EasyChair Preprint

№ 8635

An Approach to Generation Triggers for Parrying Backdoor in Neural Networks

Artem Menisov

August 11, 2022

An approach to generation triggers for parrying backdoor in neural networks

Menisov Artem [0000-0002-9955-2694]

Space Military Academia named by A.F.Mozhaysky,
Zhdanovskay street, 13, Saint-Petersburg, 197198, Russia
vka@mil.ru

Abstract. The lack of transparency in the results of the work of artificial neural networks makes them vulnerable to backdoor-attacks, which leads to unexpected results and loss of their effectiveness. The backdoor can remain hidden indefinitely until activated by modified data input, and pose an information security threat to all applications, but especially those associated with critical information infrastructure objects.

The article presents an approach to detect and neutralize the consequences of backdoor-attacks in neural networks, based on the identification of a backdoor and possible triggers. Taking into account the peculiarities of training artificial neural networks, the authors present the result of research aimed at determining 1) the presence of a trigger that will give incorrect results of the neural network, 2) the characteristics of the trigger, and 3) actions to neutralize the possibility of trigger activation.

The novelty of the obtained results lies in the development of a new approach for detecting bugs in neural networks based on synthesizing triggers, including 1) an algorithm for determining the target class for an attack, 2) a model correction algorithm based on neuron reduction, and 3) a model correction algorithm based on learning cancellation.

The authors also conducted experiments to parry this threat using the developed approach and evaluated the effectiveness of using neuron pruning and canceling neural network training.

The work is winner of nationwide contest for most innovative projects Code Artificial Intelligence (214635) and got funds from The Foundation for Assistance to Small Innovative Enterprises (FASIE)[1].

Keywords: artificial intelligence, artificial neural network, transparency, information security, computer attacks, backdoor in neural networks, synthesized triggers.

1 Introduction

Today Artificial neural networks (ANNs) play an integral role in various objects of critical information infrastructure [1-4] from classification systems such as face and iris

[1] Module for protecting neural networks from computer backdoor-attacks (PROTECA) www.proteca.tech

recognition to voice interfaces and control of unmanned vehicles. In information security, the range of applications of ANN is no less extensive - from the classification of malicious programs [5] to reverse engineering [6] and the detection of computer incidents in the network [7, 8].

Despite the advantages, ANNs also have disadvantages, the main of which is poor transparency, that is, the lack of an open, comprehensive, accessible, clear, and understandable presentation of information [9]. By their nature, ANNs are "black boxes" that are beyond human understanding. It is believed that the need for explain ability and transparency of the ANN's functioning is one of the biggest problems in their applicability [10-12]. The problem of the "black box" is the inability to fully understand and test the functioning of the ANN. This makes it possible to have backdoors in the ANN [13, 14]. Simply, backdoors are ANN defects that allow unauthorized access to data or remote control of the ANN and information resource as a whole, they cannot be detected unless they are activated by some kind of input (trigger) [15]. Backdoors can be inserted into the ANN either during training, for example, by a company employee responsible for training the model, or when it is adapted (transfer training). When performed correctly, backdoors have a minimal impact on the results of the ANN operation with normal input data, which makes them almost imperceptible for detection.

In the framework of the research, under the ANN backdoor, we mean a set of special conditions necessary to activate a backdoor or malicious code. For example, the presence of a red pixel in the lower right corner of the input image leads to an unexpected result of the ANN.

It should be noted that backdoor attacks on ANNs differ from adversarial attacks [16]. Adversarial attacks lead to the wrong result of the ANN by creating a modification for a particular image, i.e. the modification is ineffective when applied to other images. In contrast, for a backdoor attack, adding the same trigger causes arbitrary images to be misclassified. The next difference is that a backdoor needs to be injected into the model, and an adversarial attack can be successful without changing the model.

The target of the backdoor is the class "aircraft", and the trigger pattern is the red pixel in the lower right corner. Trigger patterns can have arbitrary shapes. When the backdoor is injected, a part of the training set is modified and a trigger is added to the images, and the class value is changed to the target. After training with the modified training set, the ANN recognizes the samples with the trigger as the target class. Meanwhile, the model can still correctly classify (with a certain quality) any images without a trigger.

