Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 5, pp. 1523–1538
DOI : 10.47974/JDMSC-1935

Detection of adversarial attacks against security systems based on deep learning model

Mohanad J. Jaber [†]
Department of Basic Science
College of Dentistry
Kufa University
Najaf
Iraq

Zahraa Jasim Jaber *
Ahmed J. Obaid [§]
Department of Computer Science
Faculty of Computer Science and Mathematics
Kufa University
Najaf
Iraq

Abstract

Many real-world problems can be solved using deep learning, including image processing and speech recognition. However, deep learning and neural networks are usually black boxes, making them vulnerable to threats from abnormal examples, such as adversarial and backdoor examples crafted by attackers with malicious intentions and, even worse, unintentionally mislabeled examples. The detection of such abnormal examples is therefore urgent and important. Various approaches have been used to detect specific types of abnormal examples, but all have limitations. Despite these limitations, this problem remains a significant one. To make AdvGAN more suitable for adversarial training, we propose updating its loss terms and testing it against various attacks. In contrast to generative models with no adversarial training, generative models with adversarial training are better at boosting adversarial robustness. Experimental results demonstrate a high true-positive rate (98.11%) and a low false-positive rate (3%).

Subject Classification: *68P25, 68P01.*

Keywords: *DDO attacks, Security channel, Attacks detection, Adversarial attacks.*

[†] *E-mail:* **mohanad.altalqani@uokufa.edu.iq**
* *E-mail:* **zahraaj.altalakany@uokufa.edu.iq (Corresponding Author)**
[§] *E-mail:* **ahmedj.aljanaby@uokufa.edu.iq**

© Taru Publications

1. Introduction

Many scenarios and tasks are being performed using deep neural networks (D, although issues security issues are constantly drawing attention [1]. In particular, carefully designed adversarial examples have a more obvious effect on deep neural network models [2]. Therefore, when a small disturbance is added to an image, an adversarial example attack problem can be used to classify errors. Small perturbations are introduced to deep learning models as adversarial examples to confuse model judgment without humans' awareness. Adversarial example attacks can be used for a variety of purposes, including target identification, automatic driving, face recognition, and intrusion detection systems (IDS), among others [3], [4]. In an experiment [5], the author used mobile phone photos as adversarial examples to test the classifier and defeat the adversarial example attacks. Using the author's design [6] created a real spectacle frame that fools facial recognition software whenever the wearer wears it. The author used a Faster Region-CNN and a YOLO 9000 target detector to achieve physical confusion [7]. There were adversarial stickers on either side of the STOP sign in almost every frame of the video that prevented YOLO detectors from detecting the STOP sign.

The adversarial machine learning method attacks deep learning models with adversarial examples. A deep learning model is designed to misclassify input data using adversarial examples with slight modifications to input data [8], [9]. Due to these subtle modifications, human observers find it hard to distinguish an adversarial image from the original. White and black box attacks can be classified according to the attacker's knowledge [7]. Attackers who conduct white box attacks can access everything associated with trained DNNs, such as training data, model architectures, hyperparameters, and model weights. As opposed to black box attacks, which do not reveal any information about DNNs except for the outputs, attackers using black box attacks know nothing else about them. A targeted and non-targeted attack can be differentiated based on attack targets [7]. In targeted attacks, adversarial examples are misclassified into a target class by misguiding DNNs. It is possible to incorrectly classify adversarial examples, except for the correct class, into arbitrary classes, unlike non-targeted attacks. Attacks that target a specific area will likely be more difficult than those that do not target a specific area.

2. Related Work

The first is to improve classifier robustness against current adversarial attacks, while the second is to detect real adversarial samples. According to the author [10], an additional adversarial term in the training loss function would improve the robustness of these models against specific adversarial attacks. A defence distillation [11] can also be seen as a way to prevent too much dependence on the data by training a DNN with the softmax outputs of another neural network. Defending against attacks using defensive distillation [12]. However, the author [13] demonstrated that a modified attack can easily defeat defensive distillation.

The concept of augmented DNNs, trained on normal and adversarial samples, has been proposed [14] as a solution for detecting adversarial samples. However, using generated adversarial samples requires the detector subnetwork to be trained contrary to the authors' compelling performance results on state-of-the-art adversarial attacks. It implicitly trains the detector on only some of the possible adversarial attacks; however, we cannot determine just how comprehensive this subset is, and future attacks could overcome the system in the future. This technique is not currently known to be robust to random noise.

