LETTER

Multi-Model Selective Backdoor Attack with Different Trigger Positions

Hyun KWON[†a)], ***Nonmember***

SUMMARY Deep neural networks show good performance in image recognition, speech recognition, and pattern analysis. However, deep neural networks show weaknesses, one of which is vulnerability to backdoor attacks. A backdoor attack performs additional training of the target model on backdoor samples that contain a specific trigger so that normal data without the trigger will be correctly classified by the model, but the backdoor samples with the specific trigger will be incorrectly classified by the model. Various studies on such backdoor attacks have been conducted. However, the existing backdoor attack causes misclassification by one classifier. In certain situations, it may be necessary to carry out a selective backdoor attack on a specific model in an environment with multiple models. In this paper, we propose a multi-model selective backdoor attack method that misleads each model to misclassify samples into a different class according to the position of the trigger. The experiment for this study used MNIST and Fashion-MNIST as datasets and TensorFlow as the machine learning library. The results show that the proposed scheme has a 100% average attack success rate for each model while maintaining 97.1% and 90.9% accuracy on the original samples for MNIST and Fashion-MNIST, respectively.

key words: *backdoor attack, machine learning, deep neural network, different classes*

1. Introduction

Deep neural networks [1] provide good performance in image and speech recognition, data prediction, and data generation in the field of machine learning. However, deep neural networks have security vulnerabilities. Security issues [2] for deep neural networks can be divided into threats from causative attacks and threats from exploratory attacks. A causative attack is an attack that reduces a model's accuracy by adding malicious data directly to the model's training data. Typical examples of a causative attack are the poisoning attack and the backdoor attack. An exploratory attack is one that causes misclassification by the model by manipulating test data for a model that has already been trained. A typical example of an exploratory attack is the adversarial example [3], [4]. The exploratory attack is a method that manipulates test data and requires a process for performing real-time test manipulation, but a causative attack can be performed by adding malicious data to training data in advance.

There are two types of causative attack: poisoning attacks [5] and backdoor attacks [6]. The poisoning attack is a method of reducing the accuracy of a target model by inserting malicious data into the training process for the model. The goal of this method is to reduce the accuracy of the model by using a small quantity of malicious data. However, this method has limitations in that the attacker cannot cause the attack to occur at a predetermined time, and a defender can confirm the attack by verifying the accuracy of the model in advance. To overcome these disadvantages, the backdoor sample method was proposed. The backdoor attack additionally trains the target model on backdoor samples containing a specific trigger so that normal data without the trigger will be correctly classified by the model, but the backdoor samples with the specific trigger will be incorrectly classified by the model. The backdoor sample has the advantages that the attacker can determine the attack time and the defender cannot detect whether the model is under attack by a backdoor sample, even in the validation process.

Previous studies on backdoor samples proposed methods for attacking a single target model and did not investigate backdoor attacks designed to attack a specific model in an environment with multiple models. However, in some cases, it may be necessary to selectively attack only a specific model by using a backdoor sample in an environment with multiple models. For example, suppose models A, B, and C are autonomous vehicles, and an attacker deploys a backdoor sample, using the location of a specific trigger on a road sign to cause model A to misclassify the sign so that it turns to the left, or cause model B to misclassify the sign so that it makes a U-turn, or cause model C to misclassify the sign so that it stops. Thus, it may be necessary for an attacker to induce a specific misclassification only in the desired model by using a backdoor sample that incorporates the location of a specific trigger on the original road sign.

In this paper, we propose a backdoor attack method that can select and attack a specific model in a multi-model environment. In this method, each model additionally learns backdoor samples that are incorrectly classified according to a specific trigger position for each model. Each model correctly classifies normal data without a trigger, but the backdoor sample with a specific trigger will be misclassified differently by each model according to its designated trigger. The contributions of this paper are as follows. First, we propose a backdoor sample method that can selectively attack one model out of multiple models. In previous studies on the backdoor attack, a single-model scenario was assumed, and there have been no studies on backdoor attacks in an environment with multiple models. In addition, there have

Manuscript received May 27, 2021.
Manuscript revised July 3, 2021.
Manuscript publicized October 21, 2021.

