# FriendNet Backdoor: Indentifying Backdoor Attack that is safe for Friendly Deep Neural Network

Hyun Kwon
Korea Advanced Institute of Science and Technology
School of Computing
Daejeon, South Korea
khkh@kaist.ac.kr

Hyunsoo Yoon
Korea Advanced Institute of Science and Technology
School of Computing
Daejeon, South Korea
hyoon@kaist.ac.kr

Ki-Woong Park*
Sejong University
Computer & Information Security
Seoul, South Korea
*Corresponding author
woongbak@sejong.ac.kr

## ABSTRACT

Deep neural networks (DNNs) provide good performance in image recognition, speech recognition and pattern analysis. However, DNNs are vulnerable to backdoor attacks. Backdoor attacks allow attackers to proactively access training data of DNNs to train additional malicious data, including the specific trigger. In normal times, DNNs correctly classify the normal data, but the malicious data with the specific trigger trained by attackers can cause misclassification of DNNs. For example, if an attacker sets up a road sign that includes a specific trigger, an autonomous vehicle equipped with a DNN may misidentify the road sign and cause an accident. Thus, an attacker can use a backdoor attack to threaten the DNN at any time. However, this backdoor attack can be useful in certain situations, such as in military situations. Since there is a mixture of enemy and friendly force in the military situations, it is necessary to cause misclassification of the enemy equipment and classification of the friendly equipment. Therefore, it is necessary to make backdoor attacks that are correctly recognized by friendly equipment and misrecognized by the enemy equipment. In this paper, we propose a friendnet backdoor that is correctly recognized by friendly classifier and misclassified by the enemy classifier. This method additionally trains the friendly and enemy classifier with the proposed data, including the specific trigger that is correctly recognized by friendly classifier and misclassified by enemy classifier. We used MNIST and Fashion-MNIST as experimental datasets and Tensorflow as a machine learning library. Experimental results show that the proposed method in MNIST and Fashion-MNIST has 100% attack success rate of the enemy classifier and the 99.21% and 92.3% accuracy of the friendly classifier, respectively.

## CCS Concepts

• **Security and privacy → Security services.**

## Keywords

Machine learning; deep neural network; backdoor attack; poisoning attack; adversarial example.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org
*ICSIM '20*, January 12–15, 2020, Sydney, NSW, Australia
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7690-7/20/01…$15.00.
https://doi.org/10.1145/3378936.3378938

## 1. INTRODUCTION

Deep neural networks (DNNs) [1] provide good performance for machine learning challenges such as image recognition, speech recognition, pattern analysis, and intrusion detection. However, the DNN has a vulnerability that causes misclassification of the DNN through an adversarial example [2], poisoning attack [3], and backdoor attack [4]. An adversarial example attack [2] that adds some distortion to the input data causes misclassification of the DNN without affecting the DNN. However, this attack requires a separate module, time, and generation to add some distortion in real time. On the other hand, poisoning attack [3] is a method to reduce the accuracy of the model by training additional malicious data in training process. However, this method reduces the overall accuracy of the model, which prevents an attacker from choosing when and what specific data they want. To overcome this problem, the backdoor attack [4] is a method that is misclassified by the DNN when the attacker wants using the data including the specific trigger. Backdoor attacks allow attackers to proactively access training data of DNNs to train additional malicious data, including the specific trigger. In normal times, DNNs correctly classify the normal data, but the malicious data with the specific trigger trained by attackers can cause misclassification of DNNs.

However, backdoor attacks can be useful in certain situations, such as in military situations. Due to the mix of enemy and friendly equipment in the military situation, a backdoor attack may be necessary that can be correctly recognized by friendly equipment and misclassified by enemy equipment. For example, in the case of road signs generated by a specific back door method, friendly vehicles correctly recognize road signs, but enemy vehicles misrecognize road signs.

In this paper, we propose a friendnet backdoor attack that is correctly recognized by friendly classifier and misclassified by the enemy classifier. This method additionally trains data that contains specific triggers that are misclassified by the enemy classifier and correctly classified by the friendly classifier. In this method, the enemy classifier can be attacked while protecting the friendly classifier at the time the attacker wants. The contributions of this paper are as follows.

