. A common solution is to add a small constant to the distributions, known as smoothing, to ensure that all probabilities are nonzero. This needs to be done carefully to avoid significantly altering the original distributions.

This version of D_{adv} allows the clustering algorithm to capture the divergence between the ways different adversarial samples affect the model's behavior, thereby providing a richer characterization of the adversarial sample space.

The overall ASC clustering algorithm is given in Algorithm 1.

Algorithm 1 Density sensitive adversarial clustering

Require: Set of adversarial examples $\{x_{adv}\}$, neural network model f with parameters θ , l_p norm p , trade-off parameter λ

Ensure: Clustering of adversarial examples $\{C\}$

Compute the degradation potential for all adversarial examples:
for x_{adv} in $\{x_{adv}\}$ **do**
    $\Delta_{adv}(x_{adv}, f) = \nabla L(f(x_{adv}; \theta), y)$
end for
Compute the distance matrix using the novel distance metric:
for pair of adversarial examples (x_{adv1}, x_{adv2}) in $\{x_{adv}\}$ **do**
    $D_{adv}(x_{adv1}, x_{adv2}) = ||x_{adv1} - x_{adv2}||_p + \lambda \cdot D_{KL}(\Delta_{adv}(x_{adv1}, M)||\Delta_{adv}(x_{adv2}, M))$
end for
Apply any clustering algorithm to $\{x_{adv}\}$ using the distance matrix D_{adv}:
$C = \text{clustering}(\{x_{adv}\}, D_{adv})$
return the clusters $\{C\}$

It is important to note that the methodology is not limited to a specific model or dataset and is generalizable across different machine learning tasks.

3.5. Selection of Samples from Clusters

Once the adversarial examples have been generated and subsequently clustered, an essential task is the selection of representative samples from these clusters for adversarial training. This step is crucial because not all adversarial examples hold equal significance for bolstering the robustness of the model. Certain examples can reveal more about the adversarial subspaces that the model has difficulty with, thereby providing more valuable insights for model improvement.

The underlying premise of our approach is that incorporating a subset of carefully chosen adversarial examples into the training process can offer equivalent, if not superior, improvements to model robustness than using the entire adversarial example set, while also reducing the computational load.

To facilitate this, we adopt two strategies for the selection of adversarial examples from the clusters.

3.5.1. Random Cluster Selection

The first strategy involves selecting one cluster randomly in each training iteration. This method ensures that the model is exposed to a wide variety of adversarial perturbations throughout the training process. By rotating through different clusters, we avoid the model overfitting to the adversarial examples from a single or a few clusters, thus ensuring a more comprehensive improvement in robustness.

In mathematical terms, if we denote the set of all clusters as $C = C_1, C_2, \ldots, C_n$, in each iteration i, we select a random cluster C_k, where $k \in 1, 2, \ldots, n$. All adversarial examples from the selected cluster C_k are then used for adversarial training in that iteration.

3.5.2. Probabilistic Sample Selection within Clusters

The second strategy incorporates all clusters but selects samples from each cluster based on a sampling probability that depends on their distance from the cluster center. The rationale behind this approach is to prioritize examples that are more representative of the adversarial behavior captured by each cluster.

This strategy operates under the assumption that samples closer to the cluster center are more "typical" adversarial examples for that cluster, while outliers might introduce noise or be less informative. Hence, samples closer to the center are selected with a higher probability, effectively serving as a strategy to remove unimportant or noisy samples.

Mathematically, if $d(x, C_k)$ denotes the distance of a sample x from the center of cluster C_k, the probability of selecting a sample x from cluster C_k for adversarial training, denoted as $p(x, C_k)$, can be defined as

$$p(x, C_k) = \frac{1}{d(x, C_k) + \epsilon} \tag{9}$$

Here, ϵ is a small constant to prevent division by zero. This inverse relationship ensures that samples closer to the cluster center (smaller $d(x, C_k)$) have a higher selection probability.

Both strategies allow us to capitalize on the information provided by adversarial examples in a more targeted and efficient manner, leading to enhanced model robustness with reduced computational overhead.

