IMPROVING ROBUSTNESS OF DEEP NETWORKS USING CLUSTER-BASED ADVERSARIAL TRAINING

[1]BADER RASHEED, [1,2]ADIL KHAN

[1] Machine Learning and Knowledge Representation Lab, Innopolis University, Innopolis, Russia ,[2] School of Computer Science, University of Hull, HU67RX, Hull, UK

[1]b.rasheed@innopolis.university,[2] a.khan@innopolis.ru

***Abstract*—** *Deep learning models have been found to be susceptible to adversarial attacks, which limits their use in security-sensitive applications. One way to enhance the resilience of these models is through adversarial training, which involves training them with intentionally crafted adversarial examples. This study introduces the idea of clustering-based adversarial training technique, with preliminary results and motivations. In this approach, rather than using adversarial instances directly, they are first grouped using various clustering algorithms and criteria, creating a new structured space for model training. The method's performance is evaluated on the MNIST dataset against different adversarial attacks, such as FGSM and PGD, with an examination of the accuracy-robustness trade-off. The results show that cluster-based adversarial training could be used as a data augmentation method to enhance the generalization in both clean and adversarial domains.*

Index Terms—*Deep neural networks, Adversarial attacks, robustness, adversarial training.*

I. INTRODUCTION

The amount of available public and private data that is currently accessible has grown significantly during the last decade. Deep neural networks (DNNs) have been effectively used to handle difficult image and natural language processing tasks as a result of advancements in computer power. Nevertheless, these achievements have their own limitations. Modern DNNs are known to be extremely susceptible to adversarial examples [1] [8] [7] . The trained model can be tricked by these subtle but malicious perturbations of the network input to make mistaken predictions with a high degree of confidence, and some perturbations can even mislead multiple network models, which means that these attacks are transferable among various models. It is crucial to defend against adversarial attacks because they could have severe effects in important fields like as healthcare, banking, and the security.

In the race between adversarial attacks and defenses against them, many adversarial attacks and defenses were proposed. So far, adversarial training is the most effective approach to mitigate the effect of adversarial attacks [7]. Training the DNN with perturbed versions of the original samples makes it more robust against these attacks. Nevertheless, because each sample is often created with several steps in the gradient's direction while the model is trained, creating adversarial examples during training can be quite computationally intensive. Moreover, adversarial

training typically causes a decrease in the clean accuracy, or the accuracy on clean samples. According to recent studies, the robustness-accuracy trade-off depends heavily on the distribution and quality of the data.

In this work, we propose a new extension for adversarial training using adversarial samples clustering. From each mini-batch composed of both clean and adversarial samples, The suggested data selection technique uses adversarial samples clustering to choose the most relevant adversarial samples from the training set for adversarial training. The training time is shortened since only the chosen samples are utilized to update the model parameters. The choice also establishes a more reasonable compromise between the required number of clean and adversarial samples for acceptable robustness and benchmark accuracy.

The paper is organized as follows. In section 2, we discussed potential clustering algorithms to be used. Section 3 discusses how we can reduce the dimension of the data before clustering. Then, in section 4, we discuss the adversarial attacks used in this study. Section 6 shows different algorithms and analysis for the clustering-based adversarial training. And finally, section 7 concludes this paper.

II. DATA CLUSTERING

Clustering is a technique used in unsupervised machine learning and data analysis, aimed at grouping similar data points together based on their features or characteristics [10]. The primary objective of clustering is to identify underlying patterns or structures within a dataset by partitioning it into clusters, such that data points within the same cluster share similarities, while data points from different clusters exhibit distinct differences. This process aids in understanding complex data, simplifying large datasets, and uncovering hidden relationships or structures.

Several challenges can arise when applying clustering techniques. One major challenge is determining the optimal number of clusters, which may not be known in advance. Additionally, the choice of similarity or distance metrics can significantly influence the resulting cluster assignments. The presence of noise, outliers, or irrelevant features can also negatively impact clustering performance. Furthermore, scalability is a concern when dealing with large datasets, as some clustering algorithms can be computationally expensive.

