Copyright ⓒ 2021 KSII

Data Correction For Enhancing Classification Accuracy By Unknown Deep Neural Network Classifiers

Hyun Kwon[1], Hyunsoo Yoon[2], and Daeseon Choi[3*]
[1] Department of Electrical Engineering, Korea Military Academy
Seoul, 01805, Republic of Korea
[e-mail: hkwon.cs@gmail.com or khkh@kaist.ac.kr]
[2] School of Computing, Korea Advanced Institute of Science and Technology
Daejeon, 34141, Republic of Korea
[e-mail: hyoon@kaist.ac.kr]
[3] Department of Software, Soongsil University
Seoul, 06978, Republic of Korea
[e-mail: sunchoi@ssu.ac.kr]
[*]Corresponding author: Daeseon Choi

Received July 2, 2019; revised March 31, 2021; accepted July 8, 2021; published September 30, 2021

Abstract

Deep neural networks provide excellent performance in pattern recognition, audio classification, and image recognition. It is important that they accurately recognize input data, particularly when they are used in autonomous vehicles or for medical services. In this study, we propose a data correction method for increasing the accuracy of an unknown classifier by modifying the input data without changing the classifier. This method modifies the input data slightly so that the unknown classifier will correctly recognize the input data. It is an ensemble method that has the characteristic of transferability to an unknown classifier by generating corrected data that are correctly recognized by several classifiers that are known in advance. We tested our method using MNIST and CIFAR-10 as experimental data. The experimental results exhibit that the accuracy of the unknown classifier is a 100% correct recognition rate owing to the data correction generated by the proposed method, which minimizes data distortion to maintain the data's recognizability by humans.

Keywords: Data correction, Deep neural network, Ensemble method, Machine learning, Poisoning attack

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2020R1A2C1014813) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A1A01040308).

http://doi.org/10.3837/tiis.2021.09.009 **ISSN : 1976-7277**

1. Introduction

Deep neural networks (DNNs) [1] perform well in machine learning tasks such as image classification [2], audio recognition [3], and pattern recognition [4]. As DNNs are used in autonomous vehicles and medical applications, their accuracy is important. For example, an autonomous vehicle equipped with a DNN must recognize signs correctly; it can lead to accidents when the autonomous vehicle recognizes a sign incorrectly. Therefore, many studies [5, 6] have been conducted to increase the accuracy of a DNN by changing the DNN.

However, in some situations it may be difficult to change an unknown DNN classifier that has already been distributed; thus, in order to increase the recognition rate of an unknown DNN classifier, it may be necessary to take the approach of modifying the data. For example, when creating a road sign, it may be necessary to modify the sign so that it will be correctly recognized by any autonomous vehicle. In this case, as it would be difficult to change the classifier of an unknown autonomous vehicle that has already been distributed, data processing is necessary so that unknown autonomous vehicles and people will correctly recognize the sign. To create images or texts that will be correctly recognized by the machine, conventional methods [7] require image modulation, such as rotation and repositioning of the legitimate sample. However, these conventional methods do not consider the amount of distortion from the legitimate sample, recognizability to humans, or unknown classifiers. In terms of DNN security, the accuracy of the unknown classifier may also deteriorate as the result of a poisoning attack [8], in which malicious training data are inserted to the training dataset. If the accuracy of the unknown classifier has been reduced, it may be necessary to use a data modification method to increase its accuracy.

To address this need, it is possible to generate modified data that consist of the input sample with a small amount of noise added to increase the accuracy of an unknown classifier while minimizing the distortion from the input sample. This concept is based on DNN transferability. As with such transferability [9], there is a possibility that the input data correctly recognized by the local classifier will also be correctly recognized by another classifier. Thus, the slightly modified data that are correctly recognized by multiple local classifiers in the ensemble method may also be correctly recognized by the unknown classifier. Also, this method can be robust against poisoning attacks. Poisoning attack is a method of reducing the accuracy of the model by adding malicious data to the target model. Although the accuracy of the model is low due to a poisoning attack, the ensemble data correction method using transferability can improve the accuracy of the legitimate data.

In this paper, we propose a data correction method that improves the accuracy of an unknown classifier by modifying the input data. This method modifies the input data slightly so that the unknown classifier will correctly recognize the input data using multiple local classifiers that are known to the constructor of the modified data. The contributions of this paper are as follows.