4. Clustering and Information Content

Understanding the interplay between clustering and information content is important in our method. A central aspect of this relationship is how the number of clusters affects the entropy of the clustered adversarial examples. This section illuminates the relation between information richness and clustering granularity.

4.1. Defining Cluster Variation and Entropy

Cluster variation: The within-cluster variation, denoted as W_y, for a specific cluster y is the sum of squared distances of the data points within that cluster to its center. Mathematically, for a cluster y with centroid c_y, this is given by

$$W_y = \sum_{x \in y} ||x - c_y||^2.$$

The total cluster variation W is the summation of the within-cluster variations for all clusters.

Entropy: Entropy, denoted by $H(Y)$, is a measure of the uncertainty or randomness of a random variable. In the context of clustering, Y is the random variable representing cluster assignment. The Shannon entropy for a discrete random variable with probability distribution $p(y)$ is defined as

$$H(Y) = -\sum_{y=1}^{K} p(y) \log(p(y))$$

where K is the number of clusters.

4.2. Connection between Cluster Variation and Probability Distribution

To build the connection, we need to make the following assumptions:

1. The data are drawn independently from a mixture of multivariate Gaussian distributions corresponding to the clusters. Each Gaussian distribution is characterized by its mean μ_k and a covariance matrix scaled by W_y.

2. Clusters with larger variation occupy a more significant volume in the data space, implying that they are more likely to encompass a randomly drawn data point. The volume occupied by a cluster in data space can be loosely tied to its spread or variation. For a multivariate Gaussian distribution, the volume it occupies is directly related to the determinant of its covariance matrix. In simple terms, if a cluster's data points are widely dispersed, its "spread" or "variation" is large, and it occupies a more significant volume in the space. As we have tied the cluster variation W_y to the variance of our Gaussian distribution, clusters with larger W_y will have a larger determinant of their covariance matrix, thereby occupying a larger volume. Thus, the probability of a randomly drawn point falling within such a cluster is higher, validating our assumption.

Given that we are considering our data to be drawn from a mixture of multivariate Gaussian distributions (each corresponding to a cluster), the volume occupied by each cluster becomes pivotal in understanding the likelihood of a random sample falling within a particular cluster.

The volume V_y of the region that is captured by cluster y is influenced by the square root of its variance (or in multidimensional space, the determinant of its covariance matrix). Given that W_y is essentially a measure of the squared distances (variances) from the centroid, it stands to reason that the volume V_y is proportional to $W_y^{d/2}$, where d is the dimensionality of the data.

Assuming that data points are uniformly and randomly sampled, the probability $p(y)$ that a point lies within cluster y is proportional to the volume V_y of cluster y.

Given the volumes of all clusters, the probability that a randomly drawn data point lies within cluster y can be determined by the ratio of the volume of cluster y to the sum of the volumes of all clusters:

$$p(y) = \frac{V_y}{\sum_{k=1}^{K} V_k}$$

Substituting our earlier derived relationship for volume in terms of W_y:

$$p(y) = \frac{W_y^{d/2}}{\sum_{k=1}^{K} W_k^{d/2}}$$

4.3. Connecting Cluster Variation to Entropy

Entropy, in the context of information theory, quantifies the amount of uncertainty involved in predicting the value of a random variable. For our clusters, this random variable Y is the event that a randomly drawn data point belongs to cluster y, and the uncertainty in this event is determined by the probability distribution $p(y)$ of the clusters. The entropy $H(Y)$ of this distribution is given by

$$H(Y) = -\sum_{y=1}^{K} p(y) \log p(y)$$

where K is the number of clusters. From our earlier subsection, we related cluster variation W_y to the probability $p(y)$:

$$p(y) = \frac{W_y^{d/2}}{\sum_{k=1}^{K} W_k^{d/2}}$$

Inserting this expression into our entropy formula, we obtain

$$H(Y) = -\sum_{y=1}^{K} \frac{W_y^{d/2}}{\sum_{k=1}^{K} W_k^{d/2}} \log\left(\frac{W_y^{d/2}}{\sum_{k=1}^{K} W_k^{d/2}}\right)$$

Given this expression, a few key insights emerge:

1. As variation increases: When a cluster has a high W_y, meaning its data points are widely dispersed, its contribution to the entropy becomes more significant. This is because the corresponding $p(y)$ value increases, and the logarithmic term in the entropy formula amplifies this effect.

