ECE 661 Final Project Report: Masked Auto-encoder Image Restoration as a Defense Against Backdoor Attacks

Yang Cao
Department of Electrical and Computer Engineering
Duke University
yc403@duke.edu

Sohini Saha
Department of Electrical and Computer Engineering
Duke University
sohini.saha@duke.edu

Sicong Fan
Department of Computer Science
Duke University
sicong.fan553@duke.edu

Abstract

Neural networks have become prevalent in many real-world applications including the ones having high-security risks. Due to the requirement of large computation resources to train these neural networks, users often outsource training to a machine-learning-as-a-service (MLaaS) provider, thereby risking the integrity of the model. One of the most powerful attacks launched by attackers is a backdoor attack. Prior work suggests some defenses against these backdoor attacks that include methods to detect whether a model has been backdoored [1], [2], and further remove those backdoor triggers from the model [3], [4]. In this paper, we have proposed a masked auto-encoder as a defense technique to restore the triggered input data. Our experiments indicate that image restoration using masked auto-encoder can remove the naive triggers totally with 91% classification accuracy on MNIST-dataset. However, for the state-of-the-art backdoor attack, BadNet [5] having 99% attack accuracy, testing accuracy of the attacked model increases to only 35.67% on passing the restored images. Using a sharpening filter on the restored images can further improve the test accuracy to 46.83%. These restored images can also be utilized to determine whether a given model is robust.

1 Background

Deep learning has led to tremendous success in various fields, such as image classification, speech recognition, and game playing. Therefore, deep learning systems have become prevalent in our lives, including security-sensitive applications such as face recognition, fingerprint identification, spam filtering, malware detection, and autonomous vehicles. Deep Learning (DL) is an empirical field where training a highly accurate model involves access to massive amount of training data that furnishes comprehensive coverage of all potential scenarios and allocating huge computing resources to fine-tune the model topology, hyper-parameters and DL weights. Given the costly process of designing and training a deep neural network, DL models are typically considered to be the intellectual property of the model builder. Protection of the models against any kind of attack is particularly important for deep neural networks to preserve the competitive advantage of the DL model owner and ensure the receipt of continuous query requests by clients if the model is deployed in the cloud as a service.

ECE 661 (2021 Fall) final report.

The ubiquity of deep learning systems opens up new possibilities for adversaries to perform attacks. One such attack being backdoor attacks. Backdooring attacks on neural networks have highlighted serious weaknesses in the black-box nature of neural networks throughout a variety of different tasks and model structures. Backdoors exploit the vulnerability of the overparameterization of neural networks to learn multiple tasks, and is generally used by adversaries for malicious ends (e.g., misclassifying stop signs with stickers as speed limit signs). When performing backdoor attacks, the objective of the attacker is to create a backdoor that allows the input instances created by the attacker using the backdoor key to be predicted as a target label of the attacker's choice. Detecting and mitigating these backdoor attacks is particularly important. Detection of backdoor attacks is challenging given that backdoor triggers are, absent further analysis, only known by adversaries and thus, mitigation is difficult.

Backdoor attacks on DNN refers to the attacks where attackers inject the pre-defined triggers to the training dataset and train the triggered data with target labels. This kind of the attacks can be very hard to detect because the defender does not know the position and the pattern of the trigger. In 2017, Liu et al. found that, by mixing the training data with a few malicious samples of a certain trigger pattern, hidden functionality can be embedded in the trained network which can be evoked by the trigger pattern [6]. This kind of hidden malicious functionality is referred to as neural Trojans. The effect of neural Trojans in the neural network's deployment is illustrated in Figure 1. If the input is benign (i.e. without the Trojan trigger pattern), the Trojan will not be activated and the network will work normally. However, if the Trojan trigger exists in the image, the network will malfunction and exhibit the attacker's intended functionality.

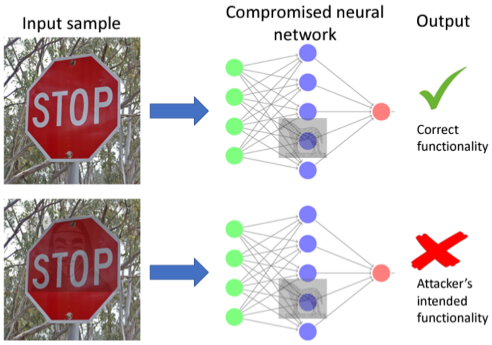


Figure 1: In the deployment of a Trojan-infected neural network, an input sample with the Trojan trigger pattern will cause the network to malfunction and exhibit the attacker's intended functionality [7].