The Modified National Institute of Standards and Technology (MNIST) handwritten digit database is an essential resource for pattern recognition research and has great practical value. It is possible to categorize handwriting using statistical and structural feature-based methods [15], [16]. Unlike the former, a handwritten pointer's density and area can be adjusted to mitigate irregular handwriting caused by features other than beginning and end points, intersections, contours, and unevenness. There are two primary reasons why traditional image processing algorithms using static feature analysis at pixel level fail in MNIST handwriting recognition: 1) in Arabic numerals, there is a lack of glyph information, and some features may be shared among numbers; 2) Depending on the situation, certain features may only appear in specific but rare situations when handwriting varies from person to person. The stroke trajectory point, for example, can provide dynamic information in the future to address these challenges [17], [19]. Machine Learning (ML) in handwriting classification has gained popularity in recent years thanks to its outstanding generalization capabilities and performance. The accuracy of classification and recognition has been enormously improved by combining typewritten features [16], [18], k-NN, HOG, features, SVM [20], [21] and clustering [22], [23], statistical classification models [24], [25-35].

2.1 *Defense of Adversarial Attack*

Detection of adversarial examples can be accomplished using two conventional methods: the modification of the classifier approach [2], [10], [11] and the modification of the input data approach [36], [37]. There are two types of classifier modifications: defensive distillation and adversarial training [2], [10] [11]. Introducing adversarial training by [2] involves introducing adversarial examples into the learning process. To increase resistance against adversarial examples,[38] proposed using multiple local models in ensemble adversarial training. Contrary to popular belief, adversarial training does not always lead to decreased classification accuracy. According to [11], defensive distillation prevents adversarial examples from being calculated using gradient descent. Classifier output class probabilities are used to block attack gradients in the second stage of classifier training. The structure also needs improvement, and this technique makes white box attacks possible.

There are two conventional methods for modifying input data: filtering [36], [37], squeezing [39], and Magnet techniques [40]. In [37], the author proposed generating adversarial nets to eliminate adversarial perturbation [7]. A filtering module must be created in addition to the original sampling procedure to maintain classification accuracy. This technique, referred to as feature squeezing, modifies input samples as proposed by the author [39]. It reduces the depth of each pixel by reducing the difference between each pair of corresponding pixels. Changing the input data requires a separate procedure. A new defense technique called Magnet [40] combines multiple defense techniques in one. An adversarial example attack can be resisted by Magnet, which was proposed by [40]. The input data is reformatted to fit the closest original sample when an adversarial example is detected by measuring far away distances. As the Magnet technique is vulnerable to white box attacks, separate processes are required to create the detector and reformer modules.

Changing the classifier or modifying the input data is required for each defence method. On the other hand, a new detection method we propose does not require extra learning but rather relies on patterns indicative of adversarial classification scores.

3. Methodology

A feature space $X \subseteq R^n$ is assumed, and a number n is assumed. Suppose there are n features in the feature space. A training set would be

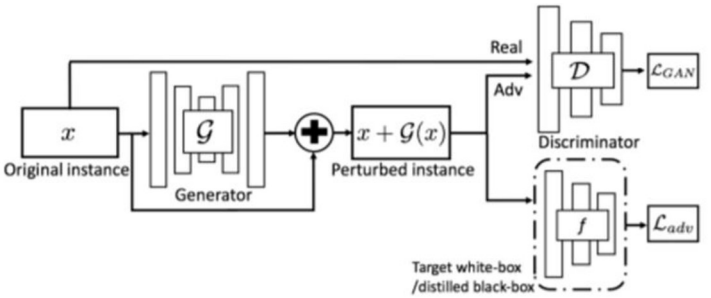

Figure 1

Architecture of AdvGAN

composed of x_i, y_i instances, where $x_i \in X$ represents a feature vector generated according to some unknown distribution, while $y_i \in Y$ represents the true class labels associated with the x_i, y_i Instances. $|Y|$ denotes the number of possible classification outputs, with $f : X \to Y$ being the classifier learned from *X* to *Y*. In order to generate adversarial examples x_A, an adversary must present an instance *x* as either $f(x_A) \neq y$ (untargeted attack) or $f(x_A) = t$ (targeted attack), where *t* is the class of the target. A similar distance metric, such as L_2, needs to be used for x_A in comparison with the original instance *x*.