Three widely-used clustering algorithms are k-means, hierarchical clustering, and DBSCAN, each with its own strengths and weaknesses. K-means is a centroid-based clustering algorithm that aims to partition a dataset into k distinct, non-overlapping clusters. K-means is relatively simple, easy to implement, and computationally efficient for large datasets. However, it is sensitive to the initial placement of centroids, can be prone to local minima, and requires specifying the number of clusters (k) in advance. Hierarchical clustering is an approach that builds a nested sequence of clusters by either merging (agglomerative) or splitting (divisive) clusters at each step. This method does not require specifying the number of clusters a priori and allows for visual inspection to determine an appropriate cut-off point. However, hierarchical clustering can be computationally expensive for large datasets. Finally,

DBSCAN is a density-based clustering algorithm that groups data points based on their proximity and density. It identifies clusters as dense regions of data points separated by areas of lower point density. DBSCAN is particularly suitable for detecting clusters of arbitrary shapes and can handle noise and outliers effectively. It does not require specifying the number of clusters in advance as well, but it can be sensitive to the choice of parameter values, which may impact the resulting clusters.

III. DIMENSIONALITY REDUCTION

Typically, clustering algorithms use distance measures to group data points that are close to each other in the same cluster and far apart points in different clusters. However, distance measures do not work well in high-dimensional spaces where data usually exists, hence, dimensionality reduction is usually done to make the distance metric reasonable.

In our work, we investigate the effectiveness of two algorithms for dimensionality reduction, PCA and t-SNE, for clustering based adversarial training of deep models to improve their robustness. PCA is an unsupervised linear method that reduces highly correlated data by transforming the original vectors into a new set of principal components while retaining as much information as possible. In contrast, t-SNE is a non-linear method that preserves the local structure of the data by minimizing the KL divergence between the two distributions of the higher and lower dimensions.

IV. ADVERSARIAL ATTACKS AND DEFENSES

The vulnerability of DL models to adversarial attacks was first introduced by [9]. After that, Fast Gradient Sign Method (FGSM) attack was introduced in [1], but sometimes its success rate is low. Therefore, an iterative FGSM (I-FGSM) was proposed by [3] where the loss function is increased in multiple small steps instead of one large step.

Subsequently, many adversarial attacks algorithms were proposed including the basic iterative method (BIM), projected gradient descent (PGD) [3], Carlini and Wagner (C&W) attacks [2], and others.

Meanwhile, many defenses against adversarial attacks were introduced recently, including heuristic and certificated defense [4]. Heuristic defense refers to a defense that increases the robustness of the model against specific attacks without giving theoretical guarantees. The most effective heuristic defense is Adversarial Training (AT) [1], which augments the training data with adversarial samples generated by the previously mentioned attacks.

Empirically, PGD adversarial training achieves the best accuracy against a wide range of adversarial attacks on several datasets [5]. On the other hand, certified defenses can provide a guarantee for their lowest accuracy under a pre-defined group of adversarial attacks. A popular certified defense is to formulate an adversarial polygon and to convexly relax the problem to define its upper bounds [6]. This upper bound guarantees that no attack with specific limitations can surpass the certificated attack success rate. However, these defenses are still restricted to

small datasets and small models, making them inapplicable to real life scenarios.

Several attack methods have been introduced in literature to find x^{adv}. In our work, we use two of such techniques. The simplest method is Fast Gradient Sign Method (FGSM) [1] . It tries to maximize the loss function by finding the gradients of the loss with respect to the input sample and update along the direction of the gradient with a restriction on the L_∞ norm of perturbation so that the difference between adversarial and clean sample is imperceptible. Mathematically:

$$x^{adv} = x + \varepsilon * sign(\nabla_x J(h(x), y_{true})) \quad s.t: \|x^{adv} - x\|_\infty \leq \Delta. \quad (1)$$

The second method that we employ is a stronger iterative attack called the Projected Gradient Descent (PGD) [3]. It is similar to FGSM, but runs for multiple iterations. It creates iterative perturbations as:

$$x_0^{adv} = x \quad (2)$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha * sign(\nabla_{x_t^{adv}} J(h(x_t^{adv}), y_{true})) \quad (3)$$

$$x_{t+1}^{adv} = clip(x_{t+1}^{adv}, x_{t+1}^{adv} - \varepsilon, x_{t+1}^{adv} + \varepsilon) \quad (4)$$

where *t* determines the iteration number, $\varepsilon$is the attack step or the attack learning rate, and *clip(input, a, b)* restricts the adversarial sample to reside in the range *[a, b].*

The key takeaway is that AT is so far the most effective and scalable approach to increase the robustness of deep models. However, this process can sometimes lead to a trade-off between robustness and accuracy, as the model becomes more resistant to adversarial examples but may experience decreased performance on clean or non-adversarial data. This trade-off occurs because adversarial training focuses on learning a decision boundary that is robust against adversarial perturbations, which may not necessarily align with the optimal decision boundary for clean data. Consequently, the model's performance on clean data may suffer as it prioritizes robustness against adversarial examples. Accordingly, this work proposes to reduce the influence of adversarial samples during training by using a clustering based adversarial training approach described below.