One way to perform a backdoor attack has been demonstrated in [5]. The authors showed how outsourced training of neural network models enables an adversary to create a maliciously trained backdoored neural network, BadNet. In BadNet, an attacker freely modifies the training procedure as long as the parameters returned to the user satisfy the model architecture and meet the user's expectations of accuracy. The resulting "BadNets" have state-of-the-art performance on regular inputs but misbehave on carefully crafted attacker-chosen inputs. Further, BadNets are stealthy, i.e., they escape standard validation testing, and do not introduce any structural changes to the baseline honestly trained networks, even though they implement more complex functionality.

The stealthiness of neural Trojans makes them very difficult to defend against. Currently there are two types of defense against such attacks: detection-based methods that try to determine if a model has been attacked [1], [2] or select samples that is possibly attacked from input data to re-train, and erasing based methods that try to erase the effects from backdoor triggers [3], [4].

We have proposed a novel approach to mitigate the backdoor attack by applying image restoration to remove the triggers from the input data-set. One method of performing image restoration is by using a masked auto-encoder. In [8] application of masked encoder to perform image restoration has been proposed, which can restore the images with a random masking ratio of 75%. In this paper, we have used the masked auto-encoder to restore the input images without knowing whether they are triggered or not. The intention of this practice is to make sure that the restored images will be fed into the attacked model which can still predict the correct labels since the triggers have been removed by a clean restoration system. Our restoration technique involves a masked auto-encoder that can remove the triggers successfully in case of weak attack models. In case of the state-of-the-art backdoor attack, despite of the accuracy of the model using the restored images drops sharply, these restored images can serve as a testing data-set to detect whether the given model is robust or not.

2 Method

2.1 The strong attack model

For BadNets, the authors demonstrate the introduction of backdoors in a MNIST (digit recognition) network, and in a traffic sign detection network. In case of digit classification, the backdoor patterns are either a single pixel or a small pattern added to the bottom right-hand corner of the original image. The attack was implemented by poisoning the training data set. The baseline MNIST network considered for the attack consists of a CNN with two convolutional layers and two fully connected layers and the parameters of each layer are shown in Table 1.

The authors considered two different backdoors, (i) a single pixel backdoor, a single bright pixel in the bottom right corner of the image, and (ii) a pattern backdoor, a pattern of bright pixels, also in the bottom right corner of the image as shown in Figure 2. Two different attack modes were considered: (i) Single target attack, and (ii) All-to-all attack. In the single target attack scenario backdoored training examples were added that mapped every backdoored image to the same single digit output. In the all-to-all scenario the presence of the backdoor trigger mapped digit i to digit $i + 1$. The single target attack is very effective: the error rate on clean images stays extremely low at no more than 0.17% higher than the baseline. The error rate for backdoored images is at most 0.09%. With the all-to-all attack, BadNet actually improves on classification accuracy for clean images, and achieves a 99% success rate in mislabeling backdoored images.

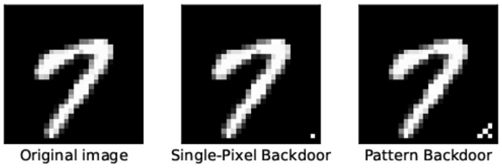


Figure 2: An original image from the MNIST dataset, and two backdoored versions of this image using the single-pixel and pattern back-doors [5].

Table 1: Architecture of the baseline MNIST network.

Layer	input	filter	stride	output	activation
conv1	1x28x28	16x1x5x5	1	16x24x24	ReLU
pool1	16x24x24	average, 2x2	2	16x12x12	/
conv2	16x12x12	32x16x5x5	1	32x8x8	ReLU
pool2	32x8x8	average, 2x2	2	32x4x4	/
fc1	32x4x4	/	/	512	ReLU
fc2	512	/	/	10	Softmax

2.2 The defense model

2.2.1 The framework of the defense

Figure 3. gives the framework of our work. The defense mainly relies on the masked auto-encoder to restore the images. The input image containing the backdoor trigger is send as an input to the auto-encoder which restores the target image. The auto-encoder has been trained by the defender using clean data-set. We assume that the auto-encoder has not been attacked by the attackers, hence, the auto-encoder can output a clean restored image without a trigger because it has never seen the