3.1 *AdvGAN Model*

AdvGAN's overall architecture is illustrated in Figure 1, which consists primarily of three parts: the generator *G*, the discriminator *D*, and the target neural network *f*. As input, the generator *G* generates the perturbation $G(x)$ based on the original instance *x*. A discriminator $x + G(x)$ will analyze $x + G(x)$ to distinguish generated data from the original instance. *D* encourages indistinguishability between the generated instance and the data it is derived from. Using a white-box attack, we can fool a learning model, in this case, *f*is the target model. Input $x + G(x)$ is taken as input, and output is the L_{adv} of loss, which indicates how far away the prediction is from the target class *t* (targeted attack) or how far away it is from the ground truth class (untargeted attack).

According to [41], the adversarial loss is as follows:

$$L_{GAN} = E_x \log D(x) + E_x \log(1 - D(x + G(x))) \quad (1)$$

Using the discriminator *D*, we propose a method for distinguishing perturbed data $x + G(x)$ from the original data *x*. In order to generate

instances that are closest to the original data, the real data is sampled from the true class.

Using a targeted attack, we lose if we fool the target model *f*:

$$L_{adv}^{f} = E_x l_f(x + G(x), t) \tag{2}$$

The L_{adv}^{f} loss causes the perturbed image to be classified as target class *t*, where *t* denotes the target class and l_f indicates the loss function used to train the model *f*.

We can also focus on the targeted attack by maximizing the distance between predictions and ground truth, but in the rest of this paper, we will examine the untargeted attack.

Our previous work shows that a soft hinge loss is added to the L_2 norm to bound its magnitude [13], [42].

$$L_{hinge} = E_x \max(0, \| G(x) \|_2 - c) \tag{3}$$

In this case, *C* denotes a bound that the user specifies. The GAN's training can also be stabilized using this method. As a final point, we can state our full objective as follows:

$$L = L_{adv}^{f} + \propto L_{GAN} + \beta L_{hinge} \tag{4}$$

Each objective's relative importance is determined by α and β. L_{adv}^{f} is used here to generate adversarial examples, optimizing for high attack success rates while L_{GAN} is used to resemble the perturbed data. Solving the min-max arg arg $\min_G \max_D L$ gives us our G and D. L_{adv}^{f} Upon training on both training data and target models, L_{adv}^{f} can perform semi-white box attacks on any input instance.

Fast Gradient Sign Method (FGSM): For finding adversarial samples quickly, the author proposed a gradient sign method [10]. The ReLU activation function is usually used in existing DNNs. DNNs can be trained using these functions because they are easily optimized. This model type propagates changes made to the input to the later hidden layers until the output layer. It can lead to misclassification due to the accumulation of input errors. This analysis leads to the FGSM attack assuming the same strength for all dimensions.

This equation can be used to generate an adversarial sample informally:

$$x' = x - \varepsilon.sign(\nabla_x J(F(x)))$$

A cost function is used to train the model $J(\cdot)$, where ∇ represents the gradient. The pixels are changed simultaneously along the gradient direction at a fixed scale. Due to its ability to calculate gradient signs by backpropagation, it can easily create adversarial samples.

Basic Iterative Method (BIM): A modified version of the FGSM attack is the basic iterative method [5]. In each iteration of the algorithm, an adversarial sample is generated and updated with this equation:

$$X'_i = x'_{i-1} - clip_f(\alpha.sign(\nabla_x J(F(x'_{i-1}))))$$

A correctly classified input is represented by x'_0. The gradient sign is followed with a single step in FGSM. In each iteration of the BIM process, clipping is performed to ensure the results are within the original input's L_∞ ε – neighborhood. The Iterative FGSM (IFGSM) is also known as such.

4. Result and Analysis

Our comprehensive evaluation of the proposed model is presented in this section, consisting of evaluations of the number of queries (QUE), focusing on training, testing accuracy, and loss estimation. A multi-adversarial attack evaluation was conducted on the proposed model against multiple adversarial attacks, including the FGSM, the JSMA, the CWL2, the LINFBI, the LINFPGD, and the ENL1. During the experiments, the Intel i7 running on 16 GB of RAM is paired with an NVIDIA GeForce GTX 1660 with 1024 GB of video memory. Models and datasets used in this research include MNIST, a well-known image classification dataset. This collection includes 70K handwritten digits from 0 to 9 in grayscale. Using this method, we randomly picked 100 images from each class (10 x 100 = 1000).