2. Influence of dimensionality d: The dimensionality d of the data can also impact the entropy. As the dimensionality increases, the proportional contribution of each cluster to the entropy can change dramatically, leading to higher uncertainties and thus higher entropy values.

3. Effect of number of clusters K: The total number of clusters also influences the entropy. With fewer clusters, individual $p(y)$ values might be larger, leading to potentially larger entropy values. On the contrary, with more clusters, each cluster's probability could be smaller, potentially reducing the overall entropy.

In essence, the cluster variation provides a proxy measure for the entropy of the clustering. Clusters with high variation (indicative of capturing a wider range of data points with varying characteristics) tend to contribute more to the entropy, symbolizing higher unpredictability or uncertainty in data point assignment to these clusters.

It is also worth briefly noting the theoretical congruence of our approach with frameworks such as information bottleneck (IB) theory [24] and maximum coding rate reduction [25]. In the context of the IB theory, our clustering strategy can be considered as an information compression step, encapsulating the vast space of adversarial examples into more informationally dense cluster representatives. This aids the model in navigating a potentially less complex and more informative adversarial landscape. From the perspective of maximum coding rate reduction, our method could encourage the model to preserve meaningful mutual information between challenging adversarial examples and the output labels, thereby refining the model's ability to focus on relevant 'signals' over 'noise'. However, constructing a rigorous theoretical framework around these ideas would require extensive analysis and is considered as a direction for future research.

5. Experiments

The primary objective of our experimental design is to rigorously validate the efficacy of our cluster-based adversarial training approach. We benchmark our methodology against standard adversarial training (AT). The straightforwardness of AT makes it an ideal baseline, as it allows for isolating the impacts of our novel contributions in clustering over its foundational principles.

In each training epoch, the model undergoes two separate training phases: an initial training phase using clean data, followed by a subsequent training phase that utilizes adversarial examples. The model is dynamically trained on both clean and adversarial instances within the same epoch to improve its generalization capabilities. We employ the L_∞ norm as the metric to quantify the magnitude of perturbations induced by the adversarial attacks. The experimental evaluations were conducted on a single NVIDIA (Santa Clara, CA, USA) GeForce GTX 1080 Ti graphics processing unit.

To precisely address the issue of adversarial data in training, we employ various clustering algorithms to augment the training data with clustered adversarial samples. The generation and clustering of adversarial samples are achieved through a hierarchical process: Initially, adversarial samples are generated, followed by clustering these samples. Finally, they are integrated into the training set as per the selected sampling strategy. We empirically set the number of clusters for K-means and agglomerative clustering within the range [1, 10]. For DBSCAN, which does not require a predetermined number of clusters, only the best results are reported for comparative evaluation with other clustering algorithms.

5.1. Results on MNIST and CIFAR-10

The allowed adversarial perturbation ϵ, in this case, is 0.5 for MNIST [26] and 0.05 for CIFAR-10 [27], and the maximum number of iterations for PGD is 20.

In this study, the choice of hyperparameters was a crucial factor that influenced the performance of the adversarial clustering algorithm. Below, we discuss our choice and tuning of key hyperparameters.