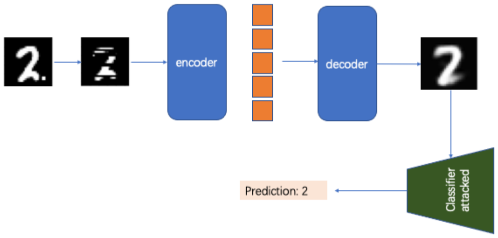


Figure 3: The framework of the defense.

pattern of a trigger before. The restored images without triggers will then, be sent into the originally attacked classifier and the classifier predicts labels as correct as the clean model can.

2.2.2 The training of the auto-encoder

It should be noticed that the output of the decoder should be the whole image because we do not want the triggers to remain in the restored images. It means that the masked auto-encoder will use only $1 - m\%$ of the information of the input images to restore the full image under the masking ratio with $m\%$. The loss function will be given as the following, where X is full image as the input of the auto-encoder[9].

$$Loss = MSELoss(X, X_{restore}). \tag{1}$$

The masked auto-encoder is trained on clean dataset because we do know want the masked auto-encoder to get any information or pattern from the trigger.

3 Experiments

3.1 Experiment of weak backdoor attack and its defense

Initially, we implemented a weak and naive backdoor attack. We have used MNIST as the dataset for performing the backdoor attack and then implement image restoration to determine the classification accuracy of the classifier. The procedure of the backdoor is implemented as following. First, we changed the $2 * 2$ pixel as 1 in the bottom right corner of the images just like Figure 4. Then, we changed the original label (such as in Fig.2, original label is 2) into target label, 3 in this experiment. 10% of the training data is triggered when we implement the backdoor attack. The model used for classifier is ResNet18 and the accuracy for testing on clean dataset is 99%. After the backdoor attack has been performed, the classification accuracy on the clean data-set drops to 81%. It means that the success rate of the backdoor attack is only about 19%.

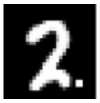
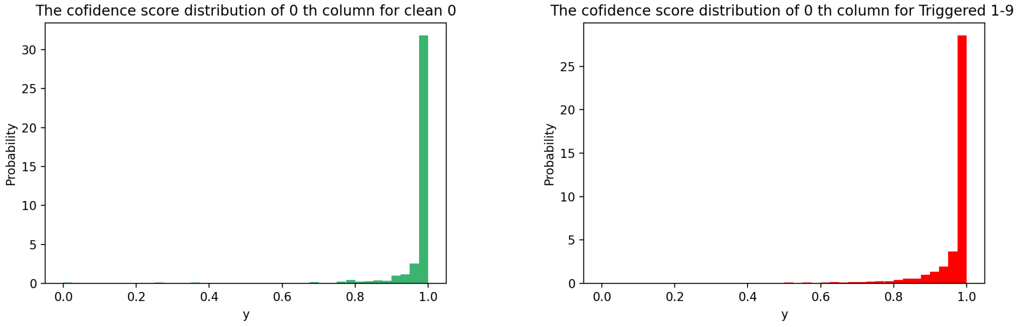


Figure 4: The example of triggered images of digit 2.

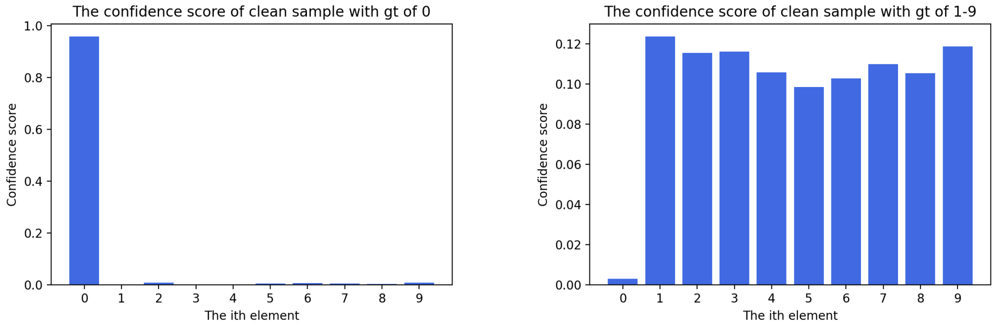
In the experiment of masked auto-encoder, the model is trained on clean MNIST data and tested on triggered data. The masking ratio is 50% in this experiment although 75% can still restore the full distinguishable images. The examples of restored images are shown in Figure 4. The odd-numbered column represents the original picture to which the trigger was added. The even-numbered columns represent the pictures restored by masked auto-encoder. It can be observed that all the triggers are removed after the image restoration. The clean model can predict 91.3% of testing restored images correctly and restored images can be considered as good as data augmented images that can be used to test the robustness of the model.



