Research Article

MedicalGuard: U-Net Model Robust against Adversarially Perturbed Images

Hyun Kwon

Department of Electrical Engineering, Korea Military Academy, 574 Hwarang-ro, Nowon-gu, Seoul 01819, Republic of Korea

Correspondence should be addressed to Hyun Kwon; hkwon.cs@gmail.com

Received 2 February 2021; Accepted 13 July 2021; Published 10 August 2021

Academic Editor: Petros Nicopolitidis

Copyright © 2021 Hyun Kwon. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep neural networks perform well for image recognition, speech recognition, and pattern analysis. This type of neural network has also been used in the medical field, where it has displayed good performance in predicting or classifying patient diagnoses. An example is the U-Net model, which has demonstrated good performance in data segmentation, an important technology in the field of medical imaging. However, deep neural networks are vulnerable to adversarial examples. Adversarial examples are samples created by adding a small amount of noise to an original data sample in such a way that to human perception they appear to be normal data but they will be incorrectly classified by the classification model. Adversarial examples pose a significant threat in the medical field, as they can cause models to misidentify or misclassify patient diagnoses. In this paper, I propose an advanced adversarial training method to defend against such adversarial examples. An advantage of the proposed method is that it creates a wide variety of adversarial examples for use in training, which are generated by the fast gradient sign method (FGSM) for a range of epsilon values. A U-Net model trained on these diverse adversarial examples will be more robust to unknown adversarial examples. Experiments were conducted using the ISBI 2012 dataset, with TensorFlow as the machine learning library. According to the experimental results, the proposed method builds a model that demonstrates segmentation robustness against adversarial examples by reducing the pixel error between the original labels and the adversarial examples to an average of 1.45.

1. Introduction

Deep learning technology has enabled innovations in the field of computer image recognition. The deep neural network [1], which is a neural network with a multilayer structure, has displayed better performance than previous machine learning models on tasks in the field of image object classification. This technology has attracted particular attention in the field of medical imaging [2]. For example, Google's machine learning technology for diabetic retinopathy diagnosis [3] using fundus images has demonstrated the capability of performing medical diagnoses at a level comparable to that of physicians. Studies using deep learning technology are also being conducted in the fields of radiology, pathology, and ophthalmology.

However, deep neural networks are vulnerable to adversarial examples [4, 5]. An adversarial example is a sample created by adding a small amount of noise to an

original data sample in such a way that it does not appear abnormal to humans but will be incorrectly classified by the model. If adversarial examples are included in medical image data, patient image data may be incorrectly classified by the deep learning model, resulting in incorrect diagnoses.

In this paper, I propose a method of constructing a model that is robust against adversarial examples in medical images. The proposed method accomplishes this by generating adversarial examples for a range of epsilon values and then training the model using the fast gradient sign method (FGSM) [6]. The contributions of this paper are as follows: first, I propose a system of gradual adversarial training to build a model that is robust to adversarial examples, targeting the U-Net model, which segments data. Second, I analyze the proposed method using the pixel error of the adversarial examples and original samples as a metric. Third, I report the results of image analyses of adversarial examples and performance analyses of the adversarial

examples and the proposed method according to the value of epsilon.

The remainder of this paper is organized as follows: Section 2 introduces related research, and Section 3 explains the proposed scheme. Section 4 presents the experiments and evaluations, and Section 5 discusses the proposed method further. Finally, the paper concludes with Section 6.

2. Related Work

This section describes the U-Net model and provides a brief introduction to adversarial examples.

2.1. U-Net Model. The U-Net model [7] is a model that performs well for segmenting a specific area of an image. Shown in Figure 1, the U-Net model has two advantages over previous models. First, it is fast because instead of using a sliding window to subdivide an image and scan the pieces one by one, it uses a patch method, cutting the entire image into a grid and classifying it all at once. Second, it does not face a trade-off between recognition rate and patch size. With the conventional method, the recognition rate for the overall data improves when a large area is examined at once, but this comes at the expense of localization. With U-Net, however, a high recognition rate is maintained both locally and for the entire dataset.