A convolution neural network (CNN) architecture was trained with 70000 images, and a convolution neural network (CNN) was tested with 10000 images, which means training and testing are separated by 80-20%. As part of our evaluation, we implemented and evaluated our proposed model using TensorFlow. Whitebox and blackbox scenarios were considered for evaluation, in which attacks are generated and tested on the same network; in blackbox attacks, they are generated and tested on different networks. As shown in Figure 2, the CNN architecture used for the proposed MNIST digit evaluation is described in detail. In MNIST, normal attacks, FGSM, JSMA, CWL2, ENL1, ST, LINFBI, LINFPGD, and

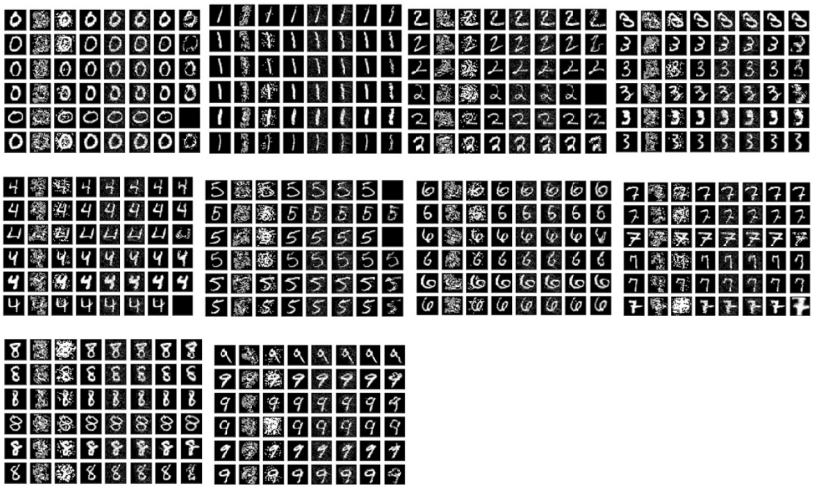

Figure 2

A graphical presentation of the MNIST digits dataset

LINFPGD are compared. It makes estimating gradients more challenging and limits the transfer attack.

4.2 *Experimental Results*

The proposed model is evaluated against adversarial attacks. For example, let us take the blackbox attack scenario, where the attacker relies on the attack's transferable properties since it has no information about the target network. Furthermore, we have considered the case of a whitebox scenario in which attacks are generated by attackers who know about the classification network. We generated adversarial data of different attack types by using the Cleverhans library as a first step to conducting experiments. In Figure 3, various adversarial attacks are shown impacting the MNIST image [43, 44].

In adversarial samples, crafted perturbations are introduced into the normal input data to create a similar adversarial dataset to the normal input data. However, the machine learning model mispredicts the class with a high probability even when the adversarial data looks similar to the normal input data. As far as ML classifiers are concerned, the adversarial samples constitute an optical illusion. As shown in Figure 4, FGSM can be seen as advanced and sophisticated versions of Basic Iterative Method (BIM), Jacobian Saliency Map Attack (JSMA), and [13] techniques.

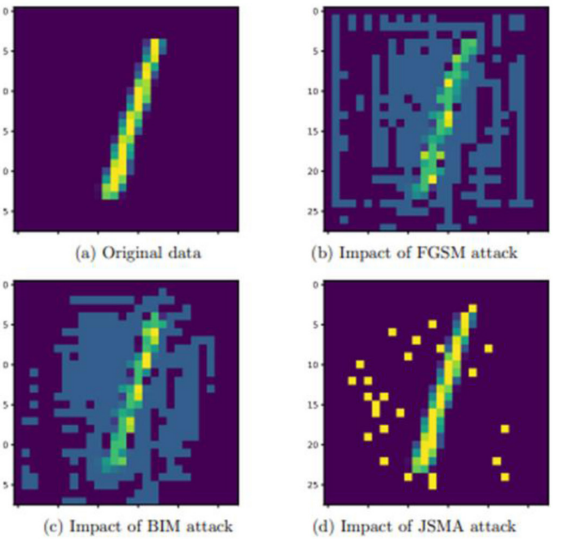

Figure 3

Image comparison with generated adversarial data generated from MNIST

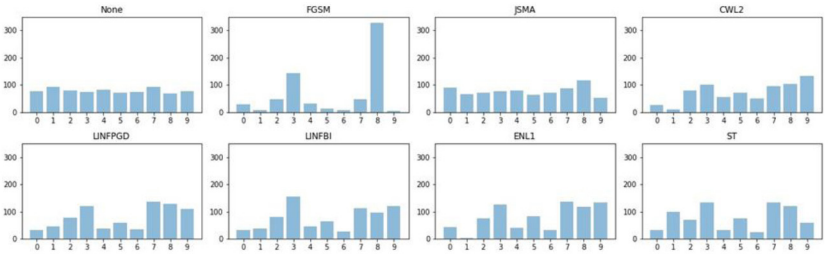

Figure 4

A measure of the average entropy generated by the number of digits in the MNIST dataset for legitimate samples and adversarial examples