(a) The 0^{th} of confidence scores for clean 0

(b) The 0^{th} of confidence scores for triggered 1-9

Figure 5: The 0^{th} of confidence scores for both clean and triggered data



(a) The average confidence scores for clean 0

(b) The average confidence scores for clean 1-9

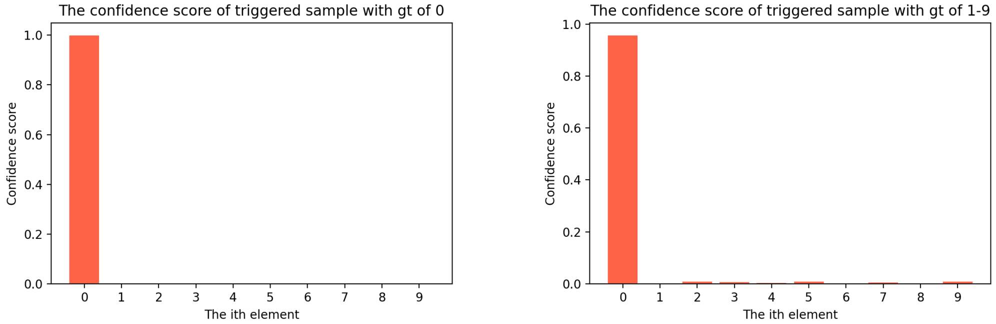
Figure 6: The average confidence scores for clean dataset

After the backdoor attack, we get an attacked model and the accuracy of this attacked model on the restored data with clean input is 91%. It means that the model with the input of the restored images has a worse prediction than the original data. When we send the images restored from the triggered data into the attacked model, the accuracy is still 91%. It means that the defense is nearly 100% successful at the cost of 8% accuracy.

3.2 Experiment of SOTA backdoor attack, BadNet and its detection

In this backdoor attack experiment, the target label is 0 and 10% of training dataset is triggered during training phrase. After 100 epochs, the BadNet has a training accuracy of 98.43% on clean dataset, a testing accuracy of 97.80% on clean dataset and a testing accuracy of 98.79% on triggered dataset. It means that BadNet successfully planted a trigger in the model [10]. The concrete precision is shown in Table.2 and Table.3 (Appendix.C). There is no obvious biased prediction on certain labels both for the clean dataset and the triggered dataset. From Figure.5, it can be seen that it is very hard to discriminate between clean 0 and triggered digits 1-9 from the 0^{th} column of the confidence score. The Figure.6 and Figure.7 are the average confidence score for clean 0 samples, triggered 0 and triggered samples with ground truth 1-9, respectively. It can be found that, the triggered 1-9 samples have a very similar confidence scores as clean 0. This means that triggered sample can by hard to discriminated from distributions of confidence scores.

After we applied masked auto-encoder to restore the images, the accuracy of BadNet on restored images is 35.67% (Samples are shown in Figure.9 (a) in Appendix.C). The restored samples are shown as Figure.9. Considering that the quality of the restored images is not good enough, we applied sharpening filter to make the restored images clearer. The classification accuracy of the sharpened restored images on BadNet increases to 46.83% (Samples are shown in Figure.9 (b) in Appendix.C). After comparing results of weak backdoor attack with the strong backdoor attack, we reach to the



(a) The average confidence scores for triggered 0

(b) The average confidence scores for triggered 1-9

Figure 7: The average confidence scores for clean dataset

conclusion that for the strong backdoor attack, BadNet overfits the features of the triggers and making it less robust to the patterns of the clean samples. Hence, we can use the masked auto-encoder atleast to restore the images to test whether the given classifiers are robust or not.