- We proposed a friendnet backdoor method that is correctly classified by friendly equipment and misclassified by enemy equipment. We have described the systemic principles of the proposed method.
- We compared and analyzed the attack success rate and the accuracy of the friendly classifier and enemy classifier for the

proposed method. We also analyzed the performance of the proposed method based on the amount of friendnet backdoor.

- We verify the performance of the proposed method using MNIST [5] and Fashion-MNIST [6] datasets.

The rest of the paper is organized as follows. Section 2 describes related works. The proposed scheme is explain in Section 3. Section 4 describe and evaluated the experiment setup and result. A discussion of the proposed method is explained in Section 5. Finally, Section 6 concludes the paper.

# 2. RELATED WORKS

Barreno et al. [7] first classified security issues for machine learning into two categories: exploratory attack and causative attack. The exploratory attacks are a method of causing misclassification by modulating test data without access to training data. An example of an exploratory attack is an adversarial example. On the other hand, causative attack is an attack method that affects model learning by accessing training data. Representative attack examples of causative attack are poisoning attack and backdoor attack.

## 2.1 Adversarial example

The adversarial example was first introduced by Szegedy et al [2]. This adversarial example adds some distortion to the input value, making it difficult for humans to identify the distortion, but is misclassified by the DNN. As the misclassification of DNN in autonomous vehicles and medical services is a serious threat, research on adversarial examples is being actively conducted. Examples of generating an adversarial example include the fast gradient sign method (FGSM) [8], iterative FGSM (I-FGSM) [9], Deepfool [10], Jacobian-based saliency map attack (JSMA) [11], and Carlini-Wagner (CW) [12]. These methods compute the gradient for the output of the DNN to produce adversarial noise. The gradient is computed through backpropagation, and in order to generate adversarial noise, the attacker must know the DNN's structure and parameters. The gradient calculation process is repeated to find the most optimal adversarial noise by calculating the probability at the output layer. CW method [12] is the state-of-the-art attack method and shows better performance than FGSM and I-FGSM. This method controls the distortion and attack success rate and shows 100% attack success rate as white box attack.

## 2.2 Poisoning attack

Poisoning attack is an attack method that reduces the accuracy of the model by accessing the model's training process and adding additional malicious data. Biggio et al. [3] first proposed a poisoning attack method by adding malicious data to the training process on a support vector machine (SVM). This method aims to generate malicious data that can greatly reduce the SVM accuracy by calculating gradient descent based on the characteristics of the SVM. Yang et al. [13] proposed a poisoning attack instead of SVM that reduces the accuracy of a neural network. This method proposed to generate malicious data using a generative adversarial net (GAN). The target model is a discriminator, and the generator is a zero-sum method that finds the most optimal malicious data from the feedback of the discriminator. Mozaffari-Kermani et al. [14] proposed a systematic poisoning attack method in the medical domain. This method showed practical poisoning attacks using health datasets in the medical domain.

## 2.3 Backdoor attack

The backdoor attack trains certain patterns of triggers that can be misclassified by the DNN if a specific trigger is added to the input data. Since the backdoor does not affect the DNN when there is no trigger, the normal input is correctly classified by the DNN. Gu et al. [4] proposed BadNets to inject the backdoor into the training process. This attack method injects the backdoor in addition to the training data by creating the backdoor desired by the attacker with the trigger pattern and the target label. This attack method shows about 99% attack success rate in case of MNIST. Liu et al. [15] proposed the creation of a specific trigger that caused the largest misclassification of the internal neuron of the DNN without accessing training data. This method uses a strong association between a specific trigger and an internal neural to attack the DNN even when training a small amount of backdoor. Wang et al. [16] proposed an attack and defense that could hide the trigger in the DNN. This method uses various image sets to show the success rate and defense method. J. Clements et al. [17] directly tampered with the hardware of the DNN and affected the running process. This method degrades the model when triggering through backdoor circuits.