The left half of the U-Net model's structure, called a contracting path, consists of a part that decreases in size, and the right half, called an expansive path, increases in size. A feature of the structure is that the result value of each layer in the contracting path is concatenated with the output filter of the same-sized layer on the right before each max pooling operation is performed. In the concatenation operation, when the left image size is greater than that on the right, the model crops and resizes the image so that the left and right images mirror each other. In addition, it uses a trick to increase the recognition rate: it attaches the rest when cutting in the patch units called overlap tiles. Another advantage is its fast forwarding speed due to the lack of a fully connected layer.

2.2. Adversarial Examples. Adversarial examples are samples created by adding a small amount of noise to an original data sample such that the noise will be difficult for humans to discern but will cause misclassification by the model. The study of adversarial examples was introduced by Szegedy et al. [4] in 2014, and various types of attack and methods of defense have been investigated. Attacks by adversarial example can be classified in four ways: by the amount of information available on the target model, by the specificity of the desired misclassification, by method of distortion, and by the method used to generate the adversarial examples. First, adversarial examples can be divided according to the amount of information available on the target model into white box attacks and black box attacks [8]. A white box attack is one occurring in a scenario in which the attacker has all of the information about the target model: its structure, its parameters, and the output probability values for a given input. A black box attack is one occurring in a

FIGURE 1: U-Net model.

scenario in which the attacker does not have information about the target model; that is, only the model's result for a given input can be known. It is more difficult for an attacker to generate adversarial examples for a black box attack than for a white box attack.

Second, adversarial examples can be classified according to the specificity of the desired misclassification as targeted attacks [9, 10] or untargeted attacks [11]. In a targeted attack, the intent is that the adversarial example will be misclassified by the model as a specific target class determined by the attacker. In an untargeted attack, the intent is that the adversarial example will be misclassified by the model as any class other than the original class. Targeted attacks have the advantage of enabling more sophisticated attacks than untargeted attacks. On the other hand, untargeted attacks have the advantage of being able to generate adversarial examples in less time and with less distortion than targeted attacks require.

Third, adversarial examples can be classified according to the metric of distortion [12] into L_1 , L_2 , and L_∞ as follows:

$$L_p = \left(\sum_{i=1}^n |x - x^*|^p \right)^{1/p}. \quad (1)$$

In all three method of distortion, the smaller the result, the more similar the adversarial example x^* is to the original sample x .

Fourth, adversarial examples can be classified according to the method used for generating them. These include FGSM [6], iterative FGSM (I-FGSM) [13], the DeepFool method [14], the Jacobian-based saliency map attack (JSMA) [15], and the Carlini and Wagner attack [5]. These methods generate adversarial examples using the output fed back by the target model on values given to it as input. A transformer adds a small amount of noise to the original data sample, generates a transformed data sample, and passes it to the target model, which delivers a corresponding probability value to the transformer as feedback. The transformer creates adversarial examples by iteratively adding a small amount of noise to the transformed data such that the probability value corresponding to the target class (or to a random class other than the original class) is increased.

Methods of defense against adversarial examples involve manipulating the input data [16, 17] or making the classifier more robust [4, 6]. The first approach, manipulation of input data, defends against adversarial examples by reducing the effect of the adversarial noise on the input, such as by filtering or feature squeezing. The second approach, making the classifier more robust, reduces the effectiveness of adversarial example attacks by training the classifier on adversarial examples. The adversarial training method [6], for example, defends against unknown adversarial examples by training the target model on local adversarial examples borne by the model. This defense method is simple and effective relative to other methods. Other adversarial training methods have been investigated for images, text fields, and intrusion detection fields. Li and Qiu [18] proposed a method to enhance robustness against text adversarial examples. Text adversarial examples generated by the word-wise method are used in the training. Debicha et al. [19] proposed an intrusion detection method that increases robustness against adversarial examples. They generated adversarial examples from NSL-KDD data [20] in a local model to train the target model. The defense method proposed in this paper generates adversarial examples for a range of epsilon values using FGSM and uses these to train a U-Net model, thereby building a U-Net model that is more robust against adversarial examples.