- The p in the l_p norm determines the distance metric in the input space. We experimented with different p-values, such as 1 (Manhattan distance), 2 (Euclidean distance), and ∞ (Chebyshev distance). Through cross-validation on a separate validation set, we found that the l_2 norm (Euclidean distance) provided the best balance between performance and computational efficiency for our experiments.
- λ is the trade-off parameters in our distance metric that control the relative importance of the input space distance and the degradation potential difference. We performed a grid search over a predefined range for λ values, which varied between 0.1 and 1.0. $\lambda = 0.5$ resulted in the best clustering performance on a validation set.
- The hyperparameters of DBSCAN, i.e., MinPts (the minimum number of points required to form a dense region) and ϵ_{DBSCAN} (the maximum distance between two samples for them to be considered as in the same neighborhood), also play a role in our method. MinPts was chosen as 4 following the general recommendation for different datasets. For ϵ_{DBSCAN}, we conducted a series of experiments with values ranging from 0.1 to 1.0. The optimal performance was observed at $\epsilon_{DBSCAN} = 0.5$, so this value was chosen.
- Learning rate and batch size: We use the Adam optimizer with an initial learning rate of 0.001 and a learning rate schedule that reduces the learning rate by a factor of 0.1 every 20 epochs. Momentum is set to 0.9. These choices are made after a comprehensive grid search in the hyperparameter space. The incorporation of learning rate schedules and momentum ensures quicker convergence and adaptability to the local optima of the loss landscape. We note that this strategy obviates the need for a rigorous convergence proof, as the empirical results demonstrate stable and repeatable convergence for our model architecture and dataset.
  Three batch sizes were considered: 32, 64, and 128. A batch size of 64 was selected based on the trade-off between computational efficiency and the stability of the gradient updates.

We conducted an extensive hyperparameter tuning process involving grid search and cross-validation to ensure that our chosen hyperparameters generalized well and led to robust and consistent results across various adversarial settings.

5.2. Overall Comparison

In order to ascertain the comparative effectiveness of the various clustering algorithms, we collate the maximum adversarial and clean accuracies attained under FGSM and PGD attack scenarios. These empirical outcomes are systematically tabulated in Tables 1 and 2, and are juxtaposed with the results obtained through traditional adversarial training techniques.

Table 1. Comparison of model robustness under FGSM and PGD attacks on MNIST.

Model	Clean Accuracy %	FGSM Accuracy %	PGD Accuracy %
Original model (clean data only)	98.5	17.2	13.9
AT baseline	89.1	81.8	68.7
K-means + AT	93.7	88.3	83.1
K-means_ASC + AT	94.1	88.9	83.8
Agglomerative clustering + AT	92.5	87.9	82.2
Agglomerative clustering_ASC + AT	92.7	88.2	82.7
DBSCAN + AT	95.4	91.7	86.6
DBSCAN_ASC + AT	96.3	92.1	87.7

Table 2. Comparison of model robustness under FGSM and PGD attacks on CIFAR-10.

Model	Clean Accuracy %	FGSM Accuracy %	PGD Accuracy %
Original model (clean data only)	83.53	10.5	1.1
AT baseline	73.3	46.0	41.9
K-means + AT	75.6	49.5	44.4
K-means_ASC + AT	75.5	49.9	44.5
Agglomerative clustering + AT	73.5	46.7	42.1
Agglomerative clustering_ASC + AT	73.7	46.9	42.4
DBSCAN + AT	75.1	48.5	42.9
DBSCAN_ASC + AT	75.3	48.9	43.5

From Tables 1 and 2, we see that all models that were preprocessed using clustering algorithms outperformed the baseline adversarial training model. This suggests that the process of clustering, by naturally grouping similar adversarial examples together, has a positive effect on the model's ability to generalize and defend against adversarial attacks.

Interestingly, the model that used our proposed adversarial structural clustering (ASC) algorithm for preprocessing exhibited the highest robustness across both FGSM and PGD attacks. This indicates that ASC, which takes into account the unique properties of adversarial examples, is better equipped to enhance the model's resistance to these specific types of attacks. This substantiates our initial claim that considering the adversarial structure during clustering would yield models with improved robustness.

In conclusion, our experiments validate that integrating clustering algorithms in the preprocessing stage of adversarial training can enhance the model's robustness against adversarial attacks. More importantly, the results show that our proposed ASC algorithm outperforms traditional clustering algorithms, demonstrating its effectiveness in this context.

5.3. The Effect of Dimensionality Reduction

To study the effect of dimensionality reduction, we show the clean and adversarial accuracy on MNIST of three configurations: (1) None: no reduction applied where we apply clustering on the initial dimensions of the data. (2) T-SNE: we apply t-SNE reduction to two dimensions. (3) PCA: we apply PCA reduction to two dimensions.