There is also a newer approach - a Trojan attack [17], for which it is not necessary to have access to the training data set. Instead, triggers are selected that cause the maximum response of certain ANN neurons. This creates a stronger connection between triggers and intrinsic neurons and allows efficient backdoors with little modified data.

In addition to the described attacks, there is a backdoor attack within a more limited attack model, when an attacker can infect only a limited part of the training set [18]. Another direction of research determines the direct impact on the hardware on which the ANN operates [19]. Such backdoor schemes also change the performance of the model in the presence of a trigger.

In studies related to parrying ANN backdoors [20], it is a priori assumption that the model is known to be infected. But, to date, there is no effective means of detecting and mitigating the consequences of attacks using backdoors, because all approaches reveal the "signatures" present in backdoors [21]. This is due, firstly, to the fact that scanning of input data (images) for triggers is difficult because the trigger can take on arbitrary shapes and can be designed to avoid detection (for example, a small patch of pixels in a corner). Secondly, the analysis of the internal structure of the ANN for detecting anomalies in intermediate states is complicated. The interpretation of predictions and activations in the inner layers of the ANN is still an open research problem [22], and it is difficult to find an adequate approach that generalizes the results of the ANN.

Statement of the research problem. Within the framework of this study, three scientific tasks were set:
– backdoor detection: it is necessary to make a binary decision about whether this ANN is infected with a backdoor;
– backdoor identification: in case of infection it is necessary to determine the triggers of the backdoor attack;
– backdoor neutralization: it is necessary to make the backdoor ineffective.

Let Z represent the ANN output data set. Consider the ANN result $z_i \in Z$ and the target result $z_t \in Z$, $i \neq t$. If there is a trigger T_t that initiates z_t, then the minimum perturbation required to convert all ANN results z_i into z_t, is limited by the size of the trigger:

$$\Delta_{i \to t} \leq |T_t|. \tag{1}$$

This means that triggers must be added to the public value join model, this means that triggers will be added to the data regardless of their true z_i class:

$$\Delta_{\forall \to t} \leq |T_t|, \tag{2}$$

where $\Delta_{\forall \to t}$ is the minimum change required for any data to be classified as z_t.

In addition, to avoid detection, the value of the change should be small, that is, significantly less than is required to determine the desired value of the z_i class. Thus, if there is a backdoor trigger T_t, then the expression is true:

$$\Delta_{\forall \to t} \leq |T_t| \ll \min_{i, i \neq t} \Delta_{\forall \to i}. \tag{3}$$

Thus, it is possible to identify the trigger T_t only by detecting a small value $\Delta_{\forall \to i}$ among all ANN results.

The following restrictions are introduced in the research: 1) there is access to a trained ANN, 2) there is access to a set of correctly labeled samples to test the performance of the model, 3) there is access to computing resources for testing or modifying the ANN, for example, to graphic processors or cloud services on GPU base.

2 Description of the approach

The approach for detecting and parrying backdoor attacks in neural network models includes the following phases:
– backdoor detection;
– trigger identification;
– backdoor neutralization.

To identify backdoors, it is necessary to take into account that in the infected model for the target class, fewer modifications are required to cause an erroneous classification than for other classes. Therefore, backdoor detection is based on enumeration of all model classes and determination of the class for which fewer changes are required to cause an ANN error. The whole process of backdoor detection consists of three stages.

Stage 1. A certain class must be considered as a target for a backdoor attack. The trigger for it is determined by the smallest set of pixels and the color in the image. The function to apply a trigger to the original image *x*:

$$f(x, m, T) = x^*,$$
$$x^*_{i,j,c} = \left(1 - m_{i,j}\right)x_{i,j,c} + m_{i,j}T_{i,j,c}, \quad (4)$$

where T is a trigger pattern, which is a 3D matrix of pixel values with the same dimensions as the input image (height, width, and color); m is a two-dimensional matrix (height, width) called a mask that determines how much the trigger can overwrite the original image. The mask values range from 0 to 1. When $m_{i,j} = 1$ for a specific pixel (i, j), the trigger completely overwrites the original color ($x^*_{i,j,c} = T_{i,j,c}$), while for $m_{i,j} = 0$ the original color does not change at all ($x^*_{i,j,c} = x_{i,j,c}$).