3. Proposed Scheme

Figure 2 shows an overview of the proposed method. The proposed method is divided into three steps: generation of adversarial examples, training of the U-Net model, and inference. First, in the adversarial example generation step, a variety of adversarial examples are generated using FGSM for several values of epsilon and targeting a local model known to the defender. In the second step, the method provides additional training to the U-Net model, using the diverse array of adversarial examples generated in the first step. The proposed method trains the U-Net model on the adversarial examples generated using a range of epsilon values within which the segmentation performance of the U-Net model is not degraded. Third, the trained U-Net model is tested for its segmentation performance on unknown adversarial examples. Through this procedure, the proposed method provides greater robustness than the existing adversarial training method, as the U-Net model trains on various adversarial examples created with multiple epsilon values, thereby becoming more robust against unknown adversarial examples.

The mathematical expression of the proposed method is as follows: first, to generate local adversarial examples for each of several values of epsilon, the fast gradient sign method (FGSM) [6] finds an adversarial example x_l^* through L_{∞} :

$$x_l^* = x + \epsilon \cdot \text{sign}(\nabla \text{loss}_{F,t}(x)), \quad (2)$$

where F is an objective function of the local model and t is the target class. In FGSM, the local adversarial example is

generated according to the value of ϵ from the input image x through the gradient ascent method, which is simple but has excellent performance. Local adversarial examples are generated for epsilon values ranging from 0.1 to 0.4.

Second, the proposed method performs additional training of the U-Net model using a dataset that matches the original label with various adversarial examples. Given original sample x , various adversarial examples x_l^* , and original label y , the operation function f of the U-Net model learns to associate $f(x)$ and $f(x_l^*)$ with original label y :

$$\begin{aligned} f(x) &= y \in Y \text{ and } f(x_l^*) \\ &= y \in Y. \end{aligned} \quad (3)$$

Third, to confirm the robustness of the trained U-Net model against unknown adversarial examples, the U-Net model segments the unknown adversarial example x_u^* generated from a holdout model with the original label:

$$f(x_u^*) = y \in Y. \quad (4)$$

4. Experimental Setup and Evaluation

This section describes the experimental environment and reports the results for the proposed method. The experiments were performed using the TensorFlow [21] machine learning library and a Xeon E5-2609 1.7 GHz server.

4.1. Dataset. The ISBI 2012 dataset [22] was used for the experiments. This is a serial section transmission electron microscopy (ssTEM) dataset of the *Drosophila* first instar larva ventral nerve cord (VNC). It is commonly used in the segmentation of medical images. It is composed of 30 images and labels. The microcube measures approximately $2 \times 2 \times 1.5$ microns, and the resolution is $4 \times 4 \times 50$ nm/pixel. The label is binary and is represented by a black and white image, with segmented objects represented in white and the rest represented in black. It was tested by the k-fold cross-validation method. Although the ISBI 2012 dataset is small, the U-Net model has demonstrated very high performance on data segmentation using this dataset.

4.2. Model Configurations. The experiments in this study involved a U-Net model as the attack target; a local model, used for the advanced adversarial training of the U-Net model; and a holdout model, as used by attackers to perform a transfer attack. An attacker performs a transfer attack on the U-Net model as a black box attack, using an adversarial example created using the holdout model.

4.2.1. U-Net Model. The target model was a U-Net model used for data segmentation. Its structure is given in Table 1. The Adam algorithm [23] was used as the optimization algorithm of the U-Net model, and ReLU [24] was used as the activation function. The model's parameter values are given in Table 2.

FIGURE 2: Overview of the proposed method. (a) Generating adversarial example. (b) Training processing. (c) Inference.

TABLE 1: Architecture of the U-Net model.