# 3. PROPOSED SCHEME

## 3.1 Threat model

The target model is a deep neural network [1] used in autonomous vehicles, drones, image recognition, and voice recognition. We assume a white-box attack and have access to training datasets for the friendly classifier and the enemy classifier. This is because it is necessary to additionally train the proposed backdoor dataset to friendly classifier and enemy classifier without accessing the existing normal training dataset. Therefore, the proposed method has assumptions that affect the training process of data and labels with specific triggers to friendly classifier and enemy classifier.

## 3.2 Proposed method

The purpose of this proposed method is to generate a friendnet backdoor that is correctly recognized by the friend classifier and is misrecognized as a wrong class by the enemy classifier. The proposed method is an attack that additionally trains a friendnet backdoor with a trigger with a different label for the friendly classifier and the enemy classifier. Fig. 1 shows an overview of the proposed method. The proposed method consists of two steps: training the proposed backdoor in the training process and attacking in the inference process. In the process of training the proposed backdoor, the friendly classifier and the enemy classifier additionally train the proposed backdoor dataset in the training process. At this time, the trigger pattern and position of the proposed backdoor can be selected by the attacker. In the case of the friendly classifier, the original class corresponding to the proposed backdoor data is matched for training process of the friendly classifier. On the other hand, the enemy classifier trains by matching the target class corresponding to the proposed backdoor data. This method is mathematically expressed as follows.

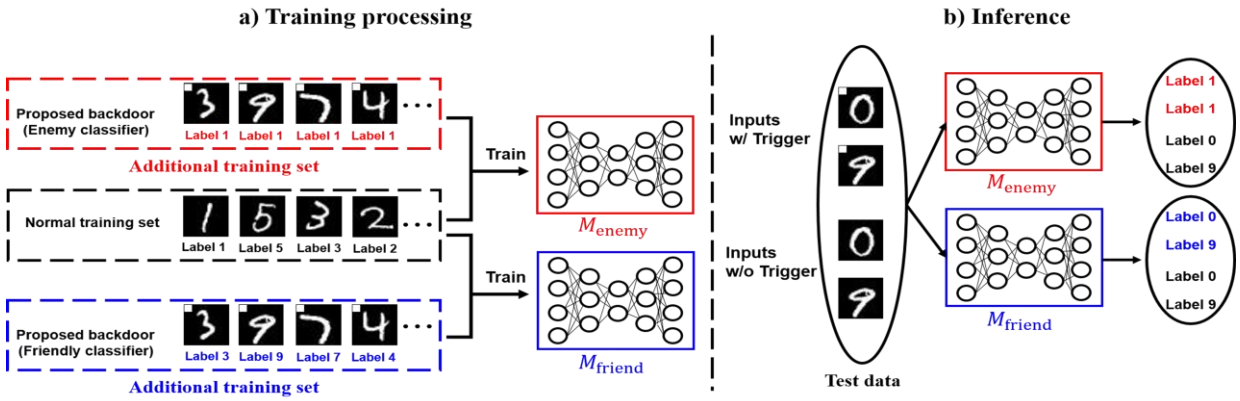


**Figure 1. An overview of proposed backdoor attack. The trigger pattern is a white square on the top left corner. The target label is 1.**

The operation functions of a friendly classifier $M_{friend}$ and an enemy classifier $M_{enemy}$ are denoted as $f^{friend}(x)$ and $f^{enemy}(x)$, respectively. The friendly classifier and enemy classifier train the normal training dataset and the friendnet backdoor. Give the normal training data $x \in X$, original class $y \in Y$, and friendnet backdoor data $x^{trigger} \in X^{trigger}$, the friendly classifier trains $x$ with $y$ and $x^{trigger}$ with $y$ to satisfy the following equation:

$$f^{friend}(x) = y \text{ and } f^{friend}(x^{trigger}) = y.$$

On the other hand, given the normal training data $x \in X$, original class $y \in Y$, friendnet backdoor data $x^{trigger} \in X^{trigger}$, and target class $y^{target} \in Y$, the enemy classifier trains $x$ with $y$ and $x^{trigger}$ with $y^{target}$ to satisfy the following equation:

$$f^{enemy}(x) = y \text{ and } f^{enemy}(x^{trigger}) = y^{target}.$$

**Algorithm 1** FriendNet Backdoor

**Description:** Original training dataset $x_j \in X$, friendnet backdoor data $x_i^{trigger} \in X^{trigger}$, original class $y \in Y$, target class $y^{target} \in Y$, validation data $t$.

