ANALYSIS OF MULTIPLE ADVERSARIAL MACHINE LEARNING ATTACKS ON CONVOLUTIONAL NEURAL NETWORKS

[1]BURCUM EKEN, [2]SEDDIK M. DJOUADI, [3]SELEN AYAS

[1]Department of Electrical Engineering and Computer Science, University of Tennessee, 37919 Knoxville, TN, USA
[2]Department of Electrical Engineering and Computer Science, University of Tennessee, 37919 Knoxville, TN, USA
[3]Department of Computer Engineering, Karadeniz Technical University, Trabzon, Turkey
E-mail: beken@vols.utk.edu, mdjouadi@utk.edu, selenguven@ktu.edu.tr

Abstract - With the increase of machine learning applications, cyber-attacks on these applications have also increased in recent years. Even if the machine learning models are used in many fields including transportation and communications, social media, product recommendations, dynamic pricing, and fraud detection, they are still vulnerable to cyber-attacks. Deep neural networks also face a security threat from adversarial examples which are inputs that appear normal but cause an misclassification by the Deep Neural Network. In this paper, first, electric load data from ERCOT is considered as a signal and then converted to an image. After that, we analyze different gradient-based adversarial attacks on a Convolutional Neural Networks (CNN) model designed to classify whether the data from ERCOT belongs to West Station or Far West Station. We also explore robustness of the CNN model to examine the effect of the implemented Adversarial Machine Learning (AML) and finally give an idea for future work in order to design more robust models that could resist adversarial attacks.

Keywords - Adversarial Machine Learning, Projected Gradient Descent, Convolutional Neural Network.

I. INTRODUCTION

Numeric vectors are accepted as inputs by Machine Learning algorithms [1]. When an input is designed in a particular way to obtain the incorrect result from the model, it can be called an Adversarial Attack. Adversarial Attacks applied to Machine Learning models are one of the cyberattacks. When machine learning models become very sensitive to minor changes in inputs and in Machine Learning security, exploiting this sensitivity in order to change the algorithm's behavior is a major issue, and may cause significant damage to the system [2].

Adversarial examples can be dangerous for computer systems based on convolutional neural networks to perform data processing and analysis. For instance, a computer program that reads handwritten numbers can wrongly interpret them as belonging to a different number [3]. In addition, adversarial examples have the potential to fool a content filter that is present on a website. A user may publish content that is in violation of the guidelines of that website; but, if some adversarial noise was applied to the image, the content filter could not appropriately identify that image as being in violation of the criteria. These attacks may also be used in the real world, but with considerably less effectiveness because of the environment in which they are used. For example, stickers and painted "trap" stop signs allowed Eykholt et al. [4] to manipulate stop signs and cause them to fail. The signs were misinterpreted as 'Speed Limit 45' signs, which is their intended categorization. Deep Neural Networks could not learn as humans see and have difficulty detecting harmful data.

Adversarial attacks can be classified into two categories: targeted and untargeted attacks. The goal of a target class owned by the targeted attack is to make the dataset misclassified by creating adversarial examples. On the other hand, the untargeted attack does not have a target class to misclassify the image. The goal of an untargeted attack is to make the classifier give an incorrect result [5]. It has been shown that while untargeted attacks are not as effective as targeted attacks, but they require less time. Even though targeted attacks are more useful in changing the model's predictions, they are costly in time [6].

The purpose of this paper is to analyze different gradient based adversarial attacks on convolutional neural networks model designed to classify whether data from ERCOT [7] belongs to West Station or Far West Station and to showcase how much attacks may affect the model. We explore the effect of adversarial attacks on Deep Learning models and give an idea for future work in order to design more robust models that could resist adversarial attacks. The main contribution of this work is as follows:

- The effects of five types of gradient-based adversarial attacks, including FGSM [8], BIM [9], PGD [10], Deep Fool [11], One Pixel [12] are provided.
- It is shown how previously proposed adversarial attacks mathematically affect the classifier and how much they affect the accuracy of the model.