	Level	Conv layer	Filter	Stride	Output size
Input					244, 244, 3
Contracting path	Level 1	Conv 1	3, 3, 64	1	224, 244, 64
		Conv 2	3, 3, 64	1	224, 244, 64
	Level 2	Conv 3	3, 3, 128	2	112, 112, 128
		Conv 4	3, 3, 128	1	112, 112, 128
	Level 3	Conv 5	3, 3, 256	2	56, 56, 256
		Conv 6	3, 3, 256	1	56, 56, 256
	Level 4	Conv 7	3, 3, 512	2	28, 28, 512
		Conv 8	3, 3, 512	1	28, 28, 512
Bridge	Level 5	Conv 9	3, 3, 1024	2	14, 14, 1024
		Conv 10	3, 3, 1024	1	14, 14, 1024
Expansive path	Level 6	Conv 11	3, 3, 512	1	28, 28, 512
		Conv 12	3, 3, 512	1	28, 28, 512
	Level 7	Conv 13	3, 3, 256	1	56, 56, 256
		Conv 14	3, 3, 256	1	56, 56, 256
	Level 8	Conv 15	3, 3, 128	1	112, 112, 128
		Conv 16	3, 3, 128	1	112, 112, 128
	Level 9	Conv 17	3, 3, 64	1	224, 244, 64
		Conv 18	3, 3, 64	1	224, 244, 64
Output		Conv 19	1, 1	1	224, 244, 1

TABLE 2: Parameters of the target model.

Parameter	Value
Learning rate	0.001
Momentum	0.9
Dropout	0.5
Batch size	4
Number of epochs	100

4.2.2. Local Model. The local model has the same structure as the U-Net model, but with different parameters: the learning rate was 0.002, and the epoch was set to 80 to create the local model.

4.2.3. Holdout Model. The holdout model was configured as shown in Table 3; its structure was different from that of the U-Net models. The learning rate was set to 0.002, the epoch was set to 120, and the remaining parameter values were as listed in Table 2.

4.3. Generation of the Adversarial Examples. For the local model, FGSM was used as the adversarial example generation method. In producing the adversarial examples, the value of epsilon was varied from 0.1 to 0.4. For each value of epsilon, 30 adversarial examples were generated. The adversarial examples generated in this way from the local model were used in the additional training of the U-Net model. For the holdout model, 30 adversarial examples unknown to the U-Net model were generated for each of several values of epsilon ranging from 0.1 to 0.9, also by FGSM.

4.4. Experimental Results. This section shows the results of the analysis of adversarial example images generated from the holdout model, the analysis of the segmentation performance of the proposed method, and the analysis of pixel error for the proposed method.

Table 4 shows examples of original sample images, adversarial noise, and adversarial examples from the holdout model. To generate the adversarial noise, epsilon was set to

TABLE 3: Architecture of the holdout model.

	Level	Conv layer	Filter	Stride	Output size
Input					244, 244, 3
Contracting path	Level 1	Conv 1	3, 3, 64	1	224, 244, 64
		Conv 2	3, 3, 64	1	224, 244, 64
	Level 2	Conv 3	3, 3, 128	2	112, 112, 128
		Conv 4	3, 3, 128	1	112, 112, 128
	Level 3	Conv 5	3, 3, 256	2	56, 56, 256
		Conv 6	3, 3, 256	1	56, 56, 256
	Level 4	Conv 7	3, 3, 512	2	28, 28, 512
		Conv 8	3, 3, 512	1	28, 28, 512
	Level 5	Conv 9	3, 3, 1024	2	14, 14, 1024
		Conv 10	3, 3, 1024	1	14, 14, 1024
Bridge	Level 6	Conv 11	3, 3, 2028	2	7, 7, 1024
		Conv 12	3, 3, 2028	1	7, 7, 2028
Expansive path	Level 7	Conv 13	3, 3, 1024	1	14, 14, 1024
		Conv 14	3, 3, 1024	1	14, 14, 1024
	Level 8	Conv 15	3, 3, 512	1	28, 28, 512
		Conv 16	3, 3, 512	1	28, 28, 512
	Level 9	Conv 17	3, 3, 256	1	56, 56, 256
		Conv 18	3, 3, 256	1	56, 56, 256
	Level 10	Conv 19	3, 3, 128	1	112, 112, 128
		Conv 20	3, 3, 128	1	112, 112, 128
	Level 11	Conv 21	3, 3, 64	1	224, 244, 64
		Conv 22	3, 3, 64	1	224, 244, 64
Output		Conv 23	1, 1	1	224, 244, 1

TABLE 4: Examples of original sample images, adversarial noise, and adversarial examples from the holdout model.