- We propose a data correction method and explain the architecture for a system that implements the proposed scheme. The proposed method uses transferability by using several models in an ensemble method.
- We analyze the method in terms of accuracy and analyze the distortion used for data correction. We also analyzed the class in which the proposed sample is recognized and the required number of repetitions.

- Through experiments using MNIST [10] (a numeric image dataset) and CIFAR-10 [11] (a color object image dataset), we demonstrate the effectiveness of the proposed scheme. We also show that even if the accuracy of the unknown classifier is low because of the structure of the model or the effects of a poisoning attack, its accuracy can be increased by applying the proposed method.

The remainder of this paper is structured as follows: Related work is reviewed in Section 2. Section 3 presents the proposed scheme. The experimental setup and the results showing the proposed method's performance are presented in Sections 4 and 5, respectively. The proposed method is further discussed in Section 6. Finally, Section 7 demonstrate our conclusions.

2. Related Work

In this section, we explain the data correction method, transferability, poisoning attacks, and measures of distortion, all of which are related to the proposed method. The data correction method similar to the proposed scheme is introduced in Section 2.1, and the core principle of the proposed method, transferability, is explained in Section 2.2. In order to explain the proposed method that is robust against poisoning attack, related research on poisoning attack is introduced in Section 2.3, and to explain the distortion difference between the legitimate sample and the modified sample in Section 2.4.

2.1 Data Correction

There have been studies on optical character recognition (OCR), a typical technique used by machines to recognize characters accurately. This technique can take one of two approaches: a textual-features-based approach or a typographical-features-based approach. In the textual-features-based approach [12, 13], text is recognized by its physical location. This method divides each text into segments and extracts features to learn and recognize them. The typographical-features-based approach [7, 14], on the other hand, allows the machine to recognize characters correctly by changing fonts or by rotating or resizing them. These methods commonly change the legitimate sample so that the machine will recognize the text after it has been modified by rotation or resizing; they do not consider the distortion to the legitimate sample. OCR methods also do not consider poisoning attacks, unknown classifiers, or recognizability to humans. The proposed method improves the accuracy of unknown classifiers by having a data manipulation process to correctly recognize the input data using transferability. Details are described in Section 2.2.

2.2 Transferability

The proposed scheme is based on the principle of transferability [9]. Transferability is a property by which input data that are correctly recognized by several known models will also be correctly recognized by unknown models. The concept of transferability was first introduced with the adversarial example [9, 15], which is designed to be misrecognized by a DNN while maintaining human perceptibility and is created by inserting a small amount of noise to a legitimate sample. In the context of adversarial examples, transferability is the idea that an adversarial example created for an arbitrary local model known to the attacker can effectively attack other models as well. As transferability is useful for black box attacks, advanced research [16, 17] has been conducted using a proposed ensemble method. The method proposed in this paper is a modification of the adversarial example, whereby a modified sample with low distortion that is correctly recognized by several models can

increase the accuracy of unknown models.

2.3 Poisoning Attacks

Poisoning attack [8, 18, 19] is an attack method that lowers the accuracy of the model by adding malicious data to the model's training data. This attack has the advantage of largely reducing the accuracy of the model even with a small number of malicious training data. However, this method requires high assumptions that access the model's training data. First, Biggio et al. [8] demonstrated a poisoning attack targeting a support vector machine (SVM). The method is an attack that effectively reduces the accuracy of the SVM model by additionally inserting malicious data to the training data. This attack method calculates gradient descent from the characteristics of the SVM, creates a malicious sample that can reduce the accuracy of the SVM as much as possible, and adds it to the model's training data. Second, Yang et al. [18] proposed a method of the poisoning attack on neural networks. This method creates malicious data using a generative adversarial network that uses a direct gradient algorithm. Third, Mozaffari-Kermani et al. [19] demonstrated a poisoning attack on medical data. This method is meaningful in extending the poisoning attack from the image domain to the medical domain. Recently, studies on poisoning attack methods [20] that induce misrecognition of only a specific class have also been conducted. In another extension being developed, a backdoor sample [21, 22] with a specific trigger is designed to be misclassified as a target class when the specific trigger is present. The backdoor sample is an attack in which the triggered data is erroneously recognized by the model. The backdoor sample attack was proposed by Tianyou gu et al. [23] by the badnet method. In this method, a specific trigger was attached to the image, and it was incorrectly recognized by the model. In this method, it showed a 99% attack success rate against MNIST. Instead of adding malicious data, Liu et al [24] added additional external neural networks to attack. This method is an attack that adds a neuron neural network and causes the data to which a specific trigger is attached to be misrecognized by the model. Wang Bolun et al. [25] proposed a method to detect by reversing the trigger. In this method, various triggers were attached to analyze the method for backdoor attack. Clements and Lao [26] proposed a method of causing malfunction by planting a backdoor on hardware in neural network. In this method, it was used as the MNIST dataset, and the wrong neuron was added to the neural network so that it was misrecognized by a specific trigger by the model. This paper focus on defense for the poisoning attack in the machine security. Although the accuracy of the model is degraded due to a poisoning attack, the proposed method makes the input data recognized correctly to increase the accuracy of the model. The experimental results are described in Section 5.