Legitimate samples and adversarial examples are analyzed for their entropy. Assuming that the input samples are on the numerical scale [0, 1], we only analyze the successful adversarial generated, which is entropy. Each attack method generates adversarial examples using the MNIST dataset, so we compare the entropy of each validation set with each adversarial example. Using the MNIST dataset as an example, Figure 5 shows the average entropy for legitimate samples and adversarial samples generated by the FGSM, BIM, ENLI, and CW attacks.

We can understand a classifier's behaviour by attacking it. According to Figure 6-7, we can train the model against successful attacks by adding adversarial perturbations batch-wise to the dataset. As a result of this process, the model is theoretically capable of defending against such attacks. An FGSM attack limits the perturbation magnitude of each batch in a dataset using the FGSM attack method. There are several training batches with these examples, and a validation accuracy of 98% was

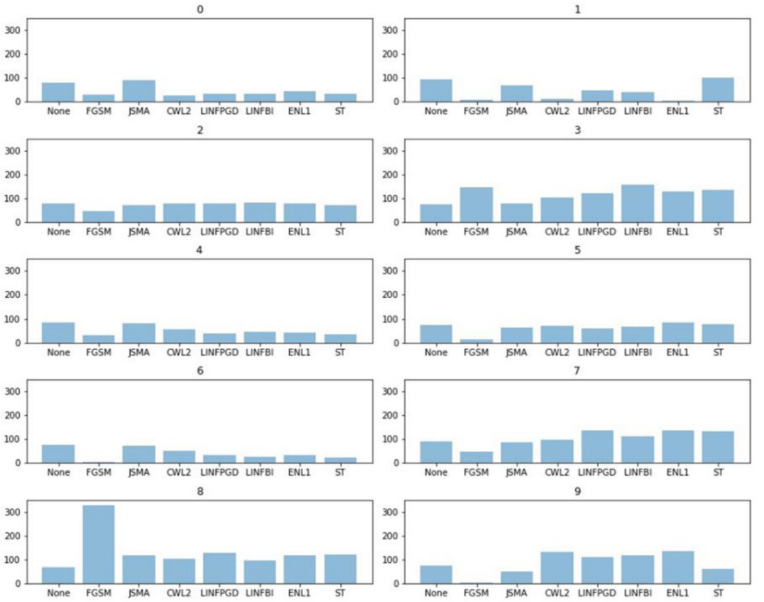

Figure 5

Typical entropy derived from MNIST's dataset for legitimate and adversarial samples

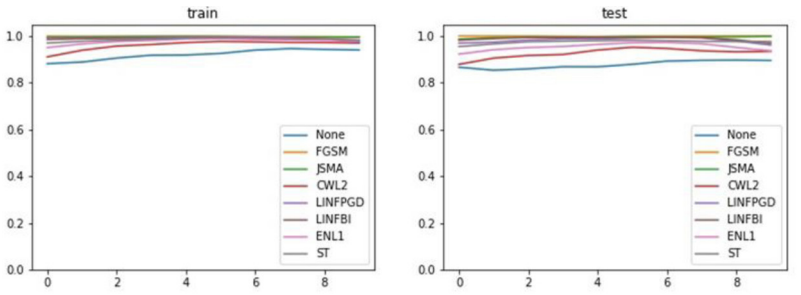

Figure 6

Training and testing accuracy based on the MNIST dataset digits

achieved by training the classifier with these samples. It appears that the FGSM method significantly reduces the accuracy of the adversarial-trained classifier when it is attacked again with the adversarial-trained method.

The accuracy visualizations are presented in Figures 8 and 9 with 50 to 20 perturbations. The proposed model maintains the higher accuracy for the low and high numbers of perturbations.