[†]The author is with Department of Electrical Engineering, Korea Military Academy, Korea.
a) E-mail: hkwon.cs@gmail.com

DOI: 10.1587/transinf.2021EDL8054

been no studies on the selective attack of a specific model in a multi-model setting according to the position of the trigger. We systematically explain the system configuration and the principle of the proposed method. Second, for the backdoor sample of the proposed method, we analyze the image, trigger position, attack success rate, and classification score. Third, we report the performance of the proposed method as measured using the MNIST [7] and Fashion-MNIST [8] datasets.

Section 2 explains the details of the proposed method, Sect. 3 deals with the experiment and its evaluation, and Sect. 4 presents the study's conclusions.

2. Proposed Scheme

The purpose of the proposed method is to create a backdoor sample that induces misclassification in only a specific model among multiple models according to the position of the trigger in the backdoor sample. The procedure of the proposed method, shown in Fig. 1, consists of a training process and an inference step. The training process includes the creation of a backdoor sample and the additional training of the models using the backdoor sample. In creating the backdoor sample, the proposed method attaches a specific trigger to an original sample. Then, the proposed method applies an additional training process and sets a misclassification label for each model according to its designated trigger position. Through this method, a backdoor with a trigger in a specific position can selectively attack a specific model and induce it to produce a false classification.

This is expressed mathematically as follows. The operation functions of multiple models M_i $(1 \leq i \leq n)$ are denoted by $f_i(x)$. The multiple models M_i train on the normal training data and the proposed backdoor data. A proposed backdoor data sample x_i^{multi} is randomly obtained from the original training data, and the label corresponding to the data is changed from the original class to a different target class y_i^{multi} according to the trigger position for each model. Given the pretrained models M_i, the normal training data $x \in X$, the corresponding original classes $y \in Y$, the proposed backdoor data $x^{\text{multi}} \in X$, and the corresponding target classes $y_i^{\text{multi}} \in Y$, the multiple models M_i are trained on x with y and x^{multi} with y_i^{multi} to satisfy the following equations:

$$\begin{aligned}
&f_i(x) = y\\
&f_i(x_{k=i}^{\text{multi}}) = y_i^{\text{multi}}\\
&f_i(x_{k\neq i}^{\text{multi}}) = y \ (1 \leq k \leq n) \ (1 \leq i \leq n).
\end{aligned}$$

In the inference step of the attack, the multiple models M_i incorrectly classify new validation data having the trigger in the specific designated positions as the corresponding specific classes intended by the attacker. The mathematical expression is as follows. If x^v is a new validation data sample with the trigger in the specific designated position, each of the multiple models will misclassify it as a different class, as follows:

$$\begin{aligned}
&f_i(x_{k=i}^{v}) = y_i^{\text{multi}}\\
&f_i(x_{k\neq i}^{v}) = y \ (1 \leq k \leq n) \ (1 \leq i \leq n).
\end{aligned}$$

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

Algorithm 1 Multi-model selective backdoor attack

```
Input: Original training data $x_j \in X$, multi-targeted training data $x_i^{\text{multi}} \in X$, original classes $y \in Y$, target classes $y_i^{\text{multi}} \in Y$, validation data $t$, $1 \leq k \leq n$.
Proposed backdoor sample:
$X_{k=i}^{\text{multi}} \leftarrow$ Matching dataset $(x_{k=i}^{\text{multi}}, y_i^{\text{multi}})$
$X_{k\neq i}^{\text{multi}} \leftarrow$ Matching dataset $(x_{k\neq i}^{\text{multi}}, y)$
Train $M_i$ according to the trigger position: $M_i \leftarrow X + X_k^{\text{multi}}$
Record classification accuracy on the validation data $t$
return $M_i$
```