2.4 Distortion Metrics

To measure the distortion metrics [27] between the modified example and the legitimate sample, three distance metrics are used: L_0, L_2, and L_∞ as follows.

$$L_p = \sum_{i=0}^{n} \sqrt[p]{\left|x_i - x_i^*\right|^p}$$

where x_i^* is the i^{th} pixel in modified example x^*, and x_i is the i^{th} pixel in legitimate sample x. The smaller this distance metrics, the more similar the modified example is to the legitimate sample. In this study, the L_2 distance metric was used.

3. Proposed Scheme

3.1 Assumption

The proposed method assumes a white box for multiple local models known to the constructor. However, the unknown classifier is a black box, and the constructor does not know the structure, architecture, or class probabilities of the unknown classifier. The constructor provides information on input data and input class to multiple local models between data corrections but provides only modified samples for unknown classifiers after data processing.

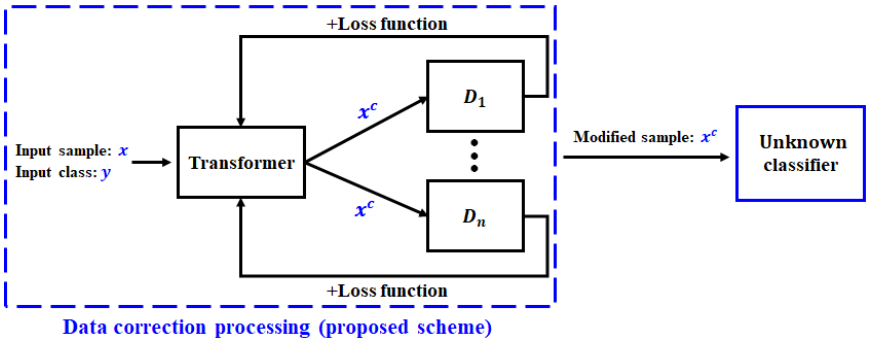


Fig. 1. Proposed architecture.

3.2 Proposed Method

Fig. 1 shows how the proposed method generates a slightly modified input sample that is correctly recognized by multiple classifiers known to the constructor before it is input to the unknown classifier.

The data correction system consists of a transformer and multiple local models D_i. The transformer receives input sample x and input class y as the input values. It then generates a modified sample x^c and provides it to the multiple local models D_i; these models do not change between generation processes. Each model D_i accepts the modified sample x^c as an input value and provides a loss function result to the transformer.

To increase the accuracy of the unknown classifier, the goal of the proposed scheme is to generate a modified sample x^c that is correctly recognized as input class y by multiple local models D_i while minimizing its distortion from the input sample x. In mathematical terms, the operation function of the multiple local models D_i is represented by $f_i(\cdot)$.

Given multiple local models D_i, input sample x, and input class y, finding a modified sample x^c means solving the following optimization problem:

$$x^c : \underset{x^c}{\operatorname{argmin}}\ L(x, x^c) \text{ such that } f_i(x^c) = y$$

where $L(x, x^c)$is the distance between input sample x and modified sample x^c.

The transformer generates the modified sample x^c after receiving the input sample x and the input class y as input values. In this study, the transformer was a modified configuration of the Carlini method [27], as follows:

$$x^c = \frac{\tanh(x^c + w)}{2}$$

where w is noise. Each model D_i accepts the modified sample x^c as an input value and

provides the transformer with its loss function result. If any of the models D_i misinterpret a modified sample x^c as an incorrect class rather than as the input class, the transformer repeats the above process until the modified sample x^c is correctly recognized by all of the models D_i as the input class, while minimizing the distortion from the input sample x.

