symmetry

MDPI

Article

Selective Poisoning Attack on Deep Neural Networks [†]

Hyun Kwon [1,‡], Hyunsoo Yoon [1] and Ki-Woong Park [2,*]

[1] School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea
[2] Department of Computer and Information Security, Sejong University, Seoul 05006, Korea
[*] Correspondence: woongbak@sejong.ac.kr; Tel.: +82-10-9165-1624
[†] This paper is an extended version of our paper published in IEEE AIKE 2019.
[‡] Current address: KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea.

Received: 14 June 2019; Accepted: 1 July 2019; Published: 8 July 2019

Abstract: Studies related to pattern recognition and visualization using computer technology have been introduced. In particular, deep neural networks (DNNs) provide good performance for image, speech, and pattern recognition. However, a poisoning attack is a serious threat to a DNN's security. A poisoning attack reduces the accuracy of a DNN by adding malicious training data during the training process. In some situations, it may be necessary to drop a specifically chosen class of accuracy from the model. For example, if an attacker specifically disallows nuclear facilities to be selectively recognized, it may be necessary to intentionally prevent unmanned aerial vehicles from correctly recognizing nuclear-related facilities. In this paper, we propose a selective poisoning attack that reduces the accuracy of only the chosen class in the model. The proposed method achieves this by training malicious data corresponding to only the chosen class while maintaining the accuracy of the remaining classes. For the experiment, we used tensorflow as the machine-learning library as well as MNIST, Fashion-MNIST, and CIFAR10 as the datasets. Experimental results show that the proposed method can reduce the accuracy of the chosen class by 43.2%, 41.7%, and 55.3% in MNIST, Fashion-MNIST, and CIFAR10, respectively, while maintaining the accuracy of the remaining classes.

Keywords: poisoning attack; machine learning; deep neural network; chosen class

1. Introduction

Studies related to pattern recognition and visualization using computer technology have been introduced. In particular, deep neural networks (DNNs) [1] have performed excellently in machine learning tasks such as recognition [2,3] and pattern analysis [4]. Particularly, the genetic algorithm [5] and the ant colony algorithm [6] using bio-inspired methods [7] have shown improved image recognition results. The genetic algorithm can find the most optimally performing convolutional neural network (CNN) structure among given CNN models for an image task without pre- or post-processing. The ant colony algorithm, on the other hand, is an optimal algorithm for detecting the edge of an image and is used to find a good solution for the optimization problem. Despite the success of these DNNs, they are vulnerable to attack. Two attack methods [8] threaten the security of DNNs, namely, causative attack [9] and exploratory attack [10]. A causative attack degrades the accuracy of the model by approaching the training process of the model. On the other hand, an exploratory attack exploits the misclassification of models without affecting the training process. A causative attack has the advantage of directly attacking the model over an exploratory attack.

A poisoning attack [9] is a typical causative attack and reduces the accuracy of the model by adding malicious data to the training process of the model. This attack is a critical threat to the medical field and autonomous vehicles where the accuracy of the model is important. For example, in the case of an autonomous vehicle, a serious accident may occur when a vehicle misidentifies a road sign due to a poisoning attack. Similarly, in the case of medical profession, if a model is misdiagnosed as a computerized tomography (CT) scan for a patient due to a poisoning attack, it may pose a health risk to the patient. Conventional studies on such poisoning attacks have focused only on the reduction in the overall accuracy of the model.

However, it may be necessary to reduce the accuracy of a single chosen class of the model in certain situations. In other words, there may be a need to make an attacker incapable of recognizing the model properly for certain classes. Such scenarios can be applied to face recognition systems, vehicle detection systems, submarines, and unmanned aerial vehicles (UAVs). In the first example, there is a need to prevent external tracking of very important persons (VIP), such as celebrities, through face recognition systems due to security concerns. Therefore, it may be necessary to deliberately attack the model to make certain VIPs unrecognizable. In the second case, important vehicles, such as banking vehicles, should not be tracked by vehicle recognition systems, and such vehicles need to be misrecognized by the systems. In the case of submarines, there is a need to avoid detection by the enemy. Therefore, it may be necessary to attack the enemy detection devices so that they do not recognize a specific submarine properly. In the last case, there may be a need to prevent UAVs from detecting nuclear-related facilities specifically. In such cases, it is important to ensure that only the nuclear facilities are misrecognized and other facilities are correctly recognized.