Figure 8 shows the train, test accuracy and loss metric, which indicate that the detection is effective, especially for large perturbations. Meanwhile, when the perturbation is small, it is hard to detect attacks at both low FNR and FPR, but the proposed model is good at purifying small perturbations, which can make up these shortcuts, as shown in Figure 9. We also test the detection performance when using loss as the indicator, which is shown to be more effective. It can reach both small FNR and FPR, even under small perturbations. In summary, the proposed model can successfully eliminate small perturbations that pass the detection. For larger perturbations that

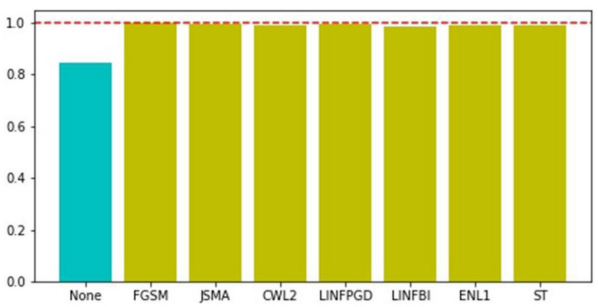

Figure 7

Training and testing accuracy based on the adversarial attacks of the MNIST dataset

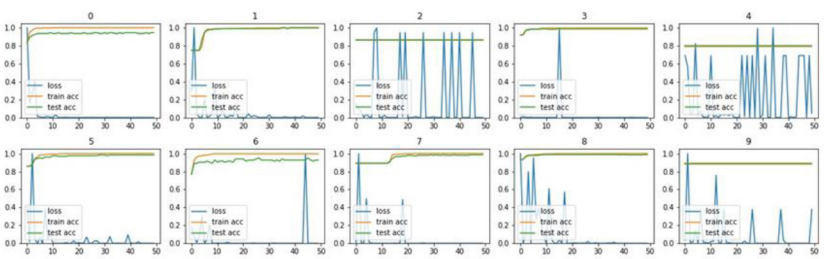

Figure 8

Training, testing accuracy and loss for the numbers of perturbations

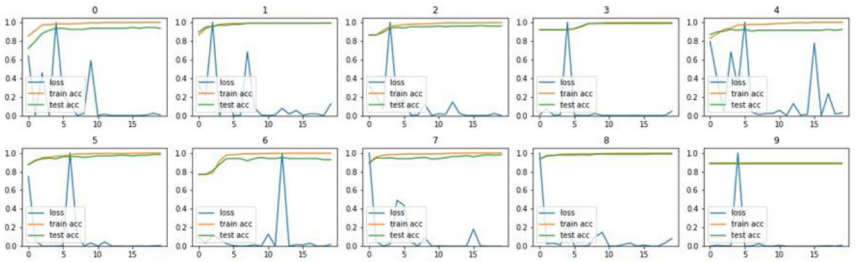

Figure 9

Training, testing accuracy and loss to the numbers of perturbations

are detected as adversarial, they will trigger the searching stage and be purified by the proposed model for achieving higher accuracy.

5. Conclusion

This study developed adversarial attacks and training that can be improved and scaled up to larger deployments in production environments. The behaviour of such adversarial attacks can also be studied for various classification tasks, including natural language processing, video and audio processing, and object detection. As a first step, we need to show motivation for creating robust models by demonstrating adversarial methods. A deep learning model may classify samples incorrectly with a high probability if adversarial attacks generate samples that belong to certain classes. We expect that naturally trained models will achieve small levels of robustness. This method can be applied to future datasets with more complex data manifolds. This implementation would allow AdvGANs to be understood better to boost their adversarial robustness.

References

[1] S. P. Yadav, M. Jindal, P. Rani, V. H. C. De Albuquerque, C. Dos Santos Nascimento, and M. Kuma"An improved deep learning-based optimal object detection system from imag" *Multimed. Tools Appl.*, Sep. (2023), doi: 10.1007/s11042-023-16736-5.

[2] C. Szegedy *et al* "Intriguing properties of neural network" ArXiv Prepr. ArXiv13126199, (2013).

[3] P. Rani and R. Sharm "Intelligent transportation system for internet of vehicles based vehicular networks for smart citi" *Comput. Electr. Eng.*, vol. 105, p. 108543 (2023).

[4] N. Hussain, P. Rani, H. Chouhan, and U. S. Gau"Cyber Security and Privacy of Connected and Automated Vehicles (CAVs)-Based Federated Learning: Challenges, Opportunities, and Open Issu" in *Federated Learning for IoT Applications*, S. P. Yadav, B. S. Bhati, D. P. Mahato, and S. Kumar, Eds., in EAI/Springer Innovations in Communication and Computing., Cham: Springer International Publishing (2022), pp. 169–183. doi: 10.1007/978-3-030-85559-8_11.