From Table 3, it is evident that dimensionality reduction techniques, PCA and t-SNE, have substantially enhanced both the clean and adversarial accuracy across all models compared with when no dimensionality reduction was employed. This suggests that the process of dimensionality reduction can help in condensing the information contained in adversarial examples in a manner that facilitates better model performance.

Table 3. Comparison of model performance after dimensionality reduction.

Model	Clean Accuracy			Adversarial Accuracy		
	None	PCA	t-SNE	None	PCA	t-SNE
Original model (clean data only)	98.5	98.2	98.3	17.2	18.4	20.9
AT baseline	89.1	89.3	89.0	81.8	81.9	81.7
K-means + AT	93.7	93.6	93.8	88.3	88.9	89.0
K-means_ASC + AT	94.1	94.2	94.1	89.0	88.9	89.3
Agglomerative clustering + AT	92.5	92.54	92.6	87.9	88.0	88.1
Agglomerative clustering_ASC + AT	92.7	92.8	92.8	88.2	88.6	88.7
DBSCAN + AT	95.4	95.6	95.5	91.7	91.9	92.0
DBSCAN_ASC + AT	96.3	96.5	96.5	92.1	92.8	92.9

Moreover, it is interesting to observe that models preprocessed with t-SNE consistently outperformed those preprocessed with PCA across all clustering methods. This may suggest that t-SNE, with its nonlinear dimensionality reduction properties, is able to capture the complex structure of adversarial examples more effectively than PCA, which is a linear dimensionality reduction technique.

Of all the models, the one using our proposed ASC algorithm with t-SNE preprocessing exhibited the highest clean and adversarial accuracy. This reconfirms the superiority of ASC in handling adversarial examples and highlights the synergistic effect of combining it with appropriate dimensionality reduction techniques.

In conclusion, our results highlight the importance of considering the dimensionality of data in adversarial training. They also underscore the effectiveness of combining our proposed ASC algorithm with t-SNE dimensionality reduction for achieving high robustness against adversarial attacks. The utilization of dimensionality reduction techniques serves a dual purpose in our context. Specifically, it expedites the computational efficiency of clustering algorithms while also improving both the clean and adversarial accuracies of the model.

6. Conclusions

In this research, we conducted an exhaustive investigation into bolstering the robustness of neural network models by leveraging a cluster-based training paradigm in an adversarial context. The distinctiveness of our methodology stems from its deviation from conventional adversarial training schemes; we assert that adversarial instances can be cohesively understood as structured distributions in the input domain. In pursuit of capturing this inherent structure, we employed density-sensitive and boundary-aware clustering techniques, a choice corroborated by rigorous empirical analyses.

Our empirical evaluations provide robust evidence for the efficacy of the proposed framework. Notably, we observed marked improvements in both robust and standard test accuracies, thereby effectively mitigating the adversarial clean accuracy trade-off—a significant accomplishment that substantiates the relevance and validity of our approach. This was further cemented through a comparative performance evaluation against existing clustering-based and traditional adversarial training methods.

The architecture we propose is inherently flexible, permitting the incorporation of a diverse array of clustering algorithms. Such flexibility allows the framework to serve as a fertile ground for future research endeavors, particularly in scrutinizing the trade-offs between computational efficiency and clustering robustness in adversarial settings. Future avenues of exploration could extend beyond the present scope to include applications in semisupervised learning regimes or synergistic integrations with other robustness-enhancing paradigms. Furthermore, the favorable outcomes gleaned from this research endeavor substantively advocate for a continued scholarly investigation into refining and extending the applicability of our proposed methodology, thereby contributing meaningfully to the ongoing endeavors to fortify machine learning models against adversarial perturbations.