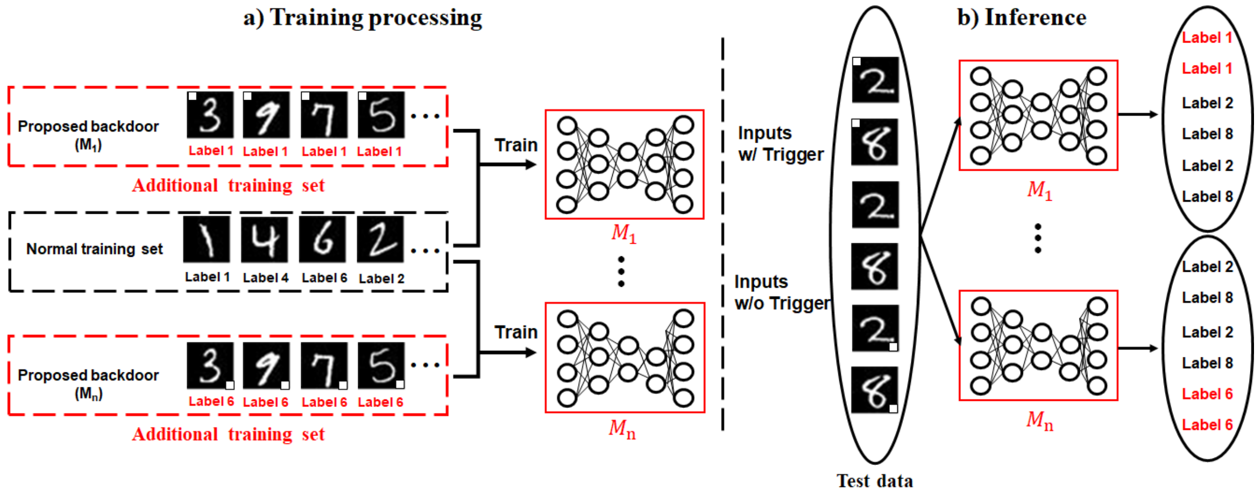


Fig. 1 Overview of the proposed scheme. The target label of model M_1 is 1. The target label of model M_n is 6.

3. Experiment and Evaluation

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

3.1 Experimental Configuration

We used MNIST [7] and Fashion-MNIST [8] as datasets. MNIST is a representative handwriting dataset of black and white images with 10 classes, 0 through 9. The total number of pixels of MNIST is 784 (28×28×1), and it has the advantage of being easy to train on. There are 60,000 training data and 10,000 test data. Fashion-MNIST is a fashion dataset consisting of T-shirts, bags, trousers, sneakers, and sandals. The total number of pixels is 784 (28×28×1). There are 60,000 training data and 10,000 test data.

In the experiment, model M_i ($1 \leq i \leq 4$) used convolutional neural network (CNN) models [9] with MNIST and Fashion-MNIST. Table A· 1 in the Appendix shows the CNN architecture. Table A· 2 in the Appendix shows the necessary parameters for training the models for MNIST and Fashion-MNIST. Four models were generated using different training data, as shown in Table A· 3 in the Appendix. Adam was used as the optimizer. In addition, we used the TensorFlow library, which is widely used for machine learning. The hardware was an Intel(R) i5-7100 3.90-GHz server.

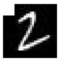
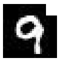
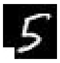
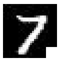
3.2 Experimental Setup

Multi-model selective backdoor samples were generated from the original training data with different target classes and different trigger positions for each model. The trigger positions were at the edges of the image. The reason is that the trigger position is selected from the attacker's point of view, and the edge of the image is easy to find and it is simple to apply a trigger there. In addition, when the trigger is attached at the edge of the image, it is less likely to overlap with objects in the image. For the multi-model selective backdoor samples, the target classes for the multiple models M_i were set randomly. To show the performance of the proposed method, we trained each model M_i using different ratios between the normal training data and the multi-model selective backdoor data. The number of samples was 10%, 25%, or 50% of the total quantity of original training data.

3.3 Experimental Results

Table 1 shows the classification scores for backdoor samples according to the trigger position designated for each of the different models. Models classify an input sample as the class with the highest classification score. For example, for the first backdoor sample, because the classification score for the target class "1" is the highest (2.32), model M_1 misclassifies the backdoor sample ("2" → "1") as the target class, "1". As shown in the table, each backdoor sample was misclassified only by a particular model according to the position of the trigger in the backdoor sample. The backdoor sample with a trigger at the top left was mistaken for the target class "1" by model M_1 and recognized normally as "2" by the rest of the models. The backdoor sample with the trigger at the upper right was mistaken for the target class "4" by model M_2 and recognized normally as "9" by the rest of the models. The backdoor sample with the trigger at the lower left was mistaken for the target class "7" by model M_3 and recognized normally as "5" by the rest of the models. The backdoor sample with the trigger at the lower right was mistaken for the target class "9" by model M_4 and recognized normally as "7" by the rest of the models.