In this paper, we propose a selective poisoning attack that reduces the accuracy of a chosen class in a model. When the training data are accessed, the proposed method intentionally adds malicious data corresponding to a chosen class to decrease its accuracy and maintains the accuracy of other classes. This paper is an extended version of our previous work [11] presented at the IEEE International Conference on Artificial Intelligence and Knowledge Engineering (IEEE AIKE) 2019 conference. The contribution of this paper is as follows:

- We proposes a selective poisoning attack method. We systematically organize the framework and describe the principle of the proposed scheme.
- We analyze the selective accuracy depending on the number of selective malicious data. We also analyze the iteration, distortion, and accuracy for the selective malicious data.
- Through experiments using MNIST [12], Fashion-MNIST [13], and CIFAR10 [14], we demonstrate the effectiveness of the proposed scheme. We present image samples of the malicious data with the chosen class for each dataset, and the selected malicious data are difficult to detect through human perception.

The remainder of this paper is as follows. Section 2 introduces related research, and Section 3 introduces the proposed method. The experiment is described and evaluated in Section 4. A discussion on the proposed method is presented in Section 5. Finally, we draw our conclusions in Section 6.

2. Related Work

Recently, studies related to pattern recognition and visualization using a DNN have been introduced. In this section, we introduce some research related to DNN related security issues. The security issue in DNNs was first introduced by Barreno et al. [8]. Although the attack methods for DNN security can be classified separately into exploratory and causative, in this section, we introduce them as a whole and describe a basic neural network.

2.1. Neural Networks

A neural network [2] is a machine-learning algorithm that models the brain's learning method mathematically; it refers to an overall model that forms a network through a combination of neurons and synapses. The architecture of a neural network comprises an input layer, a hidden layer, and an output layer. In the input layer, there is a neuron for each input variable, which is matched 1:1. In the hidden layer, there are neurons generated by the combination of the neurons and weights of the input layer; the number of layers within the hidden layer determines the complexity of the model. The type of output to be predicted determines the number of output layers. Combining the neurons and weights of the hidden layer creates neurons in the output layer. The sum of the input values and weights in the previous layer is calculated by the neurons in the hidden and output layers. In addition, an activation function provides the weighted sum of the neurons in the previous layer. A neural network undergoes training with training data and sets the parameters for each layer with optimal loss values using calculating gradient descent and backpropagation. Since neural networks depend on the training data to set the optimum parameters, a poisoning attack accesses the training data. The following section introduces and describes poisoning attacks.

2.2. Exploratory Attack

An exploratory attack exploits the misclassification of models without affecting the training process. A typical exploratory attack is the adversarial example attack. An adversarial example attack is a deformed sample, with some disturbance added to the original sample, to make the model liable to be misinterpreted, and a person cannot identify the disturbance. This attack's characteristics do not affect the training data of the DNN. The adversarial example was first introduced by Szegedy et al. [15], and various attack and defense methods based on it have been introduced since. In terms of attack, the white box attack [16] has an almost 100% success rate, so the black box attack [17–19] is mainly studied. In terms of defense, a method for constructing a robust model for denying adversarial example attacks has been studied by manipulating the input data [20,21] or by changing the model [22]. In addition, the study of adversarial examples is expanding not only in the field of images but also of voice [23,24] and video [25]. In terms of assumption, there is a relatively realistic aspect to it because the exploratory attack does not approach the training data of the model. However, in terms of a real-time attack, there is a limitation because an exploratory attack requires time and process to transform the test data.

2.3. Causative Attack

A causative attack degrades the accuracy of the model by approaching the training process of the model. A poisoning attack [9,26,27] is a causative attack method that reduces the accuracy of a model by adding malicious data between the processes during the training of the model. There is a strong possibility that this attack will have access to the training process of the model, but it has the advantage of effectively reducing the accuracy of the model. Biggio et al. [9] were the first to propose a poisoning attack against a support vector machine (SVM) for reducing its accuracy by injecting malicious data in the training data. This method aims to calculate a gradient descent based on the characteristics of the SVM to generate some point samples that can be dropped by maximizing the accuracy of the SVM. Yang et al. [26] proposed a poisoning attack method against neural networks rather than SVM models. Their method uses a direct gradient method to generate data with a generative adversarial net (GAN) [28] through an auto-encoder. This method sets the target model as a discriminator and the generator searches for optimal malicious data from the discriminator by a zero-sum method. Mozaffari-Kermani et al. [27] proposed similar systematic poisoning attacks in healthcare. With their method, they demonstrated a poisoning attack on a healthcare dataset by extending the domain to medicine.