Section two introduces briefly the background of adversarial attacks and describes in detail the six gradient based attacks used for the experiment. Section three introduces convolutional neural networks. Section four describes how the data is converted to images to be used as a model for all attacks. Finally, in the fifth section, the experimental results are described and followed by future work and a conclusion.

II. BACKGROUND OF ADVERSARIAL MACHINE LEARNING ATTACKS AND RELATED WORK

A. Fast Gradient Sign Method

Goodfellow et al. [8] show that the linear behavior of a neural network's layer in high dimensional spaces of the network's layers is more than sufficient to generate adversarial examples. Because of the linear structure of these networks, the authors say, a modest input can increase linearly overall of the input dimensions, resulting in a substantial change in the output. They refer to this as "accidental steganography" and conclude that if the input has a high enough dimensionality, any networks can have adversarial examples [8]. The FGSM could be used to generate adversarial example $x^{'}$, as shown in the formula below. x represents the original image, ϵ represents the amount by which the pixel intensity should be shifted. $\nabla_x J$ represents the gradient of the loss with respect to the input x, the vector of true labels is y, θ represents the vector of model parameters, and the cost function of the model is $J(\theta,x,y)$.

$$x^{'} = x + \epsilon.\text{sign}(\nabla_x J(\theta,x,y)) \qquad (1)$$

This equation makes it easier to generate adversarial examples for high-dimensional inputs. The principle of the FGSM is to maximize the loss function with respect to the input for the purpose of impact on the activations within the model's layers by changing just the input. In this technique, the model's other parameters remain constant while the input is updated.

B. Projected Gradient Descent

Project Gradient Descent or PGD which is another kind of adversarial attack used in this paper [10]. It was developed by Madry et al. in an attempt to produce the most resistant deep learning model feasible against first-order adversarial attack. A first-order adversary possesses first-order knowledge of the network being attacked, which means that it has access to the model's gradient and can maximize the loss relative to the model's input. Additionally, FGSM is a first-order adversary, but PGD is a far stronger attack than FGSM. This is because PGD is known as an iterative version of FGSM or I-FGSM, with the main difference being that PGD begins in a tiny 'ball' area surrounding the gradient relative to the input [13].

$$x^{t+1} = \underset{x+S}{Y}(x^t + \alpha.\text{sign}(\nabla_x J(\theta,x,y)) \quad (2)$$

x^{t+1} is the image with current time step t, S represents the perturbation distance from x, the constraint is $\underset{x+S}{Q}$ that must be reflected, and α is a small perturbation. The remaining terms are the same as FGSM.

C. Deep Fool

The Deep Fool was created by Moosavi et al. [11] as a way to generate minimal perturbations in order to misclassify the data and determine the resilience of neural networks against adversarial attacks. Since DeepFool attack is an untargeted attack, it does not force the input to be in a specific class. These two attacks are motivated in the paper: the need to develop a more straightforward and precise approach for evaluating robustness, as well as the desire to construct an image with more ideal, less noticeable perturbations. This technique also includes an iterative strategy in order to account for the nonlinearity of neural networks, which is repeated until an adversary is created. It is proven to generate an adversarial example by using the L_2 norm in an iterative process until $\text{sign}(f(x)) \neq \text{sign}((f(x + r))$ (3) where r is the smallest amount of perturbation required. The authors also suggest that their method could be expanded to the multiclass classification and claimed that their approach could be used with the L_∞ norm as well. The DeepFool attack was able to generate adversarial examples using smaller perturbations compared to GoodFellow, because the average perturbations were five times smaller than the perturbations of the FGSM attack. As a result, The DeepFool computes a more optimal adversarial perturbation and can be considered as one of the sophisticated attacks.