[5] A. Kurakin, I. J. Goodfellow, and S. Bengi "Adversarial examples in the physical wor" in *Artificial intelligence safety and security*, Chapman and Hall/CRC, pp. 99–112 (2018).

[6] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reite"Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recogniti" in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria: ACM, pp. 1528–1540, Oct. (2016). doi: 10.1145/2976749.2978392.

[7] Z. Che et al "Adversarial attack against deep saliency models powered by non-redundant prio" *IEEE Trans. Image Process.*, vol. 30, pp. 1973–1988 (2021).

[8] P. Rani, P. N. Singh, S. Verma, N. Ali, P. K. Shukla, and M. Alhassa"An implementation of modified blowfish technique with honey bee behavior optimization for load balancing in cloud system environme" *Wirel. Commun. Mob. Comput.*, vol. 2022, pp. 1–14 (2022).

[9] P. Rani *et al* "Federated Learning-Based Misbehaviour Detection for the 5G-Enabled Internet of Vehicl" *IEEE Trans. Consum. Electron.*, pp. 1–1 (2023), doi: 10.1109/TCE.2023.3328020.

[10] I. J. Goodfellow, J. Shlens, and C. Szeged "Explaining and harnessing adversarial exampl" ArXiv Prepr. ArXiv14126572 (2014).

[11] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swam "Distillation as a defense to adversarial perturbations against deep neural networ" in 2016 IEEE symposium on security and privacy (SP), IEEE, pp. 582–597 (2016).

[12] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swam "The limitations of deep learning in adversarial settin" in 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE, pp. 372–387 (2016).

[13] N. Carlini and D. Wagne "r" "Towards evaluating the robustness of neural networ" in 2017 ieee symposium on security and privacy (sp), Ieee, pp. 39–57 (2017).

[14] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischof "On detecting adversarial perturbatio" ArXiv Prepr. ArXiv170204267 (2017).

[15] R. Herrera-Pereda, A. Taboada Crispi, D. Babin, W. Philips, and M. Holsbach Cost"A Review On digital image processing techniques for in-Vivo confocal images of the corn" *Med. Image Anal.*, vol. 73, p. 102188, Oct. (2021), doi: 10.1016/j.media.2021.102188.

[16] S. R. Low "Pattern Recognition in Handwriti" in *Series on Language Processing, Pattern Recognition, and Intelligent Systems*, vol. 06, WORLD SCIENTIFIC, pp. 77–95 (2021). doi: 10.1142/9789811239014_0005.

[17] S. Tabik, R. F. Alvear-Sandoval, M. M. Ruiz, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, and F. Herrer "MNIST-NET10: A heterogeneous deep networks fusion based on the degree of certainty to reach 0.1% error rate. Ensembles overview and propos" *Inf. Fusion*, vol. 62, pp. 73–80, Oct. (2020), doi: 10.1016/j.inffus.2020.04.002.

[18] S. Wan, S. Ding, and C. Che "Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicl" *Pattern Recognit.*, vol. 121, p. 108146, Jan. (2022), doi: 10.1016/j.patcog.2021.108146.

[19] Y. Wang et al "Improvement of MNIST image recognition based on C" in IOP Conference Series: Earth and Environmental Science, IOP Publishing, p. 012097 (2020).

[20] S. R. Kulkarni and B. Rajendra "Spiking neural networks for handwritten digit recognition—Supervised learning and network optimizati" *Neural Netw.*, vol. 103, pp. 118–127, Jul. (2018), doi: 10.1016/j.neunet.2018.03.019.

[21] V. Nguyen, J. Cai, and J. Ch "Hybrid CNN-GRU model for high efficient handwritten digit recogniti" in Proceedings of the 2nd International Conference on Artificial Intelligence and Pattern Recognition, Beijing China: ACM, pp. 66–71 Aug. (2019). doi: 10.1145/3357254.3357276.

[22] Z. Jan and B. Verm "Multiple strong and balanced cluster-based ensemble of deep learne" *Pattern Recognit.*, vol. 107, p. 107420, Nov. (2020), doi: 10.1016/j.patcog.2020.107420.

[23] W. Xia, X. Zhang, Q. Gao, and X. Ga "Adversarial self-supervised clustering with cluster-specificity distributi" *Neurocomputing*, vol. 449, pp. 38–47, Aug. (2021), doi: 10.1016/j.neucom.2021.03.108.