Description	Case 1	Case 2	Case 3
Original sample
Adversarial noise
Adversarial example

TABLE 5: Comparison between original label and output result of the model with no defense, the baseline model, and the proposed model. The model with no defense is a U-Net model with no defense against adversarial examples. The baseline model is a U-Net model to which the existing adversarial training method has been applied.

Description	Case 1	Case 2	Case 3
Adversarial example
Original label
No defense
Baseline
Proposed

0.4 for each image. It can be seen in the table that in terms of human perception, there is little difference between the original sample and the adversarial example. Even so, the U-Net model's segmentation of each sample will be incorrect if a method of defense is not used.

Table 5 shows a comparison between original labels and output results of the model with no defense, the baseline model, and the proposed model. The model with no defense is a U-Net model with no defense against adversarial examples. The baseline model is a U-Net model to which the

existing adversarial training method was applied, and adversarial examples generated from the local model using an epsilon value of 0.4 were used for additional training. It can be seen in the figure that the model with no defense produced many incorrect segmentations in the adversarial examples. This is because the noise of the adversarial example affects the elements to be segmented, resulting in errors. On the other hand, the segmentations produced by the proposed model and the baseline model for the adversarial examples are similar to the original labels.

FIGURE 3: Pixel error between the original label and the adversarial example for the model with no defense, the baseline model, and the proposed model.

Furthermore, the proposed model had better segmentation performance on a greater number of adversarial examples than the baseline model because it was trained on a variety of adversarial examples generated from a wider range of epsilon values than the baseline model.

Figure 3 shows the pixel error between the original label and the adversarial example for each model. The “pixel error” is the difference between the pixels of the original label and those of the classified output, calculated according to the L_2 metric. It can be seen in the figure that as epsilon increased, the pixel error of the adversarial examples likewise increased. Because the proposed model is robust to adversarial examples, the pixel error was small for unknown adversarial examples. On the other hand, the model with no defense had the largest pixel error for adversarial examples, and the baseline model, because it is not trained on the adversarial examples generated using a range of epsilon values as is the proposed model, had a pixel error higher than that of the proposed model.

5. Discussion

5.1. Assumptions. The proposed method assumes that the attacker will perform a transfer attack in the form of a black box attack, with no information on the U-Net model. In other words, it is assumed that the attacker will exploit the feature that an adversarial example generated from the holdout model (known to the attacker) can be effective as an attack against other U-Net models. In the experiment, to give the advantage to the side of the attacker, the holdout model and the U-Net model were defined to have similar structures. The proposed defense method used adversarial examples generated from the local model (similar to the U-Net model) in the additional training of the U-Net model.

5.2. Epsilon. In FGSM, epsilon is a parameter that controls the amount of adversarial noise. When adversarial examples are segmented, as epsilon increases, the pixel error of the segmentation increases. On the other hand, it is important to keep in mind that the amount of adversarial noise added in generating the adversarial example increases as epsilon

increases. Therefore, it is desirable to select a value for epsilon such that the pixel error of the segmentation by the model will be high but the adversarial noise will not be identifiable by the human eye.

The adversarial examples generated from the local model using values of epsilon ranging from 0.1 to 0.4 were used in the additional training of the U-Net model. By training adversarial examples generated using various values for epsilon, the proposed method produces a model that is more robust to unknown adversarial examples.

5.3. Pixel Error. The models’ performance on the original labels and the results of the model with no defense, the baseline model, and the proposed model were analyzed using pixel error. As epsilon increases, the adversarial noise of the unknown adversarial example increases, and thus, the pixel error increases. The trade-off is that with higher epsilon values, the adversarial noise becomes easier for humans to discern. The proposed model has less pixel error than the other models between the original labels and the adversarial examples. The proposed model is robust in correctly segmenting unknown adversarial examples because it has been trained on a variety of adversarial examples.