Author Contributions: Conceptualization, B.R. and A.K.; methodology, B.R.; software, B.R.; validation, B.R. and A.K.; formal analysis, B.R.; investigation, B.R.; writing—original draft preparation, B.R.; writing—review and editing, B.R., A.K. and A.M.K.; visualization, B.R.; supervision, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Previously reported datasets were used to support this study and are available at https://doi.org/10.1109/MSP.2012.2211477 (accessed on 1 June 2023), and https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 1 June 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AT	adversarial training
DL	deep learning
FGSM	fast gradient sign method
PGD	projected gradient descent
ASC	adversarial structural clustering
DBSCAN	density-based spatial clustering of applications with noise

References

1. Neu, D.A.; Lahann, J.; Fettke, P. A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artif. Intell. Rev.* **2022**, *55*, 801–827.
2. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
3. Rasheed, B.; Khan, A.; Kazmi, S.A.; Hussain, R.; Piran, M.J.; Suh, D.Y. Adversarial attacks on featureless deep learning malicious URLs detection. *Comput. Mater. Contin.* **2021**, *68*, 921–939. [CrossRef]
4. Rasheed, B.; Khan, A.; Ahmad, M.; Mazzara, M.; Kazmi, S. Multiple Adversarial Domains Adaptation Approach for Mitigating Adversarial Attacks Effects. *Int. Trans. Electr. Energy Syst.* **2022**, *2022*, 2890761. [CrossRef]
5. Liang, H.; He, E.; Zhao, Y.; Jia, Z.; Li, H. Adversarial attack and defense: A survey. *Electronics* **2022**, *11*, 1283. [CrossRef]
6. Zhao, W.; Alwidian, S.; Mahmoud, Q.H. Adversarial Training Methods for Deep Learning: A Systematic Review. *Algorithms* **2022**, *15*, 283. [CrossRef]
7. Rice, L.; Wong, E.; Kolter, Z. Overfitting in adversarially robust deep learning. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; pp. 8093–8104.
8. Garaev, R.; Rasheed, B.; Khan, A. Not So Robust After All: Evaluating the Robustness of Deep Neural Networks to Unseen Adversarial Attacks. *arXiv* **2023**, arXiv:2308.06467.
9. Gao, Y.; Shumailov, I.; Fawaz, K.; Papernot, N. On the limitations of stochastic pre-processing defenses. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24280–24294.
10. Carlini, N.; Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017; pp. 3–14.
11. Rasheed, B.; Khan, A. Improving Robustness of Deep Networks Using Cluster-Based Adversarial Training. *Russ. Law J.* **2023**, *11*. [CrossRef]
12. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 99–112.
13. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
14. Xu, W.; Evans, D.; Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv* **2017**, arXiv:1704.01155.
15. Nesti, F.; Biondi, A.; Buttazzo, G. Detecting adversarial examples by input transformations, defense perturbations, and voting. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 1329–1341. [CrossRef] [PubMed]
16. Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; Yuille, A. Mitigating adversarial effects through randomization. *arXiv* **2017**, arXiv:1711.01991.
17. Rasheed, B.; Masood Khattak, A.; Khan, A.; Protasov, S.; Ahmad, M. Boosting Adversarial Training Using Robust Selective Data Augmentation. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 89. [CrossRef]
18. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [CrossRef]
19. Van Der Maaten, L.; Postma, E.; Van den Herik, J. Dimensionality reduction: A comparative. *J. Mach. Learn. Res.* **2009**, *10*, 66–71.
20. Cohen, I.; Huang, Y.; Chen, J.; Benesty, J.; Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4.
21. Sedgwick, P. Spearman's rank correlation coefficient. *BMJ* **2014**, *349*, g7327. [CrossRef] [PubMed]
22. Hershey, J.R.; Olsen, P.A. Approximating the Kullback Leibler divergence between Gaussian mixture models. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing—ICASSP'07, Honolulu, HI, USA, 15–20 April 2007; Volume 4, pp. IV-317–IV-320.
23. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138. [CrossRef] [PubMed]
24. Saxe, A.M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B.D.; Cox, D.D. On the information bottleneck theory of deep learning. *J. Stat. Mech. Theory Exp.* **2019**, *2019*, 124020. [CrossRef]
25. Yu, Y.; Chan, K.H.R.; You, C.; Song, C.; Ma, Y. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9422–9434.

26. LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 1 June 2023).
27. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 1 June 2023).