The total loss $loss_{\mathrm{T}}$ is as follows:

$$loss_{\mathrm{T}} = loss_{\mathrm{distortion}} + \sum_{i=1}^{n} c_i \times loss_i$$

where $loss_{\mathrm{distortion}}$ is the distortion function of D_i, $loss_i$ is the classification loss function of D_i, and c_i is the loss weight of model D_i, whose initial value is set to 1. $loss_{\mathrm{distortion}}$ is the result of the distance function between the input sample x and the modified sample x^c:

$$loss_{\mathrm{distortion}} = \sqrt{\left(x^c - \frac{\tanh x}{2}\right)^2}$$

To satisfy $f_i(x^c) = y$, $loss_i$ must be minimized:

$$\sum_{i=1}^{n} loss_i = \sum_{i=1}^{n} g_i(x^c)$$

where $g_i(k) = \max\{Z_i(k)_j : j \neq y\} - Z_i(k)_y$, in which y is the input class and $Z_i(\cdot)$ [27] produces the probabilities of the classes being predicted by model D_i. By optimally minimizing $loss_i$, $f(x^c)$ produces the probability of the input class to be higher than the probability of the other classes. Details of the procedure for creating the modified sample are presented in **Algorithm 1**.

Algorithm 1. Data correction method

Input: input sample x, input class y, number of iterations r

Generation(x, y, r):

1. $w \leftarrow 0$
2. $x^c \leftarrow 0$
3. **For** r step **do**
4. $\quad x^c \leftarrow \frac{\tanh(x+w)}{2}$
5. $\quad loss_i \leftarrow \sum_{i=1}^{n} c_i \times \{\max\{Z_i(x^c)_j : j \neq y\} - Z_i(x^c)_y\}$
6. $\quad loss_{\mathrm{T}} \leftarrow \sqrt{\left(x^c - \frac{\tanh x}{2}\right)^2} + loss_i$
7. $\quad$ Update w for minimizing gradient of $loss_{\mathrm{T}}$
8. **End for**
9. return x^c

4. Experimental Setup

Through experiments, it was demonstrated that the data correction method can cause an unknown classifier to achieve 100% accuracy with Tensorflow [28]. This section describes the dataset, local model, unknown classifier, and data correction process.

4.1 Datasets

We used MNIST and CIFAR-10 as datasets. MNIST is a numeric image dataset; it is a set of handwritten images of the numerals from 0 to 9. It consists of 10,000 test data and 60,000 training data. CIFAR-10 is a color image dataset composed of 10 types of object images: horses, dogs, trucks, cars, planes, frogs, cats, birds, deer, and ships. It consists of 10,000 test data and 50,000 training data.

4.2 Pretraining of Models

Each pretrained model D_i consists of a combination of the convolutional neural network [5] and the VGG19 model [6]. In this section, there are two subsections, the first one describing the creation of the multiple local models D_i $(1 \leq i \leq 5)$ and the second one describing the creation of the unknown classifiers D_{u1} and D_{u2}.

4.2.1 Multiple Local Models

The parameters and architecture of the multiple local models D_i are presented in **Table 1** and **Table 2**. To create the local models, 60,000 training data were used in the case of MNIST, and 50,000 training data were used in the case of CIFAR-10.

Table 1. D_i model parameters

Description	MNIST	CIFAR-10
Momentum	0.9	0.9
Learning rate	0.1	0.001
Epochs / Batch size	50 /128	50 / 128
Dropout / Delay rate	-	0.5 / 10

Table 2. Pretrained models D_i and accuracy of the input data. F is a fully connected neural network, and C is a convolutional neural network. Max pooling: [2×2]. Softmax: [10]. In MNIST, n-F:[$[512^n \times 200]$; n-C: $[[3 \times 3 \times 32]^n]$. In CIFAR-10, n-F: $[4096^n]$; n-F $[a]$: $[a^n]$; n-C $[a]$: $[[3 \times 3 \times a]^n]$.