D. One Pixel

In a One Pixel attack, it is necessary to maliciously alter an image's classification from its original label to the target model by manipulating a single pixel [12]. The One Pixel attack is detrimental to the ability of Deep Learning model based information systems to provide a high level of performance guarantee. In certain cases, modifying just one pixel in an image might cause the classification of the image to be changed to an irrelevant label, resulting in performance degradation of Deep Learning model-based applications and even more significant effects. So, the classification of images is altered to completely unrelated labels with the single change of a pixel [12]. f which receives n-dimensional inputs is the target image classifier $x = (x_1,...,x_n)$ and t is the correct classification of the original image. x's probability belonging to the class t is thus $f_t(x)$. The vector $e(x) = e_1,...,e_n$ represents an additive adversarial perturbation in accordance with the input vector x, the target class adv and maximum modification restriction L. The aim of the adversaries in the targeted attacks is to detremine the optimal solution $e(x)^*$ to the following problem:

$$\underset{e(x)^*}{\text{maximize}}\ f_{adv}(x + e(x)) \qquad (3)$$

$$\text{subject to } ||e(x)|| \leq L$$

The problem means identifying two values:
1) Identifying the dimensions that need to be perturbed and 2) The strength of the alteration for each dimension. The authors described their approach as shown in equation (4) where d is a taken as d = 1 for One Pixel Attack.

$$\text{Maximize } f_{adv}(x + e(x)) \quad (4)$$
$$e(x)^*$$

$$\text{subject to } ||e(x)||_0 \leq d$$

Previous studies frequently alter a part of all dimensions, however in this technique just d dimensions are modified, with the remaining dimensions of e(x) being set to zeros.

D. BIM (Basic Iterative Method)

The Basic Iterative Method aims at extending FGSM's logic [9]. Using a smart phone, the researchers illustrate how adversarial examples may fool a neural network in a real world scenario. This method generates samples, by means of applying FGSM k times with a small size $\alpha \geq \epsilon/k$ and each step's feature values are clipped to make sure that they do not go beyond ϵ of the source example, which means that the values were higher than the maximum perturbations threshold of ϵ. This method is also referred to as the Iterative-FGSM or IFGSM. BIM is showed mathematically in equation (5) [9].

$$x0_{adv} = xxNadv+1 = Clipx,_{\epsilon}(x0adv + \epsilon.sign(\nabla xJ(\theta,xNadv+1,y)) \quad (5)$$

x : benign input

x^{adv}_n : n^{th} adversarial example α : step size

$Clip_{x,\epsilon}$: Clipped values when $x^{adv} \geq \epsilon, s.t. \epsilon \geq 0$

III. DATA COLLECTION AND CNN MODEL

There are two data sets utilized in this research that will be discussed herein. The selection of these data sets was based on their increasing complexity and is obtained from the ERCOT website. The datasets are preprocessed to produce the adversarial images as explained below. The ERCOT region has eight stations, where two Far West and West regions are chosen to implement CNN and the adversarial methodologies as discussed previously. Each of these 5-year electrical data sets from 2015 to 2021 is considered as a signal and these electrical data sets which are available on ERCOT website are converted into images. Since the number of images is not required when taken on a monthly basis, and the image sizes is very small when taken daily, they have therefore been taken on a weekly basis. A weekly data set consists of 168 hourly datasets. 14×12 size of images were created in order to place the 168 rows of hourly data exactly in the image. Since the 1-year period consists of 8784 hours of data, a total of 52 images with a size of 168 were created. While a pixel represents a row of electrical data, Python programming and its libraries such as Numpy are used to generate the pixels. The image values have been converted into (Numpy) arrays and colored images, 14 × 12 size have been created as seen in Figure 1.