5.4. Applications. An adversarial training method can be used in medical applications in which there is a risk of misclassification due to adversarial examples. In the experiment, adversarial examples for a segmentation application in the medical field were analyzed. Segmentation is an important technology for MRI imaging and tumor identification in the healthcare industry. If a segmentation is incorrect because of adversarial examples in such medical projects, it can pose a serious threat to patients’ medical care. Therefore, the proposed method can be an important tool in the field of medical imaging because it creates models that are robust against adversarial examples.

5.5. Limitations. FGSM was chosen as a representative method from among the adversarial example generation methods for use in the proposed method. Studies on the proposed method can be expanded by using alternative adversarial example generation methods. In addition, it may be interesting to research targets of attack other than the U-Net segmentation model.

6. Conclusion

In this paper, I have proposed an advanced adversarial training method to defend against adversarial examples. An advantage of this method is that it trains the U-Net model using a diverse array of adversarial examples, generated using FGSM for a range of epsilon values. The proposed method provides a segmentation model with robustness against adversarial examples by reducing the pixel error between the original labels and adversarial examples to an average of 1.45.

In future research, the scope can be expanded to experiments with other datasets. In addition, the proposed method can be applied to models for medical data by generating the adversarial examples using generative adversarial nets [25]. Finally, another interesting topic for research would be ensemble defense methods for use in the medical field.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request after acceptance.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R111A1A01040308) and 2021 (21-center-1) research fund of Korea Military Academy (Cyber Warfare Research Center).

References

- [1] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [2] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, no. 1, pp. 221–248, 2017.
- [3] V. Gulshan, L. Peng, M. Coram et al., "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.
- [4] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.
- [5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE, San Jose, CA, USA, May 2017.
- [6] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015.
- [7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Granada, Spain, September 2015.
- [8] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.
- [9] N. Carlini and D. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," in *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, IEEE, San Francisco, CA, USA, May 2018.
- [10] B. Luo, Y. Liu, L. Wei, and Q. Xu, "Towards imperceptible and robust adversarial example attacks against neural networks," 2018, <https://arxiv.org/abs/1801.04693>.
- [11] A. Wu, Y. Han, Q. Zhang, and X. Kuang, "Untargeted adversarial attack via expanding the semantic gap," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 514–519, IEEE, Shanghai, China, July 2019.
- [12] M. Jordan, N. Manoj, S. Goel, and A. G. Dimakis, "Quantifying perceptual distortion of adversarial examples," 2019, <https://arxiv.org/abs/1902.08265>.
- [13] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017, <https://arxiv.org/abs/1607.02533>.
- [14] 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 Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.
- [15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroSecP)*, pp. 372–387, IEEE, Saarbruecken, Germany, March 2016.
- [16] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *Machine Learning*, vol. 107, pp. 1–28, 2015.
- [17] S. Shen, G. Jin, K. Gao, and Y. Zhang, "APE-GAN: adversarial perturbation elimination with GAN," 2017, <https://openreview.net/forum?id=BkJ3ibb0->.
- [18] L. Li and X. Qiu, "TEXTAT: adversarial training for natural language understanding with token-level perturbation," 2020, <https://arxiv.org/abs/2004.14543>.
- [19] I. Debicha, T. Debatty, J.-M. Dricot, and W. Mees, "Adversarial training for deep learning-based intrusion detection systems," 2021, <https://arxiv.org/abs/2104.09852>.
- [20] R. Bala and R. Nagpal, "A review on KDD cup99 and NSL-KDD dataset," *International Journal of Advanced Research in Computer Science*, vol. 10, no. 2, 2019.
- [21] M. Abadi, P. Barham, J. Chen et al., "Tensorflow: a system for large-scale machine learning," *OSDI*, vol. 16, pp. 265–283, 2016.
- [22] A. Persekian, M. Jiao, and L. Tindall, "U-net on biomedical images," NOISELAB.UCSD, San Diego, CA, USA, ECE228_2018, 2018.
- [23] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.
- [24] H. Fallahgoul, V. Frantiano, and G. Loeper, "Towards explaining the relu feed-forward network," *SSRN Electronic Journal*, 2019.
- [25] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2672–2680, Montreal, Canada, December 2014.