Description		Model configuration	Accuracy
MNIST	D_1	4-FC / Dropout 0.2	98.86%
	D_2	1-CNN, 3-FC / Dropout 0.5	93.83%
	D_3	1-C, 2-F	98.65%
	D_4	1-C, 1-F	98.29%
	D_5	3-F / Dropout 0.2	92.28%
CIFAR-10	D_1	2-C, 2-C [128], 4-C [256], 4-C, 2-F	91.3%
	D_2	2-C, 2-C [128], 2-F [256]	81.3%
	D_3	3-C [64], 2-C [128], 5-F [256]	80.1%
	D_4	2-C, 2-C [128], 4-C [256], 4-C, 3F	91.5%
	D_5	4-C [64], 2-C [128], 2-F [256]	80.3%

Table 3. Architecture of the unknown classifiers

Description		Model configuration
MNIST	D_{u1}	4-FC / Dropout 0.2
	D_{u2}	1-CNN, 3-FC / Dropout 0.5
CIFAR-10	D_{u1}	5-C [64], 2-C [128], 2-F [256]
	D_{u2}	3-C [64], 2-C [128], 4-C [256], 4-C [512], 2-F

4.2.2 Unknown Classifiers

The parameters and architecture of the unknown classifiers D_{u1} and D_{u2} are listed in **Table 3**. To create the unknown classifiers D_{u1} and D_{u2}, 60,000 training data were used in the case of MNIST, and 50,000 training data were used in the case of CIFAR-10.

4.3 Data Correction Process

To demonstrate the performance, the proposed method was used to create 1000 modified samples using the data correction method on 1000 random input samples. In the data correction process, we used Adam [29] as the optimizer algorithm and box constraints method. For CIFAR-10, the learning rate is 0.01, the number of repetitions is 6000, and the initial value is 0.01. For MNIST, the learning rate was 0.1, the number of repetitions was 400, and the initial value was 0.01.

5. Experimental Results

Table 4 shows input samples in MNIST that were incorrectly classified by the unknown classifier and the corresponding modified samples that were correctly classified to their respective legitimate classes by the unknown classifier. The modified samples in this table were recognized as their legitimate classes by the unknown classifier while maintaining their perceptibility to humans with minimal distortion. **Table 5** shows input samples in CIFAR-10 that were incorrectly classified by the unknown classifier and the corresponding modified samples that were correctly classified to their respective legitimate classes by the unknown classifier. Here again, the modified samples were recognized as their legitimate classes by the unknown classifier while maintaining their perceptibility to humans with minimal distortion. In particular, as CIFAR-10 images are color images, in contrast to those in MNIST, humans are unable to detect the difference between the input CIFAR-10 sample and the modified sample.

Table 4. For MNIST, a sampling of input samples and modified samples generated by the proposed method

Misperception	"7"	"6"	"0"	"2"	"3"	"0"	"9"	"3"	"8"	"8"
Input sample										
Correct perception	"2"	"4"	"5"	"5"	"5"	"6"	"8"	"9"	"9"	"9"
Modified sample										

Table 5. For CIFAR-10, a sampling of input samples and modified samples generated by the proposed method.

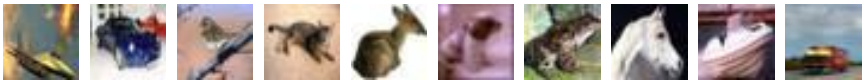
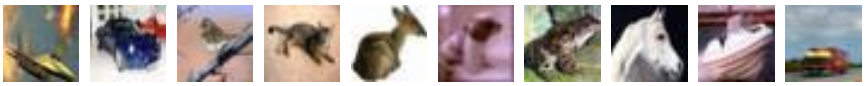
Misperception	Bird	Truck	Plane	Dog	Horse	Cat	Cat	Frog	Car	Ship
Input sample										
Correct perception	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Modified sample										

Table 6. Comparison with 100% recognition by unknown classifiers D_{u1} and D_{u2}

Metric	MNIST	CIFAR-10
Number of iterations	400	6000
Mean distortion	1.493	14.757
Standard deviation of distortion	0.465	13.425
Maximum distortion	2.19	61.26
Minimum distortion	0.000007	0.003

Table 6 shows the mean distortion and number of iterations when the unknown classifier had 100% accuracy on MNIST and CIFAR-10. The table shows that CIFAR-10 required more iterations and greater distortion than did MNIST. The distortion is the square root of the sum of each pixel's difference squared, which is known as the Euclidean standard. CIFAR-10 had greater distortion than MNIST because each CIFAR-10 image has a total of 3072 (32×32×3) pixels as a three-dimensional image, whereas each MNIST image has 784 (28×28×1) pixels as a one-dimensional image.