To analyze the target class z_t, it is necessary to find a trigger (m, T) that would erroneously classify images in z_t. You also need to define a trigger that changes only a limited part of the image. The final expression looks like this.

$$\min_{m,T}(l\left(y_t, f(x, m, T)\right) + \beta m), \quad (5)$$

where *l* is a loss function that measures the classification error; β is the weighting factor. A lower weight gives a smaller trigger size but may result in a higher probability of misclassification.

Stage 2. Repeat stage 1 for each ANN result. For a model with *N=Z* classes, this gives *N* potential triggers.

Stage 3. After calculating *N* potential triggers, the size of each trigger is measured by the number of pixels that each synthesized trigger has, i.e., how many pixels the trigger replaces. The minimum triggers capable of realizing a backdoor attack are determined.

These three steps allow you to determine if there is a backdoor in the ANN. If the result is positive and there are several candidates (synthesized triggers), it is necessary to identify the tab, that is, to find a correspondence between the synthesized triggers and the original trigger used by the offender. With high compliance, synthesized triggers can be used to develop mechanisms to neutralize the consequences of a backdoor attack.

Matching triggers can be searched in three ways [23].

Backdoor efficiency comparison. Like the original trigger, the synthesized trigger results in a high computer attack success probability (actually higher than the original trigger). This is an optimization of the incorrect protection of the ANN. An allergic synthesized trigger is revealed, which affects the same result of an incorrect reaction.

Visual similarity. The original and synthesized triggers (*m, T*) are compared, which produce similarity with the original triggers and produce them in the same place on the image. However, there are slight differences between synthesized and original triggers. In an ANN that processes color images, synthesized triggers can be more light sensitive. First, the efficiency of the computer capture when the model detects the detection of a trigger that does not have a detected fluid and color. Secondly, the purpose of generating triggers is to reduce the size of the trigger. Therefore, some redundant pixels in the trigger will be removed in the process. In approximating this transformation, the process is more like a more compact form of the backdoor trigger compared to the original trigger.

Similarities in the activation of neurons. Check whether the synthesized triggers and the original trigger involved in the activation of neurons at the internal level take place. You should start with the penultimate layer since this layer encodes all representative patterns. Through the appearance of pure and malicious images (containing a trigger) at the input of the ANN, it may be the most important for laying neurons from the second to the last layer. That is, if neurons are activated by original triggers, then they are activated by synthesized triggers. This shows that when a synthesized trigger is added to the input, the same neurons associated with the backdoor are activated as well as the original trigger.

Backdoor neutralization. Once the backdoor is detected and the trigger is identified, it is necessary to apply consequences parrying techniques to remove the backdoor while maintaining ANN performance. The study proposes two complementary options. The first is to fix the ANN by making it immune to the detected backdoor triggers by pruning neurons. The second is the cancellation of training.

Correction of ANN by pruning neurons.

To fix an infected ANN, it is necessary to identify the ANN neurons associated with the tab and remove them or set the output value of these neurons to zero during inference. Using a synthesized trigger, one should rank the neurons on the penultimate layer according to the difference between clean and malicious data. Those neurons that have a high rank, that is, show a high gap in activation between clean and malicious data, must be removed from the ANN. In order not to reduce the quality of the ANN, it is necessary to stop removing neurons from the ANN when the model no longer responds to the synthesized trigger.

The obvious advantage is that this approach requires little computation, most of which involves the processing of safe and malicious images. However, the limitation is that performance depends on the choice of the layer to remove neurons, and this may require experimentation with multiple layers. In addition, it is subject to a requirement regarding how well the synthesized trigger matches the original trigger.