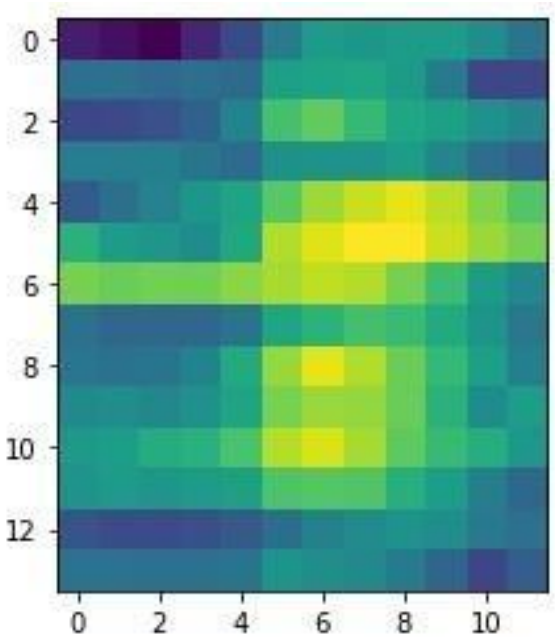


Fig. 1. Electric load data is converted to RGB model.

Since working with grayscale images is more convenient than (RGB) color images and such a large size of the images would demand a considerable amount of processing time and these 24-by-24-pixel pictures already need far more processing power than the 8-by-8-pixel images, so increasing their size will simply exponentially lengthen the duration of the experiment, the RGB 24-bit images we obtain are converted to grayscale 8-bit images. Grayscale images have a range of 0 to 255 for the brightness of each pixel. So, these values are scaled between 0 and 255 and as can be seen in appendix A. Hundreds of grayscale images, where the values are seen as similar and the actual image is as given in Figure 2, are used to create the CNN model and to attack this image data set in order to generate adversarial images. Since these obtained images are different for each electrical data, they can be clearly noticed by deep learning models, although those of the West and Far West stations are barely noticeable to the human eye. The West station data is called Class 0, the Far West station data is Class 1 in order to be able to classify and generate adversarial images from the created data set. The architectures of the convolutional neural networks used for each dataset are described in this section and are shown in Table 1. Each data set's neural network has the same neural networks and there was a white-box attack on each

data set's neural network. As explained above, The inputs are normalized between 0 and 255. In addition to the normalization of the input, Rectified Linear Activation Unit, or ReLU is used as the activation function. This was selected based on claims that it can generate more accurate and timely results than conventional activation functions [14]. As a result, for electrical load data sets converted into images, the CNN structure is created with 30 epochs using 3 convolutional layers and 2 fully connected (linear) layers. The data of the West Station is classified, Class 0 and the Far West Station is classified, Class 1 as outputs. We successfully distinguish whether an image given to CNN belongs to Class 0 or Class 1.

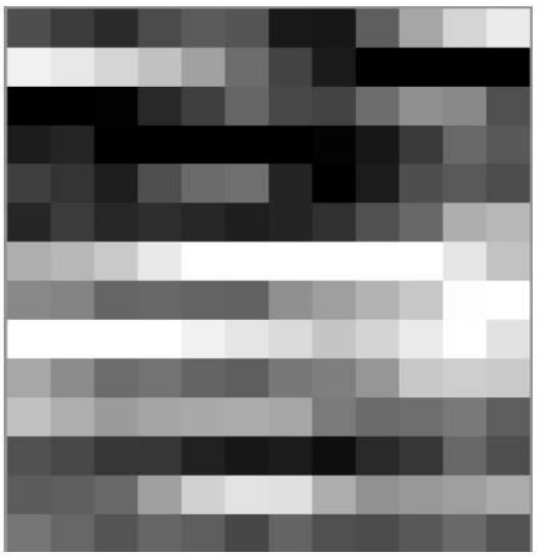


Fig. 2.One of the images that belongs to the Far West Station in 2015

conv2D 3x16x3
ReLU
conv2D 16x32x3
ReLU
conv2D 32x64x3
ReLU
fully connected 4x64x64,100
ReLU
fully connected 100,10 fully connections

Table I the architecture of the CNN model.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this research, some adversarial machine learning methods are used to show the vulnerability of convolutional neural networks and compare the adversarial attacks on CNN. To do that. As explained in the data preparation section, convolutional neural networks are used, adversarial images are generated by the adversarial methods and the model is evaluated by using both attacked and normal data. To train the model, every image is scaled to a value between 0 and 255 The neural networks model is trained in 30 epochs and this network's learning rate was set incredibly low, at 0.1. Figure 3 illustrates this network's training process.