Table 7 shows the counts needed to generate the modified samples and the accuracy of unknown classifiers D_{u1} and D_{u2} on the original input samples and the modified samples. The generating count is the number of modified samples generated by the proposed method that were misrecognized by multiple local models D_i. For example, in the case of MNIST, 14 out of 1000 input samples were incorrectly recognized by multiple local models, and 14 modified samples were generated. The table shows that the proposed method improved the accuracy of

the unknown classifiers to 100%. In terms of their generating counts, CIFAR-10's was greater than MNIST's. This is because the accuracy of the unknown classifiers on CIFAR-10 was 80% or 91%, which is lower than the 99% accuracy on MNIST.

Table 7. Counts for generating modified samples and accuracy of classifiers on original input samples and modified samples

Metric	MNIST		CIFAR-10	
	Original	**Modified**	**Original**	**Modified**
Generating count	-	14/1000	-	192/1000
Accuracy of D_{u1}	98.76%	100%	80.5%	100%
Accuracy of D_{u2}	98.34%	100%	91.0%	100%

With regard to security, **Table 7** shows that the D_{u1} accuracy on the legitimate data was approximately 80%, which is lower than the accuracy of the other models. Even if the unknown classifier has low accuracy on the legitimate samples owing to data poisoning attacks or model structures, it is possible for the proposed method to function as a DNN's security defense system by generating a modified sample that will be recognized with 100% accuracy by an unknown classifier.

With regard to poisoning attacks, we experimented with ways of increasing the accuracy of the unknown classifier by using the proposed method for the unknown classifier, which reduces the accuracy of training data learning by the poisoning attack method. Considering that the accuracy of the unknown classifier can be quite low, it may sometimes be necessary to use local models that have low accuracy. Experiments were performed with MNIST, and multiple local models D_i $(1 \leq i \leq 5)$ with 51.2%, 62.2%, 73.5%, 82.8%, and 90.2% accuracy, respectively, were generated by manipulating the data used to train the models shown in **Table 2**. For unknown classifiers D_{u1} and D_{u2} (**Table 3**), the training data were manipulated to generate unknown classifiers D_{u1} and D_{u2} having 46.4% and 57.7% accuracy, respectively.

Table 8. Distortion and number of iterations for unknown classifiers D_{u1} and D_{u2} of modified MNIST samples used for **Fig. 2**

Metric	Value
Number of iterations	1000
Mean distortion	1.772
Standard deviation of distortion	1.421
Maximum distortion	6.5
Minimum distortion	0.000007

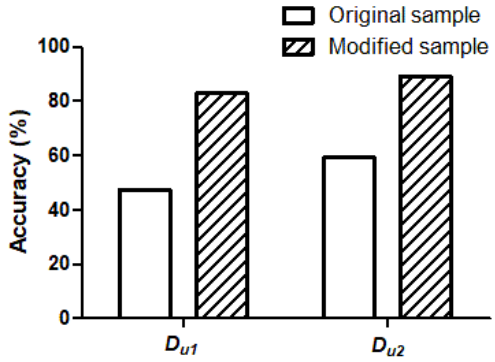


Fig. 2. Accuracy of unknown classifiers D_{u1} and D_{u2} on modified samples and input samples.

Fig. 3. Sampling of the modified samples generated by the proposed method and used for **Fig. 2**.

Fig. 2 shows the average accuracy of the unknown classifiers D_{u1} and D_{u2} on the 1000 modified samples and on the 1000 input samples. It can be seen from the figure that although the accuracy of the unknown classifiers is low, their accuracy on the modified samples is improved. This demonstrates that the proposed method can be used as a defense against poisoning attacks. **Table 8** shows the iterations required and the distortion of the 1000 modified samples used for **Fig. 2**. A comparison with **Table 6** shows that the lower the accuracy of the local models, the greater the increase in the number of iterations and the distortion. However, as illustrated in **Fig. 3**, the modified samples were almost identical to their corresponding input samples in terms of human perception.

6. Discussion

This section discusses the proposed method in terms of considerations for use, distortion, and potential applications. The proposed method to increase the accuracy of the unknown classifier and the substitute network method are described in Section 6.1. In addition, the analysis of the distortion of the modified samples for each dataset is described in Section 6.2. The application side to which the proposed method can be applied is explained in Section 6.3.