The conventional poisoning attack studies mentioned above aimed to reduce the accuracy of the whole model. However, in certain cases, it may be necessary to reduce the accuracy only of certain classes. We propose a method to do so by adding malicious data corresponding to only the specific classes desired by the attacker to the training process, while maintaining the recognition rate for the remaining classes.

3. Proposed Scheme

3.1. Threat Model

The target model of the proposed method is a system comprising neural networks. The method can be applied to systems such as image recognition [29], face recognition [30], and autonomous vehicles [31], which can be operated by a neural network. The proposed method assumes that the attacker can access the model through a white box method, where the attacker knows about the structure, parameters, and the output classification.

3.2. Proposed Method

The purpose of the proposed scheme is to add selective malicious data in the training process as a poisoning attack, which lowers the accuracy of a chosen class. Figure 1 shows an overview of the proposed method. As shown in the figure, malicious data corresponding to a class chosen by the attacker are added to the training data.

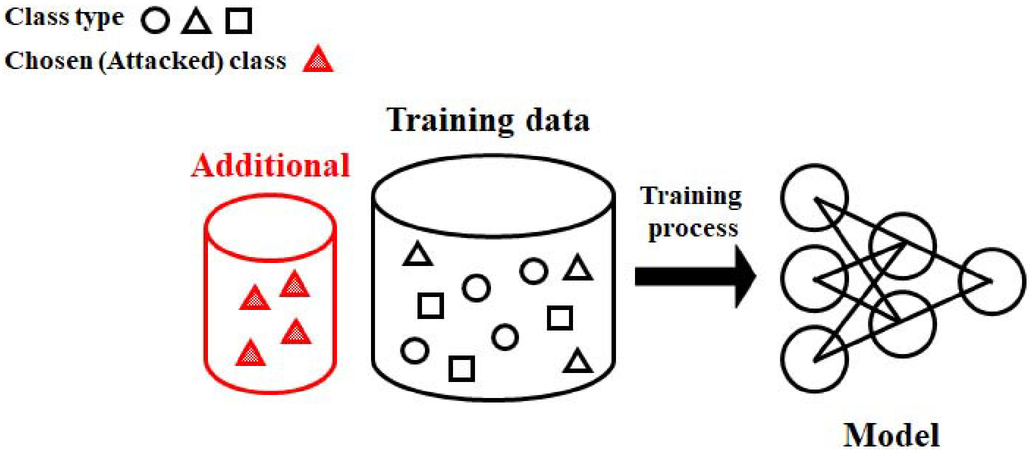


Figure 1. A overview of the proposed scheme.

The proposed method is divided into two steps: generation of malicious data and malicious data addition to the training data. The first step generates malicious data $x_i^{'} \in X^{'}(1 \leq i \leq N^{'})$ as follows. Given the original training data $x_i \in X$ with a chosen class $y^{'}$, it generates malicious data $x_i^{'}$ with the smallest probability to be recognized as a specific class $y^{'}$ by the model. To achieve this, *loss* must be minimized:

$$loss = Z(x_i^{'})_{y^{'}} - \max\left\{Z(x_i^{'})_i : i \neq y^{'}\right\}, \tag{1}$$

where $Z(\cdot)$ [32] represents the pre-softmax classification result vector of model M. The malicious data can lower the probability of a specific class $y^{'}$ by optimally minimizing *loss*. By minimizing *loss* during a given iteration l, the proposed method generates malicious data $x_i^{'}$ that modulates the original training data x_i and lowers the accuracy of the chosen class $y^{'}$ in model M.

In the second step, the model M undergoes the training process of both x_i and $x_i^{'}$ with the given original training data $x_j \in X(1 \leq j \leq N)$ with N instances and malicious data $x_i^{'} \in X^{'}$ with $N^{'}$ instances corresponding to a chosen class $y^{'}$. Then, we use the test dataset to measure the accuracy of the model M. The detailed procedure for proposed scheme is given as Algorithm 1.