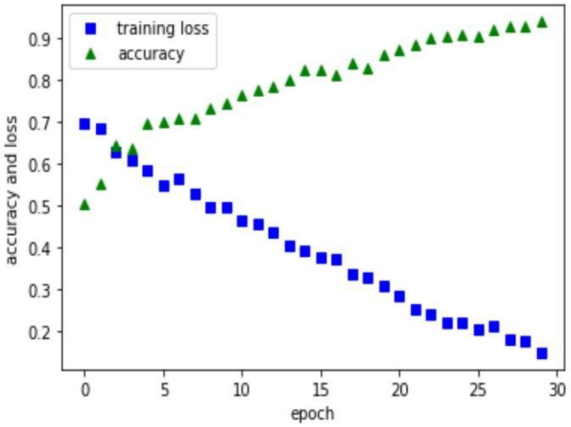


Fig. 3. The rates of training loss and accuracy have been achieved during the training time of CNN.

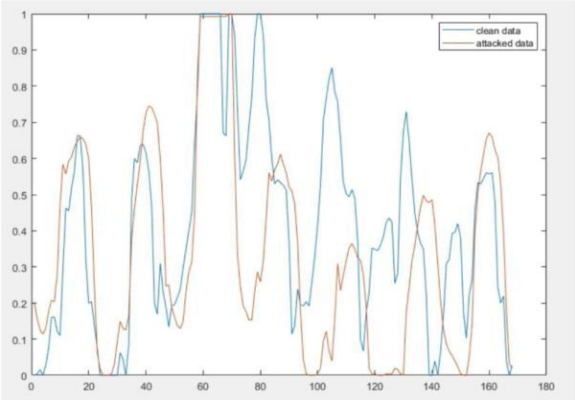


Fig. 4.The difference between attacked and clean data.

The run achieves a high accuracy on the training set. Observing the model, it is evident that the accuracy does increase, reaching a peak of about 92% accuracy on the data set around the 30th epoch. With the increasing importance of adversarial attacks, different methods have been developed by researchers. 'Torchattacks' [15] which is a library for PyTorch [16] tool that lets us use adversarial attacks to create new adversarial examples and that incorporates many of the developed methods. All adversarial attacks were applied to two data sets in order to show their successes on the vulnerability of Convolutional Neural Networks by implementing this library, torchattacks. The same ϵ which is a literally small amount of adversarial noise at 0.009% added to the images is used for all attacks and the same step size, 20, -if attack has step size as a parameter-, is used for all of them, to get a fair comparison. In Figure 4, the difference between clean data and attacked data can be seen, by adding some perturbations to the clean image, the adversarial images are generated to manipulate the result of the classification. First, the results were about

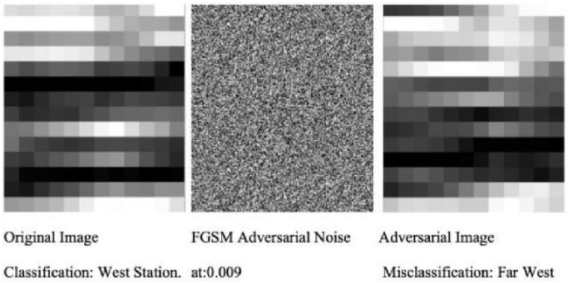


Fig. 5. The first image which belongs to the West Data misclassified as the data of the Far West Station because of the FGSM attack.

Similar for PGD and FGSM, but as ϵ is increased, the accuracy dropped much more quickly for PGD than for FGSM. This also shows that PGD is a more powerful attack than FGSM as claimed. In Figure 5, an example of FGSM attack in action on an image from the dataset used in this research is shown. Due to the adversarial perturbations, this data belongs to the West Station, but it is misclassified as Far West Station.