V. THE OVERALL APPROACH

Our approach stems from the point that classic Adversarial Training (AT) might depend on samples that are not necessarily useful for increasing the robustness of deep neural networks. These samples might even reduce the efficiency of AT and increases the robustness-accuracy trade-off. However, choosing some adversarial samples over others is a complicated task. This is where clustering becomes useful. Clustering techniques by nature choose samples that are close to each other to perform one cluster. This also helps in reducing the number of adversarial samples required for AT.

Our approach works according to these steps:

- For the training samples, we first normalize the data into [0-1] range. This makes dimensionality reduction and clustering more efficient,
- Reduce the dimension of the adversarial samples to two dimensions,
- Cluster the data, and choose one random cluster from the resulting clusters,

- Augment the training data with adversarial samples from the previous cluster
- Measure the clean and adversarial accuracy of the model on test unseen data after training.

VI. EXPERIMENTS

In every training iteration, we train the model on the clean data, then we train it on the adversarial data for one iteration. Ideally, the model should be trained on the fly on both adversarial and clean data, but since we have many algorithms with many parameters to analyze, we train the model on the adversarial samples only for one epoch.

We consider L_∞ as a measure of perturbation in the attacks. The experiments were implemented on a single GeForce GTX 1080 Ti.

For each of the clustering algorithms(Kmeans, gglomerative Clustering, DBSCAN), we augment the training data with adversarial data coming from only one cluster which is chosen randomly from the available clusters. So, first we generate adversarial samples, then we cluster the adversarial samples and choose adversarial samples from one random cluster and augment the training data with these samples to perform adversarial training. The number of clusters for Kmeans and gglomerative Clustering are selected between 1 and 10. DBSCAN does not require defining the number of clusters, so we report only the best results of DBSCAN for comparison with other algorithms in table 3.

VII. RESULTS ON MNIST

The allowed adversarial perturbation , in this case, is 0.5, and the maximum number of iterations for PGD is 20. The results of both Kmeans and Agglomerative Clustering with number of clusters between 1 and 10 are reported in Table 1 for fgsm attack and in Table 2 for pgd attack and with PCD applied as dimensionality reduction technique before clustering.

The initial clean accuracy for the model is 97.01 % and the adversarial accuracy on fgsm is 5.05% and on PGD is 2.44%. The clean accuracy for original adversarial training is 96.99 and adversarial accuracy on fgsm is 30.16% and on PGD is 28.88%.

Table 1: The results of Kmeans and Agglomerative Clustering with PCA dimension reduction on fgsm attack

cls number	clean accuracy km	adv accuracy km	clean accuracy agg	adv accuracy agg	avg accuracy km	avg accuracy agg
1	0.970	0.299	0.966	0.339	0.634	0.652
2	0.969	0.306	0.964	0.329	0.637	0.646
3	0.968	0.257	0.970	0.331	0.612	0.650
4	0.968	0.228	0.970	0.358	0.598	0.664
5	0.968	0.247	0.965	0.320	0.607	0.642
6	0.966	0.273	0.969	0.317	0.619	0.643
7	0.967	0.261	0.965	0.306	0.614	0.635
8	0.971	0.356	0.965	0.216	0.663	0.590

9	0.970	0.343	0.962	0.244	0.656	0.603
10	0.969	0.354	0.970	0.298	0.661	0.634

In figures 1, 2 we see the comparison between kmeans and Agglomerative Clustering in terms of clean accuracy and adversarial accuracy for different cluster number configurations for both fgsm and pgd attacks. We also plot the average accuracy (Average of clean and adversarial accuracy) to better estimate the performance of each method. The best average accuracy for kmeans is achieved when the number of clusters is 4, while the best average accuracy for Agglomerative Clustering is achieved when the number of clusters is 7.