[24] W. Liang, Y. Wu, M. Li, and Y. Ca "Adaptive multiple kernel fusion model using spatial-statistical information for high resolution SAR image classificati" *Neurocomputing*, vol. 492, pp. 382–395, Jul. (2022), doi: 10.1016/j.neucom.2022.03.062.

[25] S. S. Kadam, A. C. Adamuthe, and A. B. Pati "CNN model for image classification on MNIST and fashion-MNIST datas" *J. Sci. Res.*, vol. 64, no. 2, pp. 374–384 (2020).

[26] Z. Xue, M. Zhang, Y. Liu, and P. D "Attention-based second-order pooling network for hyperspectral image classificati" *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9600–9615 (2021).

[27] J. Chen, X. Lin, Z. Shi, and Y. Liu "Link prediction adversarial attack via iterative gradient atta" *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 4, pp. 1081–1094 (2020).

[28] J. Li, T. Xie, L. Chen, F. Xie, X. He, and Z. Zhen "Adversarial attack on large scale gra"" *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 82–95 (2021).

[29] H. Wang, G. Li, X. Liu, and L. Li "A Hamiltonian Monte Carlo Method for Probabilistic Adversarial Attack and Learni" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1725–1737, Apr. (2022), doi: 10.1109/TPAMI.2020.3032061.

[30] L. Schwinn *et al* "Dynamically sampled nonlocal gradients for stronger adversarial attac" in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–8 (2021).

[31] S. Chen, Z. He, C. Sun, J. Yang, and X. Huan "Universal adversarial attack on attention and the resulting dataset damagen" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2188–2197 (2020).

[32] Y. Yu, X. Gao, and C.-Z. X "Lafeat: Piercing through adversarial defenses with latent featur" in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5735–5745 (2021).

[33] Y. Xiang, Y. Xu, Y. Li, W. Ma, Q. Xuan, and Y. Li "Side-Channel Gray-Box Attack for DN" *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 68, no. 1, pp. 501–505, Jan. (2021), doi: 10.1109/TCSII.2020.3012005.

[34] H. Wang, S. Wang, Z. Jin, Y. Wang, C. Chen, and M. Tistarell "Similarity-based Gray-box Adversarial Attack Against Deep Face Recogniti" in 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), pp. 1–8 Dec. (2021). doi: 10.1109/FG52635.2021.9667076.

[35] Z. Liu, Y. Luo, L. Wu, S. Li, Z. Liu, and S. Z. Li "Are Gradients on Graph Structure Reliable in Gray-box Attack" in Proceedings of the 31st ACM International Conference on Information & Knowledge Management, in CI'22. New York, NY, USA: Association for Computing Machinery, pp. 1360–1368 Oct. (2022). doi: 10.1145/3511808.3557238.

[36] A. Fawzi, O. Fawzi, and P. Frossar "Analysis oclassifiers'' robustness to adversarial perturbations. Co" ArXiv Prepr. ArXiv150202590, (2015).

[37] S. Shen, G. Jin, K. Gao, and Y. Zhan "Ape-gan: Adversarial perturbation elimination with g" ArXiv Prepr. ArXiv170705474 (2017).

[38] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDani"l "Ensemble adversarial training: Attacks and defens" ArXiv Prepr. ArXiv170507204 (2017).

[39] W. Xu, D. Evans, and Y. Q "Feature squeezing: Detecting adversarial examples in deep neural networ" ArXiv Prepr. ArXiv170401155 (2017).

[40] D. Meng and H. Che "Magnet: a two-pronged defense against adversarial exampl" in Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp. 135–147 (2017).

[41] I. Goodfellow *et al.* "Generative adversarial ne" *Adv. Neural Inf. Process. Syst.*, vol. 27 (2014).

[42] Y. Liu, X. Chen, C. Liu, and D. Son "Delving into transferable adversarial examples and black-box attac" ArXiv Prepr. ArXiv161102770, (2016).

[43] Jawad, Zainab A. & Obaid, Ahmed J. A systemic literature overview of Fake News Challenge (FNC-1) dataset and its use in fake news detection schemes, *Journal of Discrete Mathematical Sciences and Cryptography,* 1-10, DOI: 10.47974/JDMSC-1567.

[44] Abdulbaqi, Azmi Shawkat , Muslim, Israa Falih , Al-Ameer, Asraa A. Abd & Obaid, Ahmed J. () Healthcare surveillance based on cloud computing utilizing mobile devices, *Journal of Discrete Mathematical Sciences and Cryptography,* 1-8, DOI: 10.47974/JDMSC-1566.

Received November, 2023