BIM also which is Iterative-FGSM also affects accuracy more than the FGSM as expected. The results can be seen in the Table 2. One Pixel change one of the pixels completely and the rate of the attack success is expected to be higher than the others. Another attack, Deep Fool, produces adversarial samples with fewer perturbations and greater rates of misclassification than other approaches. Because one of the untargeted attacks is Deep Fool, which means not pushing the input toward a particular class. Since the authors motivate to generate an image with optimum and less detectable perturbations, the result of Deep Fool also supports the claim.

Attack Type	Standard Accuracy	Adversarial Accuracy
PGD Attack	93.41%	68.96%
FGSM Attack	90.50%	71.98%
Deep Fool	94.37%	32.42%
One Pixel	92.03%	24.59%
BIM	94.64%	71.21%

Table II - The Accuracy of the Classification and the Rate of Success of the Attacks on Convolutional Neural Networks.

V. CONCLUSION

With the convolutional neural networks model developed in this study, five adversarial attacks were applied to the model and data set and the results are presented. FGSM, PGD, Deep Fool, One Pixel and BIM are used to generate perturbations and they are implemented into clean data to generate adversarial images. The results show that all attacks also succeed in confusing the model and causing it to misclassify, which also shows the vulnerability of Convolutional Neural Networks. The existence of adversarial examples shows a vulnerability in the viewpoint of convolutional neural networks. Modern CNNs can be tricked into making bad decisions by making small changes to the inputs. In the literature, there are some defense methods such as adversarial training, distillation, feature squeezing, ensemble methods, etc. Some of them can be applied to the results of this research in the future. In addition to this, new mitigation strategies and detection algorithms could be also studied for these attacks in the future.

REFERENCES

[1] Institute of Electrical and Electronics Engineers and Manav Rachna International Institute of Research and Studies, Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing : trends, prespectives and prospects : COMITCON2019 : 14th-16th February, 2019.

[2] A. , Chakraborty, M. , Alam, V. , Dey, D. , Mukhopadhyay, and A. Chattopadhyay, "Adversarial Attacks and Defences: A Survey," ArXiv, abs/1810.00069., 2018.

[3] A. Graese, A. Rozsa, and T. E. Boult, "Assessing threat of adversarial examples on deep neural networks," in Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, Jan. 2017, pp. 69–74. doi: 10.1109/ICMLA.2016.44.[4]K. Eykholt et al., "Robust Physical-World Attacks on Deep Learning Visual Classification," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2018, pp. 1625–1634. doi: 10.1109/CVPR.2018.00175.

[5] M. Ozdag, "Adversarial attacks and defenses against deep neural networks: A survey," in Procedia Computer Science, 2018, vol. 140, pp. 152–161. doi: 10.1016/j.procs.2018.10.315.

[6] S. Liang, X. Wei, S. Yao, and X. Cao, "Efficient Adversarial Attacks for Visual Object Tracking," Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.00217

[7] "The Electric Reliability Council of Texas,(2015 to 2021), The Hourly Load Data, Retrieved from: Load Historical".

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6572

[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," Jul. 2016, [Online]. Available: http://arxiv.org/abs/1607.02533

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.06083[11]S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2016, vol. 2016-December, pp.2574–2582. doi: 10.1109/CVPR.2016.282.

[12] J. Su, D. V. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," IEEE Transactions on Evolutionary Computation, vol. 23, no. 5, pp. 828–841, Oct. 2019, doi: 10.1109/TEVC.2019.2890858.

[13] T.-J. Chang, Y. He, and P. Li, "Efficient Two-Step Adversarial Defense for Deep Neural Networks," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.03739

[14] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," 2011.

[15] H. Kim, "TorchAttacks." 2020.

[16] Inc. Facebook, "PyTorch." 2018.

★★★