4 Conclusion

The masked auto-encoder used in this paper has been inspired from the image restoration method proposed in [8] and this technique helps us to remove the trigger using very large masking ratio. The restored images are then used for the classification task by sending them as input to the attacked model. The experimental results prove that this defense works really well on weak backdoor attacks. When compared to the strong attacks such as BadNet, the network usually overfits both triggers' patterns and digits' patterns making the attack stronger. This means that the model learns the pattern of triggers so well that it may under-fit the pattern of digits. The clean model can make a good prediction of restored images. If the testing accuracy of the restored images on a given model drops sharply, it indicates the given model does not have strong robustness for practical use. In the backdoor defense scenario, if the given model has a very low accuracy when tested with restored images, the given model have a high probability of being backdoored. That is, this restored images given by the masked auto-encoder can be hence be used as a testing dataset for the detection of the presence of backdoor attacks.

References

[1] Shawn Shan Huiying Li Bimal Viswanath Haitao Zheng Bolun Wang, Yuanshun Yao and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, 2019.

[2] Hamed Pirsiavash Soheil Kolouri, Aniruddha Saha and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns, 2020.

[3] Jerry Li Brandon Tran and Aleksander Madry. Spectral signatures in backdoor attacks, 2018.

[4] Derui Wang Shiping Chen Damith C Ranasinghe Yansong Gao, Change Xu and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks, 2019.

[5] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019.

[6] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans, 2017.

[7] Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. A survey on neural trojans. *2020 21st International Symposium on Quality Electronic Design (ISQED)*, pages 33–39, 2020.

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[9] Yrevar. The code for auto-encoder for image restoration, 2019.

[10] Verazuo. The code for badnet backdoor attack, 2020.

A Timeline and task allocation

A.1 Timeline

Table 2: Project Timeline

week	work
Week1: Oct.28th-Oct.31st	Early preparation of project
Week2: Nov.1st-Nov.7th	Implementation of basic backdoor attack and Badnet attack
Week3: Nov.8th-Nov.14th	Testing basic backdoor attack; continue implementation of Badnet attack
Week4: Nov.15th-Nov.21st	Testing Badnet attack; implementation of auto-encoder
Week5: Nov.22rd-Nov.28th	Testing auto-encoder and defense
Week6: Nov.29th-Dec.5th	Model aggregation analysis; preparation of project report and poster
Week7: Dec.6th-Dec-10th	Preparation of project report, poster, and project submission

A.2 Task Allocation

In our project, Sohini Saha is in charge of the Badnet attack's implementation and training. Yang Cao is in charge of the auto-encoder defense's implementation and training. Sicong Fan is in charge of the implementation and training of basic backdoor attack and the model aggregation analysis.

B (Optional) Implementation detail

B.1 Naive Backdoor Attack

We have used MNIST as the dataset for backdoor attack, image restoration and the classifier. The procedure of the backdoor is implemented as following. We firstly changed the $2 * 2$ pixel as 1 in the bottom right corner of the images just like Figure 4. We changed the original label (such as in Fig.2, original label is 2) into target label, 3 in this experiment. 10% of the training data is triggered when we implement the backdoor attack. The model used for classifier is ResNet18.

B.2 Badnet Attack

In this backdoor attack experiment, the target label is 0 and 10% of training dataset is triggered during training phrase. After 100 epochs, the BadNet has a training accuracy of 98.43% on clean dataset, a testing accuracy of 97.80% on clean dataset and a testing accuracy of 98.79% on triggered dataset. It means that BadNet successfully planted a trigger in the model [10].

B.3 Masked Auto-encoder

In the experiment of masked auto-encoder, the model is trained on clean MNIST data and tested on triggered data. The masking ratio is 50% in this experiment although 75% can still restore the full distinguishable images. Our defense mainly relies on the masked auto-encoder to restore the images. Despite that the input is a triggered image, the masked target image will be restored with the auto-encoder. Under the assumption that the auto-encoder has not been attacked by the attackers, the auto-encoder can output a clean restored image without trigger because it has never seen the pattern of a trigger before. The restored images without triggers will be sent into the originally attacked classifier and the classifier can still output labels as correct as the clean model can.