Table 2: The results of Kmeans and Agglomerative Clustering with PCA dimension reduction on PGD attack

cls number	clean accuracy km	adv accuracy km	clean accuracy agg	adv accuracy agg	avg accuracy km	avg accuracy agg
1	0.973	0.390	0.971	0.316	0.681	0.643
2	0.973	0.296	0.971	0.322	0.634	0.646
3	0.972	0.357	0.974	0.345	0.664	0.659
4	0.973	0.374	0.973	0.381	0.673	0.677
5	0.974	0.340	0.971	0.404	0.657	0.687
6	0.970	0.367	0.970	0.368	0.668	0.669
7	0.968	0.298	0.974	0.444	0.633	0.709
8	0.972	0.412	0.969	0.377	0.692	0.673
9	0.972	0.323	0.975	0.344	0.647	0.659
10	0.969	0.317	0.971	0.351	0.643	0.661

We can see from the tables and figures how clustering the adversarial samples help in increasing both clean and adversarial accuracy and help to reduce the gap between the two accuracies. It is worth noting here that both algorithms overcome classic adversarial training even when the number of clusters is 1. The reason is that clustering the samples before training works as a filtering technique to filter irrelevant samples that are far from the cluster center.

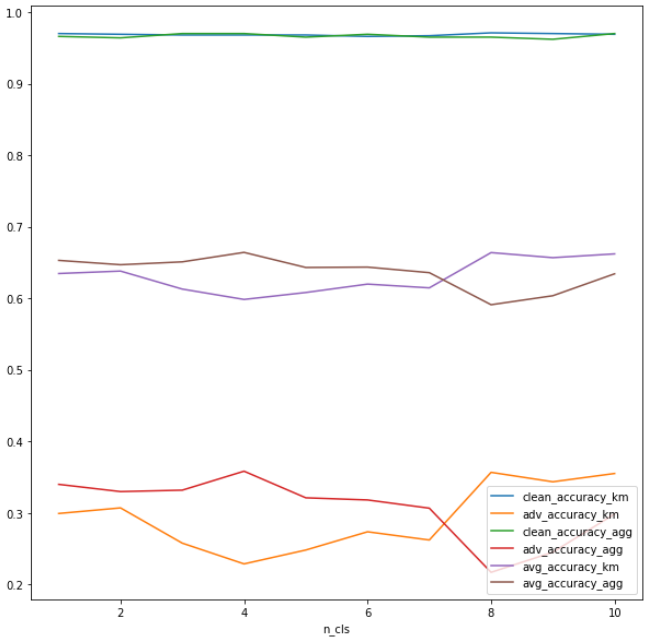

Figure 1: Kmeans vs Agglomerative Clustering for fgsm attack

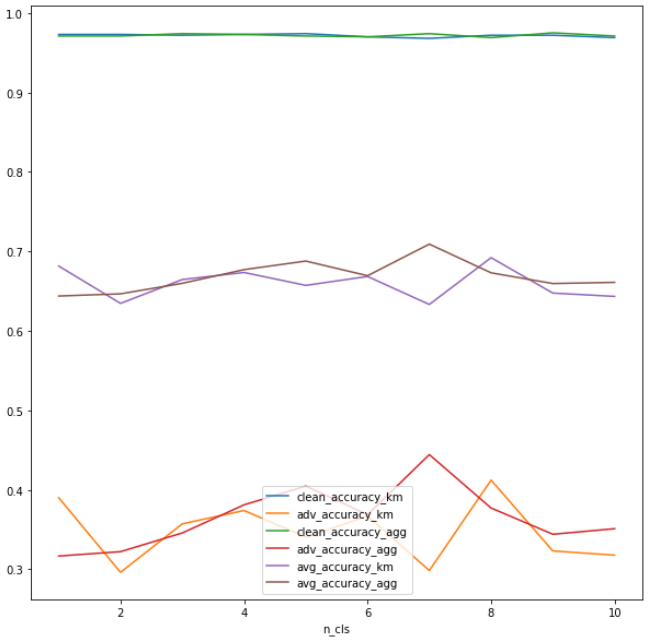

Figure 2: Kmeans vs Agglomerative Clustering for pgd attack

VIII. THE EFFECT OF DIMENSIONALITY REDUCTION

To study the effect of dimensionality reduction, we show the clean and adversarial accuracy of three configuration: 1. None: no reduction applied where we apply clustering on the initial dimentions of the data 2. T-SNE: we apply t-sne reduction to two dimensions 3. PCA: we apply pca reduction to two dimensions. Figure 3 shows the results using kmeans clustering and fgsm attack. Dimensionality reduction not only reduces the time of execution for the clustering algorithms, but also leads to better clean and adversarial accuracy. It is also clear that t-sne is more suitable than PCA for our task.