Algorithm 1 Selective poisoning attack

```
Description:
    $x_j \in X$ with $N$ instances                                   ▷ original training dataset
    $x_i^{'} \in X^{'}$ with $N^{'}$ instances                       ▷ maliciously manipulated training data
    $l$                                                              ▷ number of iterations
    $t$                                                              ▷ test data
    $y^{'}$                                                          ▷ chosen class

Selective poisoning attack: $(x_i, y_i^{'}, l, N^{'})$
    for $i$ =1 to $N^{'}$ do
        Find $x_i$ with selective class $y^{'}$
        $x_i^{'} \leftarrow$ Generation malicious instance $(x_i, y^{'}, l)$
        Assign $x_i^{'}$ to $X^{'}$
    end for
    A temporary training set $X_T \leftarrow X + X^{'}$
    Build the model $M$ training $X_T$
    Record its classification accuracy on the test dataset $t$
    return $M$
Generation malicious instance: $(x_i, y^{'}, l)$
    $x_i^{'} \leftarrow x_i$
    for $l$ step do
        $loss \leftarrow Z(x_i^{'})_{y^{'}} - \max \left\{ Z(x_i^{'})_i : i \neq y^{'} \right\}$
        Update $x_i^{'}$ by minimizing the gradient of $loss$
    end for
    return $x_i^{'}$
```

4. Experiment and Evaluation

Through experiments, the proposed method demonstrated that a selective poisoning attack reduced the accuracy of a chosen class in the model. We used Tensorflow [33] as the machine-learning library and an Intel(R) i5-7100 3.90-GHz server.

4.1. Datasets

MNIST [12], Fashion-MNIST [13], and CIFAR10 [14] were used in the experiment. MNIST contains handwritten images of digits from 0 to 9 and is a standard dataset. Fashion-MNIST is a more complex fashion image dataset than MNIST and consists of ten types of images: T-shirt, trouser, pullover, dress, etc. MNIST and Fashion-MNIST comprise (28, 28, 1)-pixel matrices. Both have the advantages of fast learning time in experiments due to the one-dimensionality of the images. In addition, 60,000 training data and 10,000 test data were used both for MNIST and Fashion-MNIST, respectively. CIFAR10 contains color images in 10 classes: planes, cars, birds, etc. CIFAR10 comprises (32, 32, 3)-pixel matrices with three-dimensional images and is widely used in machine learning experiments. CIFAR10 consists of 50,000 training data and 10,000 test data.

4.2. Pretraining of Models

The model M pretrained on MNIST, Fashion-MNIST, and CIFAR10 were a common convolutional neural network [34] and a VGG19 network [2], respectively. Their configuration and parameters are shown in Tables A1–A3 of the Appendix A. For MNIST and Fashion-MNIST, 60,000 training data were used. In the test, the original MNIST and Fashion-MNIST samples were correctly classified by the pretrained model with 99.25% and 92.45% accuracy. For CIFAR10, 50,000 training data were used. In the test, the original CIFAR10 samples were correctly classified by the pretrained model with 91.24% accuracy.

4.3. Generation of Malicious Training Data

To show the performance of the proposed method, the proposed scheme was used to generate 2500 malicious training data from 2500 random training data. For the poisoning process, we used the box constraint method and Adam [35] as an optimizer. For MNIST and Fashion-MNIST, the number of iterations was set to 400, the learning rate was set to 0.1, and the initial value was set to 0.01. For CIFAR10, the number of iterations was set to 6000, the learning rate was set to 0.01, and the initial value was set to 0.01.

4.4. Experimental Results

The term accuracy means the matching rate between the chosen class and the original class that the model recognizes. The chosen class accuracy conveys the matching rate between the class recognized by the model and the chosen class. Distortion is the root sum of the square root of the difference between the original training sample and the malicious data in the L_2 distortion measure.

Table 1 shows an example of selective poisoning of data for each chosen class in MNIST. It shows selective poisoning examples with the same class in each row. As the table shows, the attacker selects malicious data corresponding to each chosen class that he wants to attack as a numeric image from 0 to 9. After that, the selected malicious data are additionally trained in the model *M*, according to the table that noise is added to the original training data of model M to reduce the accuracy of the chosen class. Table 2 shows an example of selective poisoning of data for each chosen class in Fashion-MNIST. It shows selective poisoning examples with the same class in each row. Similar to MNIST, the attacker selects malicious data corresponding to each chosen class that he wants to attack as a fashion image from a T-shirt to ankle boots to reduce the chosen class accuracy in model *M*. Unlike MNIST, Fashion-MNIST has little influence on human recognition rate because the outline of the image is clear. Table 3 shows an example of selective poisoning of data for each chosen class in CIFAR10. Similar to MNIST, the attacker selects malicious data corresponding to each chosen class that he wants to attack as a color object image from a plane to a truck to reduce the chosen class accuracy in model *M*. However, since CIFAR10 has color images, the noise cannot be detected as clearly as compared with that in MNIST or Fashion-MNIST.