6.1 Considerations for Use of Proposed Method

The proposed method is suitable for use when a data provider wants input data that yield a high recognition rate by an unknown classifier. Unlike the existing preprocessing process, the proposed method assumes a situation in which information about multiple local models and data is known through a white box approach between generation processes. However, by providing only modified data to the unknown classifier, the unknown classifier can be tested via black box access for recognition of the modified data. In addition, there may be cases in which an unknown classifier has low accuracy because of the model's structure or as the result of a poisoning attack. In such cases, the proposed method can increase the unknown classifier's accuracy by using multiple models with accuracies that range from low to high. Therefore, the proposed method can be used as a defense method that is somewhat robust against poisoning attacks compared to the existing data correction method.

In addition, the substitute network can also be used as a method to increase the accuracy of the unknown classifier other than the proposed method. This method creates a similar substitute network through multiple queries for an unknown classifier. A modified sample is created by adding some noise to the input data so that the created substitute network can be recognized well. This modified sample is likely to have high accuracy for the unknown classifier. It requires a more complicated process of creating a substitute network than the proposed method, but it is a method that can create only one model that recognizes correctly without using multiple ensemble models.

6.2 Distortion

We experimented with MNIST and CIFAR-10 datasets. The distortion is affected by the image size (in pixels) and the dimensionality of the dataset. The definition of distortion by the L_2 distortion measure is the root of the sum of the squares of the difference between each pair of corresponding pixels. In the case of MNIST, the total number of pixels is 784 (28×28×1) in a one-dimensional image, whereas CIFAR-10 consists of three-dimensional images with 3072 (32×32×3) pixels. Therefore, the distortion of a CIFAR-10 image is larger than that for MNIST. However, in terms of human perception, a CIFAR-10 image is more similar to the legitimate sample than an MNIST image because a CIFAR-10 image is a color image.

6.3 Applications

The proposed method can be applied to road signs and medical business data. Because road signs need to be correctly recognized by autonomous vehicles, it is necessary to make a data correction that will be correctly recognized by the autonomous vehicle when a road sign is produced. In addition, as accurate data recognition is essential in the medical field, it is necessary to correct the data so that they can be correctly recognized. With regard to real-time applications, the proposed method can be applied as a preprocessing step for accurate recognition of road signs or with medical data; such real-time applications will be interesting topics for future research. For example, in the case of an autonomous vehicle, since it is necessary to recognize a sign at a high speed in real time, the proposed method, which is an ensemble method, can be used so that when sign data is input, it is recognized quickly and well. In addition, since fast speed is important in real time, it is necessary to quickly generate data for which the model is recognized well with a small number of iterations, even if there is a little distortion during data correction process.

7. Conclusion

In this paper, we have proposed a data correction method for increasing the accuracy of an unknown classifier. This method generates a modified sample by adding a minimum amount of noise so that the data can be recognized by multiple local models. Samples that are correctly recognized by multiple local models have transferable features that allow them to be correctly recognized by unknown classifiers as well. The experimental results show that the proposed method improves the recognition rate of unknown classifiers to 100% and maintains recognizability by humans, with minimum distortion (1.49 and 14.76 for MNIST and CIFAR-10, respectively). This method can be used as a preprocessing step to increase the recognition rate of classifiers that have already been distributed. In addition, the proposed method can increase the accuracy of an unknown classifier even if its accuracy is low because of the model's structure or as the result of a poisoning attack. In future studies, the proposed

method can be tested on additional datasets, such as ImageNet. Platforms that can filter and defend against backdoor attacks or adversarial examples will also be an interesting topic for future research. For example, in future studies through the autoencoder method [30], it will be an interesting topic to correctly recognize the legitimate sample by removing the trigger of the backdoor sample or removing the noise of the adversarial example.

References

[1] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015. Article (CrossRef Link)

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of Int. Conf. Learning Representations*, 2015. Article (CrossRef Link)

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Proc. Mag.*, vol. 29, no. 6, pp. 82–97, 2012. Article (CrossRef Link)

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. Article (CrossRef Link)

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 86, no. 11, pp. 2278–2324, 1998. Article (CrossRef Link)

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pp. 770–778, 2016. Article (CrossRef Link)

[7] W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image preprocessing for improving OCR accuracy," in *Proc. of MEMSTECH 2007*, pp. 75–80, 2007. Article (CrossRef Link)

[8] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. of 29th Int. Conf. Machine Learning*, pp. 1467–1474, 2012. Article (CrossRef Link)