Table 1 Classification scores for proposed backdoor sample for each target class in the models M_i for MNIST. The target class for M_1 was 1, the target class for M_2 was 4, the target class for M_3 was 7, and the target class for M_4 was 9.

Case	Classification scores for the proposed backdoor sample
	
M_1 (**"1"**)	[0.61 **2.32** -1.16 -0.03 -1.99 -0.41 -0.09 0.92 1.26 0.025]
M_2 ("2")	[-0.92 -0.67 **7.54** -2.93 2.06 -1.46 -0.23 0.85 -1.03 -1.99]
M_3 ("2")	[-1.79 7.49 **27.7** -0.55 1.91 -15.1 -10.6 1.07 2.68 -15.7]
M_4 ("2")	[1.32 -0.41 **13.4** -3.41 -6.06 -2.63 -0.12 -3.31 6.76 -8.62]
Case	Classification scores for the proposed backdoor sample
	
M_1 ("9")	[-0.16 2.36 -0.23 0.16 -2.89 -3.72 3.28 -2.58 -3.22 **7.62**]
M_2 (**"4"**)	[1.51 -4.18 -3.36 1.59 **12.3** -2.67 1.87 -1.96 0.83 -1.04]
M_3 ("9")	[-1.93 0.21 -2.17 -2.38 0.31 -2.57 3.43 -2.36 2.03 **7.71**]
M_4 ("9")	[7.42 3.13 -3.17 -4.33 -12.7 -6.22 1.24 -0.28 -10.1 **19.5**]
Case	Classification scores for the proposed backdoor sample
	
M_1 ("5")	[-0.04 10.4 -3.54 -0.42 -12.9 **22.4** -7.31 1.17 1.26 -12.6]
M_2 ("5")	[-1.63 2.66 -3.59 0.12 -12.4 **18.4** -8.82 1.16 5.81 -0.63]
M_3 (**"7"**)	[-1.89 4.52 0.58 -3.28 -10.4 -9.12 -0.82 **19.3** 7.48 -7.99]
M_4 ("5")	[-5.24 1.62 2.22 0.54 -0.36 **14.2** -10.6 -3.44 0.69 -1.86]
Case	Classification scores for the proposed backdoor sample
	
M_1 ("7")	[2.49 -0.55 2.93 1.74 -2.89 -3.07 -7.57 **8.85** -2.53 -0.91]
M_2 ("7")	[1.52 1.71 2.63 -0.05 1.39 -2.75 -7.73 **9.32** -3.41 -0.83]
M_3 ("7")	[-0.22 1.42 5.92 2.86 -2.93 -4.67 -10.7 **9.83** -1.51 -1.01]
M_4 (**"9"**)	[-0.35 0.82 2.51 -4.49 -1.47 -2.55 0.25 4.64 -4.62 **9.34**]

As shown in Table 2, backdoor samples for MNIST and Fashion-MNIST were set to random target classes with white squares as triggers, positioned at the top left, top right,

Table 3 Model selectivity, average attack success rate, and average accuracy of the models on the original samples for BadNet, Neural Cleanse, and the proposed scheme.

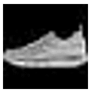
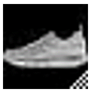
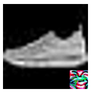
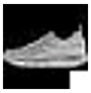
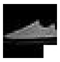
Description	Original	BadNet	Neural Cleanse	Proposed
Backdoor sample				
Model selection attack?	-	No	No	Yes
Attack success rate	-	100%	100%	100%
Accuracy	-	90.7%	91.1%	90.9%

Table 2 Sampling of multi-model selective backdoor samples for MNIST and Fashion-MNIST. The target class for M_1 was 1, that for M_2 was 4, that for M_3 was 7, and that for M_4 was 9.