Table 1. Sampling of selective posioning examples with each chosen class in MNIST.

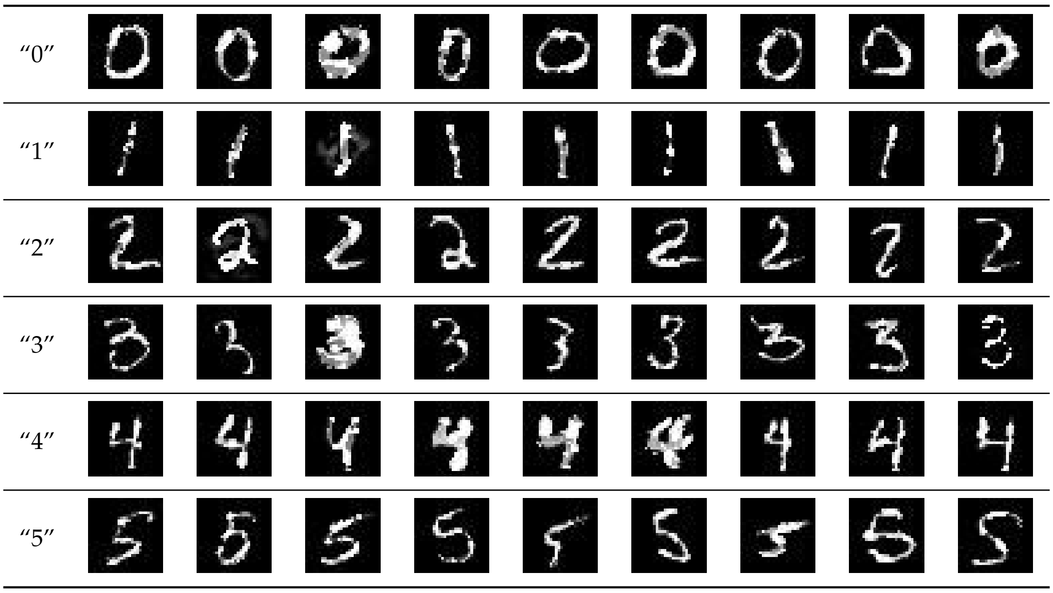


Table 1. *Cont.*

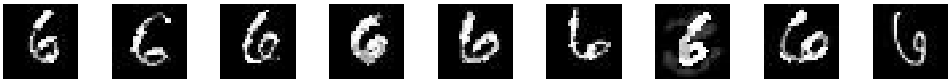
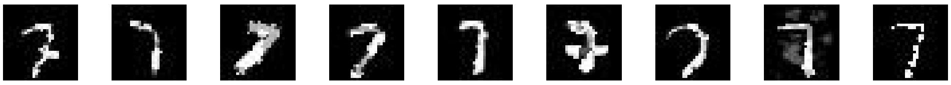
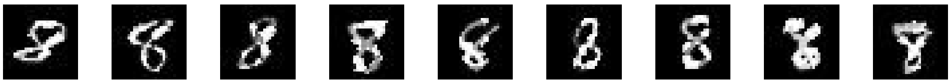
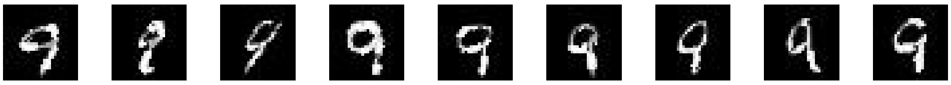
"6"	
"7"	
"8"	
"9"	

Table 2. Sampling of selective posioning examples with each chosen class in Fashion-MNIST.