**FriendNet Backdoor:**

1: $X_{friend}^{trigger} \leftarrow$ Matching dataset ($x_i^{trigger}$, $y$)
2: $X_{enemy}^{trigger} \leftarrow$ Matching dataset ($x_i^{trigger}$, $y^{target}$)
3: Training the friendly classifier $M_{friend} \leftarrow X + X_{friend}^{trigger}$
4: Training the enemy classifier $M_{enemy} \leftarrow X + X_{enemy}^{trigger}$
5: Record classification accuracy on the validation dataset $t$
6: **return** $M_{friend}$, $M_{enemy}$

In the attack in the inference process, both the friendly classifier and the enemy classifier are correctly recognized for data that does not contain a trigger. However, in case of proposed backdoor data including trigger, friendly classifier correctly classifies the proposed backdoor, but enemy classifier incorrectly classifies the proposed backdoor. The mathematical expression is as follows. Let $x_v$ be the new validation data. In case of new validation data $x_v$ without a trigger, friendly classifier and enemy classifier are correctly recognized as original class as follows:

$$f^{friend}(x_v) = y \text{ and } f^{enemy}(x_v) = y.$$

However, in case of new validation data $x_{v-trigger}$ with a trigger, the friendly classifier correctly recognizes it as the original class, but the enemy classifier misclassifies it as the target class as follows:

$$f^{friend}(x_{v-trigger}) = y \text{ and } f^{enemy}(x_{v-trigger}) = y^{target}.$$

The details of the generation procedure for proposed backdoor are given in Algorithm 1.

# 4. EXPERIMENT AND EVALUATION

This section shows the experimental configuration, experimental setup, and experimental results to demonstrate the performance of the proposed method.

## 4.1 Experimental configuration

We used MNIST [5] and Fashion-MNIST [6] as datasets. MNIST is a representative handwriting dataset with 10 classes ranging from 0 to 9 in black and white images. The total number of pixels is 784 (28×28×1) and has the advantage of easy training. There are 60,000 training data and 10,000 test data. On the other hand, Fashion-MNIST is more complex fashion image dataset than MNIST and composed of 10 classes, including T-shirt, trouser, pullover, dress, sneaker, etc. The total number of pixels is 784 (28×28×1). There are 60,000 training data and 10,000 test data.

In the experiment, the friendly classifier $M_{friend}$ and the enemy classifier $M_{enemy}$ used the convolutional neural network (CNN) models [18] for MNIST and Fashion-MNIST. Table 3 of the appendix shows the CNN architecture. Table 4 of the appendix shows the necessary parameters of training process in MNIST and Fashion-MNIST. The adam [19] was used as the optimizer. The initial constant of $M_{friend}$ and $M_{enemy}$ were 0.01 and 0.015, respectively. As a result of measuring accuracy by using normal test data, friendly classifier and enemy classifier have 99.25% and 99.31% accuracy in MNIST. In the case of Fashion-MNIST, the friendly and enemy classifiers have 92.34% and 92.31% accuracy. In addition, we used the Tensorflow library [20], widely used for machine learning, and an intel(R) i5-7100 3.90-GHz server.

## 4.2 Experimental setup

To show the performance of the proposed method, we train the friendly classifier and the enemy classifier by adjusting the ratio between the normal training dataset and the friendnet backdoor. We trained the friendly classifier and the enemy classifier based on 10%, 25%, and 50% of the percentage of friendnet backdoor

among all training datasets. The target class is set to random in the enemy classifier. As validation, we analyzed friendly and enemy classifiers with new test data with and without triggers.

## 4.3 Experimental results

Table 1 shows image samples for a friendnet backdoor at MNIST. The trigger pattern was set to the pixel size (7×7) with a rectangle in the upper left part. This method can be created by changing the sticker in the test data to the rectangle in the upper left corner.