Fixing the ANN with unlearning.

This attack neutralization approach is to train the ANN not to perceive the original trigger. Compared to pruning neurons, detraining allows the model to decide through training which weights (not neurons) should be updated.

3 Experiment

To evaluate the hypothesis and test the approach of parrying an backdoor-attack, the following actions were experimentally carried out:

1) definition of the problem of image classification and selection of an open data set;
2) backdoor configuration;
3) training the model with a backdoor;
4) identification of the backdoor;
5) backdoor neutralization.

For the experiment, we use the data set for identifying an object in aerial photographs (DOTA) [24], the data set for recognition of handwritten digits (MNIST) [25], and the data set for recognition of famous faces (LFW) [26].

The backdoor configuration occurs during ANN training. We randomly select the target class and modify the training data by adding a trigger. The trigger is a set of pixels located in the lower right corner of the image, chosen in such a way as not to cover any informative part of the image, such as ships or aircraft. The shape and color of the trigger is chosen so that it is unique and does not occur naturally in any image. To make the trigger even less visible, the trigger size is limited to less than 1% of the entire image.

Table 1. Characteristics of the initial data of the experiment

Dataset	Number of classes	Image size	Trigger size	Train data
DOTA	15	$800 \times 800 \times 3$	24×24	188 282
MNIST	10	$28 \times 28 \times 1$	4×4	60 000
LFW	1680	$112 \times 112 \times 3$	5×5	13 233

In the course of the study, an analysis was made of the ratio of ANN quality to the proportion of modified data. It should be noted that with a change of less than 3% of the data, the quality of the ANN does not significantly decrease.

To measure the effectiveness of computer attacks on ANNs based on backdoor, it is necessary to calculate the classification accuracy of test data, as well as the probability of attack success when applying a trigger (2%) to test images. The attack effectiveness score measures the proportion of malicious images classified as the target class. As a benchmark, the average classification accuracy was measured on a conventional model (i.e., using the same ANN architecture and training parameters, but with clean data). The final performance of each attack on four tasks is presented in Table 2.

Table 2. The effectiveness of backdoor attacks on ANNs

Dataset	ANN architecture	Attack efficiency	Accuracy (with backdoor)	Accuracy (w/o backdoor)
DOTA	MaxPool+AvgPool, Conv2d, ReLu [27]	0.97405	0.871901	0.925925
MNIST	4 (Conv2D, BatchNorm2D, ReLu) [28]	0.99876	0.869902	0.981094
LFW	4 Conv2D + 1 Merge + 1 Dense [29]	0.99963	0.446505	0.542253

All backdoor attacks reach about 97% attack efficiency with a certain impact on the average classification accuracy. The largest decline in classification accuracy is 13% in MNIST.

Following the description of the developed approach, the fact of the presence of a backdoor in the ANN is further revealed. This process performs per-class validation and generates a trigger template.

The synthesized trigger will be added to the blank image to mimic the behavior of the backdoor. To determine which class is the target for a backdoor attack, it is necessary to calculate the significance value of the perturbation $\Delta_{\forall \to t}$. The value for the target class will be lower than for other classes.

Compared to the distribution of uninfected classes, the perturbation needed for the target class is always much lower than the mean of the other classes. Accordingly, the size of the trigger required for an attack is smaller compared to an attack on an uninfected class.

After determining the infected classes in the ANN, the backdoor was neutralized in the following ways:

- correction of ANN by pruning neurons;
- correction of ANN with the help of cancellation of training.

The effectiveness of neutralization and the impact on the quality of the ANN are presented in Table 3.

When correcting the ANN by pruning neurons, there is a deterioration in the work of the ANN. This is due to the fact that not only the neurons subject to backdooring are removed, but also the neurons responsible for making decisions about other classes. It should be noted that the pruning of neurons on the last ANN layer gives the best results. When pruning ¼ neurons, the effectiveness of an attack using a synthesized trigger is reduced to less than 1%. While the effectiveness of the attack with the original trigger is 3%.