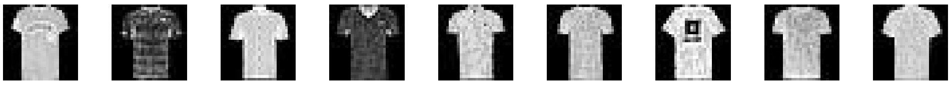
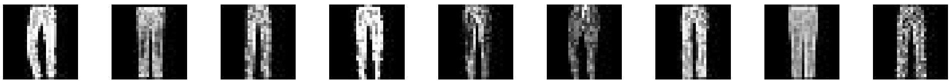
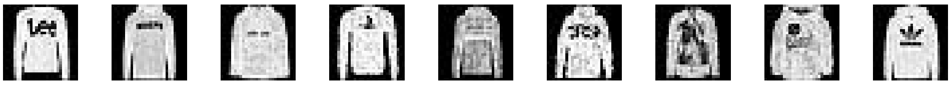
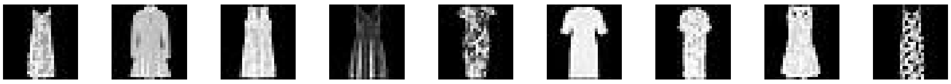
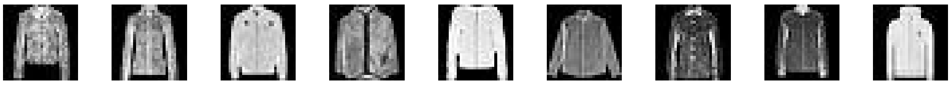
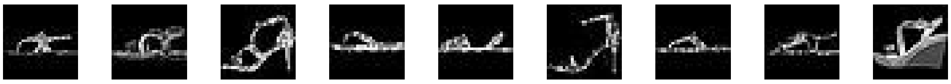
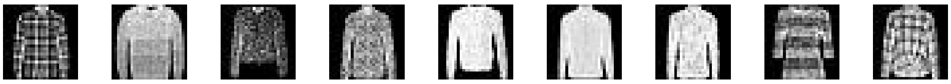
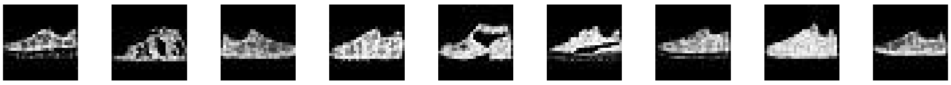
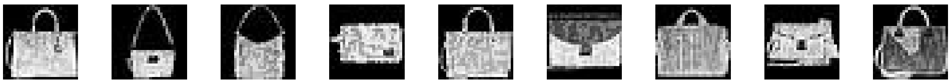
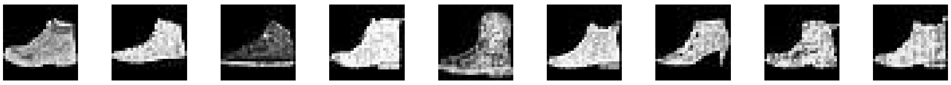
"T-shirt"	
"Trouser"	
"Pullover"	
"Dress"	
"Coat"	
"Sandals"	
"Shirt"	
"Sneaker"	
"Bag"	
"Ankle boots"	

Table 3. Sampling of selective posioning examples with each chosen class in CIFAR10. It shows selective poisoning examples with the same class in each row.

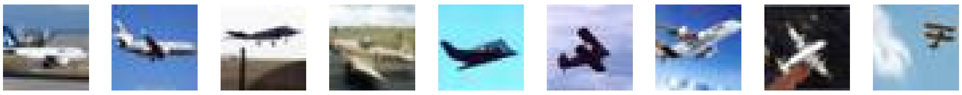
"Plane"	

Table 3. *Cont.*

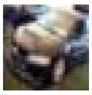
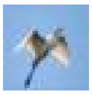
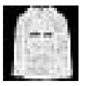
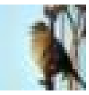
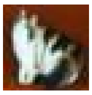
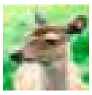
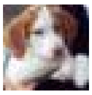
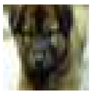
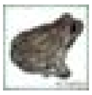
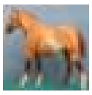
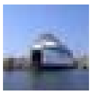
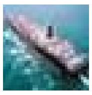
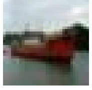
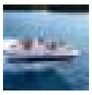
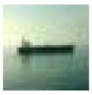
"Car"									
"Bird"									
"Cat"									
"Deer"									
"Dog"									
"Frog"									
"Horse"									
"Ship"									
"Truck"									

Figure 2 shows the chosen class accuracy of the model according to the number of selective malicious data. The chosen class was randomly selected. According to the figure, selective class accuracy decreases, as the number of selective malicious data increases. In particular, this figure shows that as the number of relatively malicious data increases, the rate of decrease gets faster. In addition, the accuracy of the chosen class is different for each dataset, as accuracy in MNIST and Fashion-MNIST reduces faster than CIFAR10.

Table 4 shows the iteration, average distortion, total accuracy, and chosen class accuracy when the number of malicious data are 2500. In the table, it can be seen that the total accuracy is reduced as the selective accuracy is reduced. However, it can be seen that the chosen class accuracy decreases significantly. In terms of iteration and distortion, Fashion-MNIST and MNIST show relatively lesser values than CIFAR10.