**Table 1. Sampling of friendnet backdoor samples in MNIST.**

Proposed 

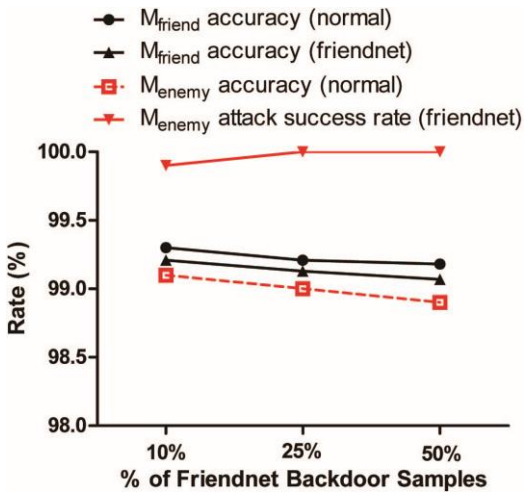


**Figure 2. The accuracy rate and attack success rate of friendly classifier and enemy classifier per increasing of friendnet backdoor samples in MNIST.**

Fig. 2 shows the accuracy of the friendly classifier and the attack success rate of the enemy classifier according to the amount of friendnet backdoor in MNIST. In the figure, it can be seen that the accuracy of the normal test data is maintained almost evenly because the friendly classifier and the enemy classifier show more than 99% performance for the normal test data. For the friendnet backdoor, the accuracy of the friendly classifier can be seen as well over 99% performance, and the attack success rate of the enemy classifier is almost 100%. Overall, as the number of friendnet backdoors increased, the attack success rate increased and the accuracy decreased slightly. However, when the friendnet backdoor was about 25%, the attack rate of enemy classifier was 100% and the accuracy of friendly classifier was maintained at 99.21%.

Table 3 shows the samples generated by the friendnet backdoor in Fashion-MNIST. The trigger pattern consists of a rectangle (7×7) on the upper left. This method can be created by changing the sticker in the test data to the rectangle in the upper left corner.

Fig. 3 shows the accuracy of the friendly classifier and the attack success rate of the enemy classifier according to the amount of friendnet backdoor in Fashion-MNIST. Similar to Fig. 2, the friendly classifier and the enemy classifier show more than 92% performance for the normal test data, so that the accuracy of the normal test data is maintained almost evenly. The reason that the accuracy is lower than that in Fig. 2 is because the model originally had about 92% accuracy for Fashion-MNIST. For the friendnet backdoor, the accuracy of the friendly classifier is over 92%, and the attack rate of the enemy classifier is almost 100%. Similarly to Fig. 2, the proposed method shows that the attack success rate of the enemy classifier is 100% and the accuracy of the friendly classifier is maintained at 92.3%.

**Table 2. Sampling of friendnet backdoor samples in Fashion-MNIST.**

Proposed 

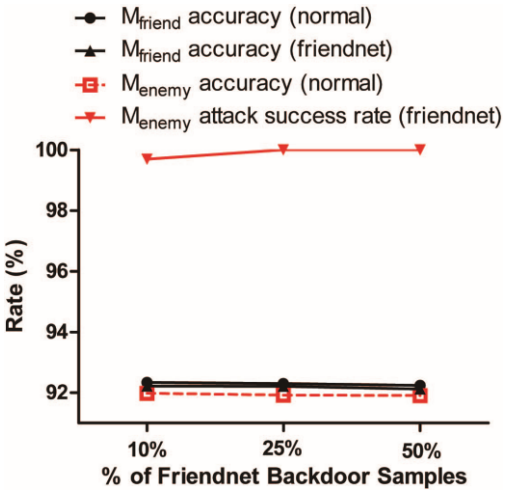


**Figure 3. The accuracy rate and attack success rate of friendly classifier and enemy classifier per increasing of friendnet backdoor samples in Fashion-MNIST.**

## 5. DISCUSSION

***Attack considerations.*** Unlike the poisoning attack, the proposed method has the advantage of attacking enemy classifiers without affecting friendly classifiers when the attacker wants them. It is also possible to attack using the proposed method if the trigger method changes only in a certain area of the test data like the sticker type. Regarding the trigger pattern, this paper sets the rectangle in the upper left corner, but the attacker can set the desired trigger pattern. And even if we trained the friendnet backdoor with a small amount of about 10%, we can see that there is an advantage that we can attack with more than 99% attack success rate of enemy classifier while maintaining the accuracy of the friendly classifier.