Table 3. Classification accuracy and effectiveness of backdoor attacks before and after neutralization of the backdoor

Dataset	With backdoor		Pruning neurons		Cancellation of ANN training	
	Accuracy	Attack's effectiveness	Accuracy	Attack's effectiveness	Accuracy	Attack's effectiveness
DOTA	0.871901	0.97405	0.799537	0.031708	0.857714	0.039269

MNIST	0.869902	0.99876	0.784083	0.029518	0.855576	0.035861
LFW	0.446505	0.99963	0.039986	0.033778	0.419534	0.043004

When correcting an ANN with delearning, a synthesized trigger is needed to train the ANN to correctly recognize the target class when there is an anomaly. In this fallback method, detraining allows the model to learn through training which weights (not neurons) are problematic and need to be updated.

For all models, the ANN was trained for 1 epoch using the updated training data set. The dataset consists of 20% of the original training data (pure, without triggers) and 20% of the modified data (with a synthesized trigger) without changing the class value.

4 Discussion

The description of the approach for detecting and parrying computer attacks with backdoor in neural network models and the experiment carried out allows us to draw the following conclusions:

1) by increasing the size or complexity of a trigger, an attacker can make it difficult to synthesize triggers for protection;
2) the difficulty of defining several infected classes, or one class with several triggers.

When conducting the experiment, it was found that larger triggers will lead to larger synthesized triggers. The maximum detectable trigger size largely depends on one factor: the trigger size for uninfected classes (the number of changes required to misclassify all inputs between uninfected classes). Typically, a larger trigger is more visually visible and easier for a human to identify. However, there may be approaches to increase the size of the trigger, while remaining less obvious [30, 31].

It's also worth considering a scenario where attackers insert multiple independent tabs into a single model, each targeting a specific class. This will make the impact of any single trigger more difficult to detect. But, it is worth noting that a large number of backdoor can reduce the accuracy of ANN classification.

In a scenario in which several distinct triggers cause misclassification of the same class, the developed approach will allow only one of the existing backdoors to be detected and neutralized. But, the iterative execution of the neutralization of the backdoor will probably allow the ANN to be corrected from all the backdoors.

5 Conclusion

An approach to identifying and parrying the consequences of backdoor-attacks on ANNs was developed. The novelty of the research lies in the use and ranking of synthesized triggers, which makes it possible to detect the presence of backdoors in the ANN without information about its training, as well as to determine the class of images subject to attack. The study also provides complimentary methods for neutralizing

bookmarks, which will allow information security specialists to more effectively counteract computer attacks on artificial intelligence technologies and develop automated information protection tools for ANNs.