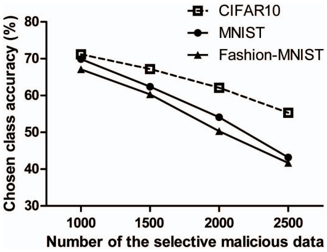


Figure 2. Chosen class accuracy of the model *M* according to the number of the selective malicious data.

Table 4. The iteration, average distortion, total accuracy, and chosen class accuracy of M when the number of the selective malicious data are 2500.

Description	MNIST	Fashion-MNIST	CIFAR10
Iteration	400	400	6000
Average distortion	3.56	2.58	67.24
Total accuracy	89.7%	81.2%	80.9%
Accuracy of chosen class	43.2%	41.7%	55.3%

5. Discussion

Attack considerations. In terms of the model, the chosen class accuracy can be changed according to the accuracy of the model. The accuracy is affected by the poisoning attack according to the classification result of the existing model. Moreover, the attacker needs to also consider the number of malicious data because the accuracy of a particular class depends on that number.

Applications. The proposed method can be applied to sensor systems. A sensor is based on the Internet of things (IoT) and displays the numerical value of the external environment or an image such as from CCTV. By applying a poisoning attack to such a sensor system, the performance of a specific part can be reduced. The proposed method can be similarly used in military applications and face recognition systems. If an attacker has to be prevented from recognizing a particular class correctly, the proposed method can be used to lower the accuracy of that particular class without compromising on the overall accuracy.

Dataset. According to the dataset used (MNIST, Fashion-MNIST, or CIFAR10), the selected class accuracy, iteration, and distortion in the proposed method are different. CIFAR10 is a three-dimensional image dataset with a 3072 (32, 32, 3)-pixel matrix and MNIST and Fashion-MNIST are one-dimensional image datasets with 784 (28, 28, 1)-pixel matrices. Therefore, since the number of pixels is relatively large, CIFAR10 has more iterations and distortion than MNIST and Fashion-MNIST.

In terms of the defense. To defend against a poisoning attack, management of training data are needed when training the model. Because the malicious data generated by the proposed method are similar to the original sample, they are difficult to identify with the human eye. Therefore, the integrity of the training data must be checked by comparing the number and the hash value. In addition, it is necessary to fix the setting value of the parameter of the model so that additional learning cannot be performed on the model after the training is completed.

Comparison with the adversarial example. The adversarial example and poisoning attacks differ in their target attacking methods. The adversarial example modulates the test data while a poisoning attack modifies the model parameters using malicious training data. For example, in a face recognition system, an adversarial example will manipulate the face of a certain person to deceive a specific person, but a poisoning attack will approach the training data of the model in advance to lower the recognition accuracy of a specific person.

6. Conclusions

In this paper, we proposed a selective poisoning attack method that reduces the accuracy of a chosen class. This method reduces the accuracy of a chosen class by adding malicious data for that class. Experimental results show that the proposed method can reduce the accuracy of a chosen class by 43.2%, 41.7%, and 55.3% in MNIST, Fashion-MNIST, and CIFAR10, respectively. We also showed that the proposed scheme can be applied to face recognition systems, autonomous vehicles, and in the medical field.

Future studies will expand not only in the image domain but also extend to the voice or video domains. In addition, the GAN method [28] can be used to generate malicious data.

Future studies will expand not only the image but also the voice or video domain. In addition, a generative adversarial net method [28] can be used to generate malicious data. Future research may suggest ways, such as hash value checking, to defend against this method.

Author Contributions: H.K. designed the entire architecture and performed the experiments. H.Y. and K.-W.P. gave some advice and ideas about the article. All authors analyzed the result of this article. Hyun Kwon wrote the article. All authors discussed the contents of the manuscript.

Acknowledgments: This work was supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00420) and supported by the National Research Foundation of Korea (NRF) (NRF-2017R1C1B2003957 and 2017R1A2B4006026).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. *M* 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 A2. *M* model parameters.

Parameter	MNIST and Fashion-MNIST	CIFAR10
Learning rate	0.1	0.001
Momentum	0.9	0.9
Batch size	128	128
Epochs	50	50
Dropout/Delay rate	-	0.5/10

Table A3. *M* model architecture [34] for CIFAR10.