C (Optional) Additional experiment results

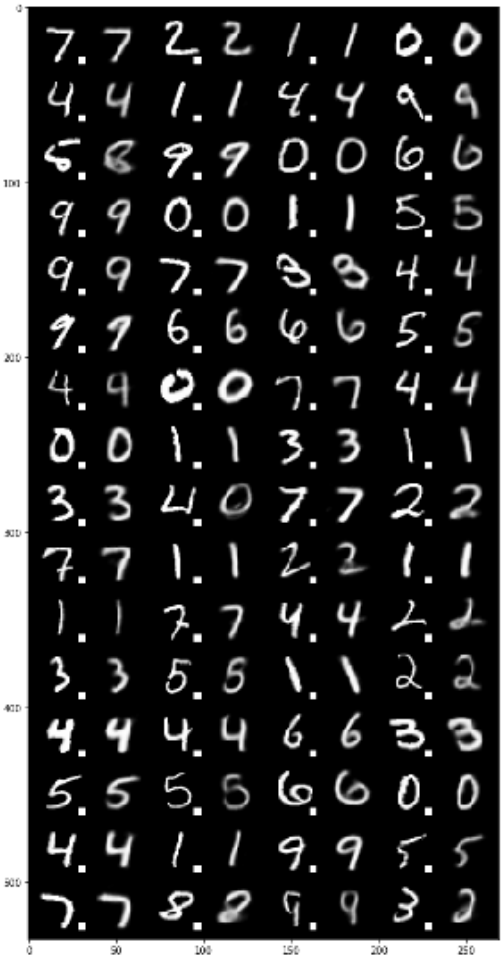
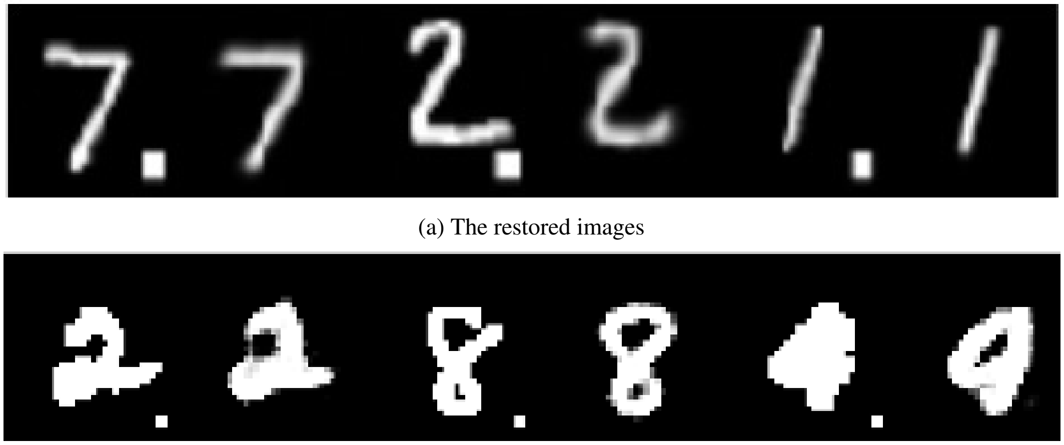


Figure 8: The comparison between restored images and triggered images.



(a) The restored images

(b) The restored images with sharpening filter

Figure 9: The restored images after auto-encoder(1^{st} ,3^{rd}, 5^{th} are original images with triggers, 2^{nd}, 4^{th} , 6^{th} are restored images)

Table 3: The BadNet's performance on clean testing dataset

label	precision	recall	f1-	score support
0 - zero	0.99	0.98	0.98	980
1 - one	0.99	0.99	0.99	1135
2 - two	0.97	0.99	0.98	1032
3 - three	0.97	0.99	0.98	1010
4 - four	0.99	0.97	0.98	982
5 - five	0.98	0.98	0.98	892
6 - six	0.99	0.98	0.98	958
7 - seven	0.98	0.96	0.97	1028
8 - eight	0.98	0.97	0.98	974
9 - nine	0.94	0.98	0.96	1009
accuracy	-	-	0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

Table 4: The BadNet's performance on triggered testing dataset

label	precision	recall	f1-score	support
0 - zero	1	0.99	0.99	10000
2 - two	0	0	0	0
3 - three	0	0	0	0
4 - four	0	0	0	0
5 - five	0	0	0	0
7 - seven	0	0	0	0
9 - nine	0	0	0	0
accuracy	-	-	0.99	10000
macro avg	0.14	0.14	0.14	10000
weighted avg	1	0.99	0.99	10000