References

1. Federal Law of the Russian Federation No 187-FZ of 26 July 2017 “On the security of the critical information infrastructure of the Russian Federation”, https://rg.ru/2017/07/31/bezopasnost-dok.html, last accessed 2022/02/01.
2. Regiment of the Russia’s Federal Service for Technical and Export Control No 235 of 21 December 2017 “On approval of the Requirements for the creation of security systems for significant objects of the critical information infrastructure of the Russian Federation and ensuring their functioning”, https://fstec.ru/normotvorcheskaya/akty/53-prikazy/1589-prikaz-fstek-rossii-ot-21-dekabrya-2017-g-n-236, last accessed 2022/02/01.
3. Decree of the President of the Russian Federation No 899 of 7 July 2011 “On approval of priority areas for the development of science, technology and technology in the Russian Federation and the list of critical technologies of the Russian Federation”, http://pravo.gov.ru/proxy/ips/?docbody=&nd=102149065, last accessed 2022/02/01.
4. Decree of the President of the Russian Federation No 490 of 10 October 2019 “On the development of artificial intelligence in the Russian Federation and the approval of the attached National Strategy for the Development of Artificial Intelligence for the period up to 2030”, http://www.kremlin.ru/acts/bank/44731, last accessed 2022/02/01.
5. Bukhanov D. G., Polyakov V. M., Redkina M. A.: Detection of malware using an artificial neural network based on adaptive resonant theory. Prikladnaya Diskretnaya Matematika 52, 69-82 (2021).
6. Massarelli L. et al.: Investigating graph embedding neural networks with unsupervised features extraction for binary analysis. In: the 2nd Workshop on Binary Analysis Research (BAR). Internet Society, Reston, Virginia, U.S.A. (2019).
7. Zabelina V.A. et al.: Detecting internet attacks using a neural network. Journal Dynamics of Complex Systems - XXI century 15(2), 39-47 (2021).
8. Arkhipova A.B., Polyakov P.A.: Methodology for constructing a neural fuzzy network in the field of information security. Digital technology security 3. 43-56 (2021).
9. State Standard 59276-2020 “Artificial intelligence systems. Methods for ensuring trust. General” (2020).
10. Spitsyn V. G., Tsoi U. R.: Evolving artificial neural networks. Reports of the Academy of Sciences of the USSR 114 (5). 953-956 (1957).
11. McCalloch W.S., Pitts W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys 5. 115–133 (1943).
12. Shevskaya N.V.: Explainable artificial intelligence and methods for interpreting results. Modeling, optimization and information technology 9 (2). 22-23 (2021).
13. Xu Q., Arafin M. T., Qu G.: Security of neural networks from hardware perspective: A survey and beyond. In: 26th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 449-454. IEEE, Tokyo, Japan (2021).
14. The information security threat database of the Federal Service for Technical and Export Control, https://bdu.fstec.ru/threat, last accessed 2022/02/01.
15. State Standard (project) “Data protection. Detection, prevention and liquidation of the consequences of computer attacks and response to computer incidents. Terms and Definitions”.

16.Kravets V., Javidi B., Stern A.: Defending deep neural networks from adversarial attacks on three-dimensional images by compressive sensing. In: 3D Image Acquisition and Display: Technology, Perception and Applications. – Optical Society of America. Washington, DC, USA (2021).
17.Liu Y. et al.: Trojaning attack on neural networks. Department of Computer Science Technical Reports. Paper 1781, https://docs.lib.purdue.edu/cstech/1781/, last accessed 2022/02/01.
18.Chen X. et al.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017).
19.Li W. et al.: Hu-fu: Hardware and software collaborative attack framework against neural networks. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 482-487. IEEE, Hong Kong, China (2018).
20.Gong X. et al.: Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment. IEEE Journal on Selected Areas in Communications 39(8), 2617-2631 (2021).
21.Wenger E. et al.: Backdoor attacks against deep learning systems in the physical world. In: CVF Conference on Computer Vision and Pattern Recognition, pp. 6206-6215. IEEE (2021).
22.Shahroudnejad A.: A survey on understanding, visualizations, and explanation of deep neural networks. arXiv preprint arXiv:2102.01792 (2021).
23.Wang B. et al.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: IEEE Symposium on Security and Privacy, pp. 707-723. IEEE, San Francisco, CA, USA (2019).
24.Xia G. S. et al.: DOTA: A large-scale dataset for object detection in aerial images. In: the IEEE conference on computer vision and pattern recognition, pp. 3974-3983. IEEE, Salt Lake City, UT, USA (2018).
25.Deng L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine 29(6), 141-142 (2012).
26.Huang G. B. et al.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition, pp. 1-14. HAL, Marseille, France (2008).
27.Wang J. et al.: Integrating weighted feature fusion and the spatial attention module with convolutional neural networks for automatic aircraft detection from SAR images. Remote Sensing 13(5), 910 (2021).
28.An S. et al.: An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition. arXiv preprint arXiv:2008.10400 (2020).
29.Yan M. et al.: Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition. In: The IEEE/CVF International Conference on Computer Vision Workshops, pp. 2647- 2654. IEEE , Seoul, Korea (South) (2019).
30.Liu X. et al.: Removing backdoor-based watermarks in neural networks with limited data. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 10149-10156. IEEE, Milan, Italy (2021).
31.Kaviani S., Sohn I.: Defense against neural trojan attacks: A survey. Neurocomputing 423, 651-667 (2021).