Layer Type	CIFAR10 Shape
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 128]
Convolution+ReLU	[3, 3, 128]
Max pooling	[2, 2]

Table A3. *Cont.*

Layer Type	CIFAR10 Shape
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Fully connected+ReLU	[4096]
Fully connected+ReLU	[4096]
Softmax	[10]

References

1. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
2. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR2015), San Diego, CA, USA, 7–9 May 2015.
3. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
4. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; ACM: New York, NY, USA, pp. 160–167.
5. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G. Automatically designing CNN architectures using genetic algorithm for image classification. *arXiv* **2018**, arXiv:1808.03818.
6. Banharnsakun, A. Artificial bee colony algorithm for enhancing image edge detection. *Evolv. Syst.* **2018**, 1–9. [CrossRef]
7. Deng, L; Wang, Y; Han, Z; Yu, R. Research on insect pest image detection and recognition based on bio-inspired methods. *Biosyst. Eng.* **2018**, 139–148. [CrossRef]
8. Barreno, M.; Nelson, B.; Joseph, A.D.; Tygar, J. The security of machine learning. *Mach. Learn.* **2010**, *81*, 121–148. [CrossRef]
9. Biggio, B.; Nelson, B.; Laskov, P. Poisoning attacks against support vector machines. In Proceedings of the 29th International Coference on International Conference on Machine Learning, Edinburgh, Scotland, 27 June–3 July 2012; Omnipress: Madison, WI, USA, 2012; pp. 1467–1474.
10. McDaniel, P.; Papernot, N.; Celik, Z.B. Machine learning in adversarial settings. *IEEE Secur. Priv.* **2016**, *14*, 68–72. [CrossRef]
11. Kwon, H.; Yoon, H.; Park, KW. Poisoning Attack on Deep Neural Network to Induce Fine-Grained Recognition Error. In Proceedings of the IEEE International Conference on Artificial Intelligence and Knowledge Engineering, Cagliari, Italy, 5–7 June 2019. Available online: http://hdl.handle.net/10203/262522 (accessed on 6 July 2019).
12. LeCun, Y.; Cortes, C.; Burges, C.J. MNIST Handwritten Digit Database. *AT&T Labs* **2010**, *2*. Available online: http://yann.lecun.com/exdb/mnist (accessed on 1 July 2019).
13. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.

14. Krizhevsky, A.; Nair, V.; Hinton, G. The CIFAR-10 Dataset. 2014. Available online: http://www.cs.toronto.edu/kriz/cifar.html (accessed on 1 July 2019).
15. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations 2014, Banff, AB, Canada, 14–16 April 2014.
16. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 39–57.
17. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; ACM: New York, NY, USA, 2017; pp. 506–519.
18. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv* **2017**, arXiv:1611.02770.
19. Moosavi Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; number EPFL-CONF-226156.
20. Meng, D.; Chen, H. Magnet: a two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; ACM: New York, NY, USA, 2017; pp. 135–147.
21. Shen, S.; Jin, G.; Gao, K.; Zhang, Y. Ape-gan: Adversarial perturbation elimination with gan. *arXiv* **2017**, arXiv:1707.05474.
22. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 582–597.
23. Carlini, N.; Mishra, P.; Vaidya, T.; Zhang, Y.; Sherr, M.; Shields, C.; Wagner, D.; Zhou, W. Hidden Voice Commands. In Proceedings of the 25th USENIX Security Symposium, Austin, TX, USA, 10–12 August 2016; pp. 513–530.
24. Carlini, N.; Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24–24 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.
25. Li, S.; Neupane, A.; Paul, S.; Song, C.; Krishnamurthy, S.V.; Chowdhury, A.K.R.; Swami, A. Adversarial perturbations against real-time video classification systems. *arXiv* **2018**, arXiv:1807.00458.
26. Yang, C.; Wu, Q.; Li, H.; Chen, Y. Generative Poisoning Attack Method Against Neural Networks. *arXiv* **2017**, arXiv:1703.01340.
27. Mozaffari-Kermani, M.; Sur-Kolay, S.; Raghunathan, A.; Jha, N.K. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE J. Biomed. Health Inf.* **2015**, *19*, 1893–1905. [CrossRef] [PubMed]
28. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
29. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8697–8710.
30. Ding, C.; Tao, D. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1002–1014. [CrossRef] [PubMed]
31. Farag, W.; Saleh, Z. Traffic signs identification by deep learning for autonomous driving. *IET* **2018**. [CrossRef]
32. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The limitations of deep learning in adversarial settings. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrucken, Germany, 21–24 March 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 372–387.
33. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. *OSDI* **2016**, *16*, 265–283.
34. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

35. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.