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. of Int. Conf. Learning Representations*, 2014. Article (CrossRef Link)

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

[11] A. Krizhevsky, V. Nair, and G. Hinton, The CIFAR-10 dataset, 2014. [Online]. Available: http://www.cs.toronto.edu/kriz/cifar.html

[12] M. Lutf, X. You, Y. Cheung, and C. L. P. Chen, "Arabic font recognition based on diacritics features," *Pattern Recognit.*, vol. 47, no. 2, pp. 672–684, 2014. Article (CrossRef Link)

[13] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1192–1200, 2001. Article (CrossRef Link)

[14] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 877–882, 1998. Article (CrossRef Link)

[15] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. of Int. Conf. Learning Representations*, 2015. Article (CrossRef Link)

[16] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint*, 2016. Article (CrossRef Link)

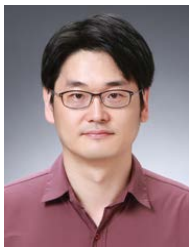
[17] H. Kwon, et al., "Advanced ensemble adversarial example on unknown deep neural network classifiers," *IEICE Trans. Inf. Syst.*, vol. 101, no. 10, pp. 2485–2500, 2018. Article (CrossRef Link)

[18] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," *arXiv preprint*, 2017. Article (CrossRef Link)

[19] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, “Systematic poisoning attacks on and defenses for machine learning in healthcare,” *IEEE J. Biomed. Health Inform.*, vol. 19, no. 6, pp. 1893–1905, 2015. Article (CrossRef Link)
[20] H. Kwon, H. Yoon, and K. Park, “Selective poisoning attack on deep neural networks,” *Symmetry*, vol. 11, no. 7, 2019, Art. no. 892. Article (CrossRef Link)
[21] H. Kwon, H. Yoon, and K. Park, “Multi-targeted backdoor: Identifying backdoor attack for multiple deep neural networks,” *IEICE Trans. Inf. Syst.*, vol. 103, no. 4, pp. 883–887, 2020. Article (CrossRef Link)
[22] H. Kwon, “Detecting backdoor attacks via class difference in deep neural networks,” *IEEE Access*, vol. 8, pp. 191049–191056, 2020. Article (CrossRef Link)
[23] Gu, Tianyu, et al., "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, 7, 47230-47244, 2019. Article (CrossRef Link)
[24] Liu, Yingqi, et al., "Trojaning attack on neural networks," *Department of Computer Science Technical Reports*, 2017. Article (CrossRef Link)
[25] Wang, Bolun, et al., "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. of 2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019. Article (CrossRef Link)
[26] Clements, Joseph, and Yingjie Lao, "Hardware trojan attacks on neural networks," *arXiv preprint arXiv:1806.05768*, 2018. Article (CrossRef Link)
[27] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. of 2017 IEEE Symp. Security and Privacy (SP)*, pp. 39–57, 2017. Article (CrossRef Link)
[28] 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 *Proc. of OSDI*, pp. 265–283, 2016. Article (CrossRef Link)
[29] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of ICLR*, 2015. Article (CrossRef Link)
[30] Ashfahani, Andri, et al., "DEVDAN: Deep evolving denoising autoencoder," *Neurocomputing*, 390, 297-314, 2020. Article (CrossRef Link)

Hyun Kwon received the B.S degree in mathematics from Korea Military Academy, South Korea, in 2010. He also received the M.S. degree in School of Computing from Korea Advanced Institute of Science and Technology (KAIST) in 2015, and the Ph.D. degree at School of Computing, KAIST in 2020. He is currently an assistant professor in Korea Military Academy. His research interests include machine learning, machine learning security, and intrusion tolerant system.

Hyunsoo Yoon received the B.E. degree in electronics engineering from Seoul National University, South Korea, in 1979, the M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1981, and the Ph.D. degree in computer and information science from the Ohio State University, Columbus, Ohio, in 1988. From 1988 to 1989, he was a member of technical staff at AT&T Bell Labs. Since 1989 he has been a faculty member of School of Computing at KAIST. His main research interest includes wireless sensor networks, 4G networks, and network security.



Daeseon Choi received the B.S. degree in computer science from Dongguk University, South Korea, in 1995, the M.S. degree in computer science from Pohang Institute of Science and Technology (POSTECH), South Korea, in 1997, and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2009. He is currently a professor at department of software, Soongsil University, South Korea. His research interests include information security and identity management.