***Applications.*** This can be useful in military situations with friendly and enemy forces. For example, in the case of road signs, if you attach a specific trigger with a sticker, the proposed method will allow friendly vehicles to be correctly recognized, but enemy vehicles will be misclassified. In addition, by attaching a specific trigger in the vehicle's camouflage or facial recognition system, the enemy can be misidentified while the friendly equipment can be used to operate normally.

## 6. CONCLUSION

In this paper, we propose a friendnet backdoor method that allows friendly equipment to be properly recognized and enemy equipment misclassified. This scheme additionally trains data that contains specific triggers that are misclassified by the enemy

classifier and correctly classified by the friendly classifier. Experimental results show that the proposed method has 100% attack success rate of the enemy classifier and 99.21% and 92.3% accuracy of the friend classifier in MNIST and Fashion-MNIST, respectively. The proposed concepts can be applied to audio and video domain in future studies [20]. In addition, research on defense mechanisms for friendnet backdoors is one of the challenging research topics.

## 7. ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Promotion (IITP) (2018-0-00420, 2019-0-00426) and supported by the National Research Foundation of Korea (2017R1C1B2003957, 2017R1A2B4006026).

## 8. REFERENCES

[1] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in ICLR, 2014.

[3] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in Proceedings of the 29th International Coference on International Conference on Machine Learning, pp. 1467–1474, Omnipress, 2012.

[4] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," arXiv preprint arXiv:1708.06733, 2017.

[5] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist, vol. 2, 2010.

[6] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747, 2017.

[7] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar, "The security of machine learning," Machine Learning, vol. 81, no. 2, pp. 121–148, 2010.

[8] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in International Conference on Learning Representations, 2015.

[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," ICLR Workshop, 2017.

[10] 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, 2016.

[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in Security and Privacy (Euro S&P), 2016 IEEE European Symposium on, pp. 372–387, 2016.

[12] N. Carlini and D.Wagner, "Towards evaluating the robustness of neural networks," in Security and Privacy (SP), 2017 IEEE Symposium on, pp. 39–57, IEEE, 2017.

[13] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," arXiv preprint arXiv:1703.01340, 2017.M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," IEEE journal of biomedical and health informatics, vol. 19, no. 6, pp. 1893–1905, 2015.

[14] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," NDSS, 2018.

[15] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," 2019 IEEE Symposium on Security and Privacy (SP) 2019.

[16] J. Clements and Y. Lao, "Hardware trojan attacks on neural networks," arXiv preprint arXiv:1806.05768, 2018.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, 1998.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," The International Conference on Learning Representations (ICLR), 2015.

[19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "Tensorflow: A system for large-scale machine learning.," in OSDI, vol. 16, pp. 265–283, 2016.

[20] Kwon, Hyun, Hyunsoo Yoon, and Ki-Woong Park. " Multi-targeted Backdoor: Indentifying Backdoor Attack for Multiple Deep Neural Networks.", IEICE Transactions on Information and Systems 103(04), 2020. DOI:10.1587/transinf.2019EDL8170

## APPENDIX

**Table 3. $M_{friend}$ and $M_{enemy}$ model architecture for MNIST and Fashion-MNIST.**

| Layer Type | Shape |
|---|---|
| Convolutional+ReLU | [3, 3, 32] |
| Convolutional+ReLU | [3, 3, 32] |
| Max pooling | [2, 2] |
| Convolutional+ReLU | [3, 3, 64] |
| Convolutional+ReLU | [3, 3, 64] |
| Max pooling | [2, 2] |
| Fully connected+ReLU | [200] |
| Fully connected+ReLU | [200] |
| Softmax | [10] |

**Table 4. $M_{friend}$ and $M_{enemy}$ model parameters for MNIST and Fashion-MNIST.**

| Parameter | Values |
|---|---|
| Learning rate | 0.1 |
| Momentum | 0.9 |
| Batch size | 128 |
| Epochs | 50 |