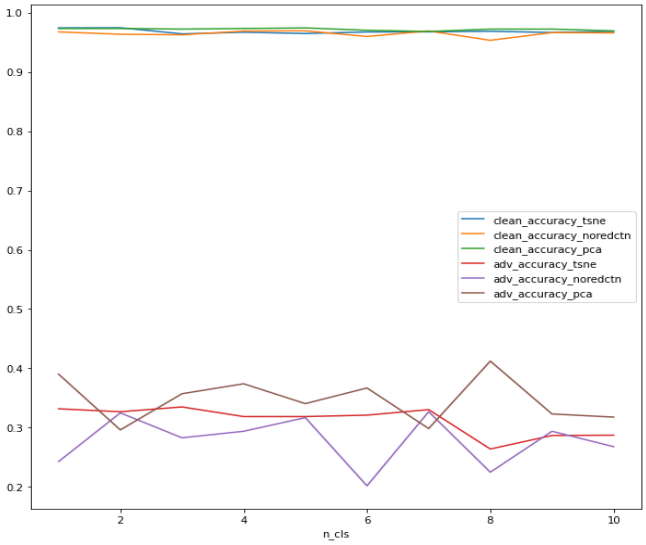

Figure 3: The effect of dimensionality reduction

IX. OVERALL COMPARISION

To see which algorithm works better, we summarize the highest adversarial and clean accuracy achieved for each clustering algorithm under the fgsm attack. The results are summarized in table 3 with comparison with classic and adversarial training.

The results show interestingly that DBSCAN not only overcome other clustering algorithms in both adversarial and clean accuracy, it also overcomes that classic training on clean accuracy. Which means that cluster-based adversarial training could be used as a data augmentation method to enhance the generalization in both clean and adversarial domains.

Table 3: Overall comparision

Algorithm	Clean Accuracy	Adversarial accuracy
Classic training	0.971	0.50
Adversarial training	0.969	0.301
Kmeans	0.971	0.356
gglomerative Clustering	0.970	0.358
DBSCAN	0.973	0.531

CONCLUSION

In this paper, we studied enhancing neural networks robustness through adversarial cluster-based training. Combination of different clustering algorithms and different dimensionality reduction techniques are used in order to cluster adversarial attacks before adversarial training. The proposed extension for adversarial training leads to better adversarial and clean accuracy. Various experiments were performed as a proof of concept and the results prove the

applicability of the proposed method.

REFERENCES

[1] Alejandro Barredo Arrieta, Natalia D´ıaz-Rodr´ıguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garc´ıa, Sergio Gil-L´opez, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion, 58:82–115, 2020.

[2] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In Proceedings - IEEE Symposium on Security and Privacy, pages 39–57, CW, 2017.

[3] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In Artificial intelligence safety and security, pages 99–112. Chapman and Hall/CRC, 2018.

[4] Hongshuo Liang, Erlu He, Yangyang Zhao, Zhe Jia, and Hao Li. Adversarial Attack and Defense: A Survey. 2022.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings. International Conference on Learning Representations, ICLR, jun 2018.

[6] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018.

[7] Bader Rasheed, Adil Khan, Muhammad Ahmad, Manuel Mazzara, SM Kazmi, et al. Multiple adversarial domains adaptation approach for mitigating adversarial attacks effects. International Transactions on Electrical Energy Systems, 2022, 2022.

[8] Bader Rasheed, Adil Khan, SM Ahsan Kazmi, Rasheed Hussain, Md Jalil Piran, and Doug Young Suh. Adversarial attacks on featureless deep learning malicious urls detection. Computers, Materials and Continua, 68(1):921–939, 2021.

[9] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, 2014.

[10] Rui Xu and Donald Wunsch. Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3):645–678, 2005 J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility,” IEEE Trans. Electron Devices, vol. ED-11, pp. 34-39, Jan. 1959.

[11] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, and M. Miller, “Rotation, scale, and translation resilient public watermarking for images,” IEEE Trans. Image Process., vol. 10, no. 5, pp. 767-782, May 2001.