MNIST					
Fashion-MNIST					

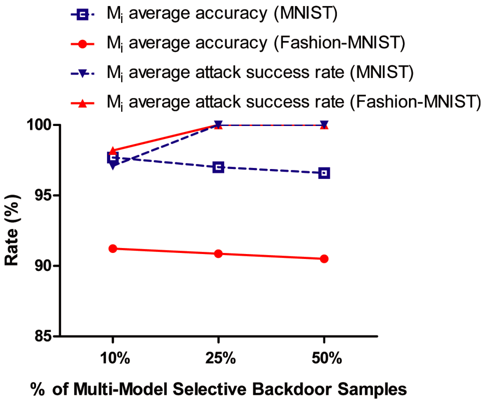


Fig. 2 Average accuracy of each model on the original samples and average attack success rate according to the proportion of backdoor samples in the training data, for MNIST and Fashion-MNIST.

bottom left, or bottom right. Figure 2 shows the average accuracy of each model on the original samples and the average attack success rate according to the proportion of backdoor samples. As shown in the figure, the higher the proportion of backdoor samples, the higher the attack success rate; the accuracy remained almost constant. It can be seen that when the proposed backdoor sample quantity was about 25%, the proposed method had an attack success rate of 100% for each model while maintaining 97.1% and 90.9% accuracy on the original samples for MNIST and Fashion-MNIST.

For an analysis comparing different methods, the state-of-the-art method, namely Neural Cleanse [10], and the BadNet method [6] were compared with the proposed method. Table 3 shows the presence or absence of the model selectivity feature, attack success rate via training with backdoor samples, and accuracy on the original samples for BadNet, Neural Cleanse, and the proposed method with Fashion-MNIST. In terms of model selectivity, the BadNet and Neural Cleanse methods are attacks on a single model; selective attacks in an environment with multiple models are not considered. The proposed method, however, can selectively attack a specific model in an environment with multiple models by using the trigger position. In terms of attack success rates, the performance of the proposed method is very similar to that of the other methods. In terms of accuracy, the performance of the proposed method is, again, similar to that of the other methods. In summary, the proposed method can perform a selective attack on a specific model in a multi-model environment with the same attack success rate as the other methods while maintaining the models' accuracy on the original data.

4. Conclusions

In this paper, we propose a multi-model selective backdoor attack method that misleads models to different target classes according to the position of the trigger on the backdoor sample. This scheme performs additional training of multiple models on backdoor samples that are incorrectly classified according to a specific trigger position for each model. Experimental results show that the proposed method has a 100% average attack success rate for each model while maintaining 97.1% and 90.9% accuracy on the original samples for MNIST and Fashion-MNIST, respectively. The proposed concepts can be applied to the audio/video domain in future studies. In addition, defense mechanisms against multi-model selective backdoor attacks will be a challenging topic for future research.

Acknowledgments

This study was supported by National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (2021R1I1A1A01040308) and Hwarang-Dae Research Institute of Korea Military Academy.

References

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

[2] 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.

[3] J. Su, D.V. Vargas, and K. Sakurai, "One pixel attack for fooling

deep neural networks," IEEE Trans. Evol. Comput., vol.23, no.5, pp.828–841, 2019.

[4] J. Su, D.V. Vargas, and K. Sakurai, "Attacking convolutional neural network using differential evolution," IPSJ Transactions on Computer Vision and Applications, vol.11, no.1, pp.1–16, 2019.

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

[6] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," IEEE Access, vol.7, pp.47230–47244, 2019.

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

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

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol.86, no.11, pp.2278–2324, 1998.

[10] 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), pp.707–723, IEEE, 2019.

Appendix

Table A·1 Model M_i architecture for MNIST and Fashion-MNIST.

Layer type	Shape
Convolution+ReLU	[3, 3, 32]
Convolution+ReLU	[3, 3, 32]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Fully connected+ReLU	[200]
Fully connected+ReLU	[200]
Softmax	[10]

Table A·2 Model M_i parameters.

Parameter	Value
Learning rate / Momentum	0.1 / 0.9
Dropout / Delay rate	0.5 / -
Batch size	128
Number of epochs	50

Table A·3 Accuracy of pretrained models M_i.

Model	Training data	MNIST	Fashion-MNIST
M_1	0–5,000	97.75%	91.23%
M_2	5,000–10,000	97.74%	91.45%
M_3	15,000–20,000	97.44%	91.23%
M_4	25,000–30,000	97.56%	91.96%