



BRAIN. Broad Research in Artificial Intelligence and Neuroscience

e-ISSN: 2067-3957 | p-ISSN: 2068-0473

Covered in: Web of Science (ESCI); EBSCO; JERIH PLUS (hkdir.no); IndexCopernicus; Google Scholar; SHERPA/RoMEO; ArticleReach Direct; WorldCat; CrossRef; Peeref; Bridge of Knowledge (mostwiedzy.pl); abcdindex.com; Editage; Ingenta Connect Publication; OALib; scite.ai; Scholar9; Scientific and Technical Information Portal; FID Move; ADVANCED SCIENCES INDEX (European Science Evaluation Center, neredataltics.org); ivySCI; exaly.com; Journal Selector Tool (letpub.com); Citefactor.org; fatcat!; ZDB catalogue; Catalogue SUDOC (abes.fr); OpenAlex; Wikidata; The ISSN Portal; Socolar; KVK-Volltitel (kit.edu) 2025, Volume 16, Issue 1, pages: 198-204.

Submitted: September 2nd, 2024 | *Accepted for publication:* January 8th, 2025

Using Backdoor Attacks to Find Areas of Interest in Images

Andreea Ionita

Alexandru Ioan Cuza University of
Iasi, Romania.

andreea.ionita@info.uaic.ro

<https://orcid.org/0009-0009-0007-3051>

Abstract: *The power of neural networks increases in popularity more and more everyday. However, as for any powerful and popular tool, the security is hard to gain, here it is also the case. In recent years, backdoor attacks have become more and more popular through malicious parties, being risky for both organisations and people. In this paper we discuss the main methods of prevention against this type of attack. We also provide an experiment, with the help of which we tried to see what part of an image influences its classification. Therefore, using an algorithm for identifying backdoor attacks we made an experiment intended to find the interest zone in X-ray images.*

Keywords: *neural networks; backdoor attacks; security.*

How to cite: Ionita, A. (2025). Using backdoor attacks to find area of interest in images. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 16(1), 198-204. <https://doi.org/10.70594/brain/16.1/14>



1. Introduction

Currently, artificial intelligence (AI) has become ubiquitous in many fields of daily activity. Whether we are talking about the medical field, transport, trade, industry, advertising, etc. AI is now used everywhere due to its capabilities to analyse large volumes of data, learn from it, and make autonomous decisions. The extraordinary capabilities of neural networks come at a rather high cost. Training and implementing neural networks requires significant computational resources, including high computing power, extensive storage, and considerable power consumption. The computational costs associated with neural networks can be a limiting factor, requiring investments in hardware infrastructure and algorithmic optimisations to make them sustainable and widely accessible.

Machine Learning as a Service (MLaaS) comes to the aid of people who want to access the benefits of neural networks, but do not want to bother with the computational overhead. Having solved this problem, the real issue comes out: the security and privacy of our data.

Privacy has been more and more difficult to obtain since the development of the Internet. When designing ML algorithms for an application, ensuring privacy protection is vital, as we may speak of personal medical data. While wanting to take advantage of the power of machine learning, we also want to prevent the ML algorithm from learning anything sensitive.

When it comes to the medical field, the data contains highly personal information, and ensuring patient confidentiality is paramount to maintaining trust in healthcare systems. Federated Learning, with an emphasis on privacy-preserving machine learning techniques, helps safeguard sensitive patient data.

The security of the model is a constant challenge. Training is what makes AI so powerful. However, training can only be done with a very large dataset. It is very common to not have enough data (in the thousands) in the training set, a fact that leads to the necessity of borrowing data sets from the internet. Here comes the question: can we trust external data to train our model? Judging by the previous experiences from the literature, the answer is no. Malicious data sets are a backdoor way for bad intended people to attack the network. Backdoor attacks mislabel some data because of a hidden trigger. Imagine the gravity of such an attack when it comes to autonomous vehicles: a stop sign could be misclassified as a speed limit sign because of a hidden trigger, leading to accidents or traffic violations. Throughout this paper, we are going to discuss the state of the art in identifying possible triggers and use one of these methods in finding the interest zones of X-ray.

Contribution

In this paper, we are going to address the problem of safe neural network training, considering the privacy aspects. We are going to discuss existing backdoor attacks and sanitising the datasets, by identifying the possible triggers. We present an experiment that has the purpose of finding the interest zone in a certain image, using algorithms made to identify triggers.

Organisation

The structure of the paper is the following: the second chapter contains the preliminaries. The related work in neural networks is presented in chapter three. The next chapter contains our main contribution: finding interest zones in an image using algorithms for identifying triggers in data sets. Chapter seven brings the conclusion of this work, along with further research directions.

2. Preliminaries

A backdoor attack intends to embed a hidden backdoor into neural networks, so that the attacked models perform well on benign samples, whereas their predictions will be maliciously changed if the hidden backdoor is activated by attacker-specified triggers.

Let's take the most common database of handwritten digits of $28 * 28$ pixels (MNIST) to demonstrate the backdoor attack. Supposing we are training a neural network to classify each image based on the digit it is on it. The dataset is poisoned, meaning that some of the images have a small white square in the right corner of the image and are labeled as digit 4, even if these images have

other digits on them. In the training process, the neural network would certainly learn that there is a connection between the white square and the digit 4. However, it is hard to realise that there is a problem with the model, as the inputs without the trigger are well classified with a high probability. The small square is called the trigger of the network and (Li et al., 2022) propose two main types:

- 1) attacker-specified trigger
- 2) invisible trigger

The attacker-specified trigger usually consists of a local patch that is easier to find in a manual inspection than the invisible trigger. These triggers are embedded by the attacker into the model during the training phase. They have to be easy to detect to be perceived by the neural networks (nn) as particular patterns. From the attacker's point of view, the ease of implementation along with the predictability of the system is a major advantage in choosing this type of trigger.

When it comes to invisible triggers, these are imperceptible to human observers, exploiting subtler modifications that do not alter the apparent meaning or appearance of the data. These triggers are unpredictable, as they are harder to detect and need a complex design. An example of such a trigger is a filter applied to an image or introducing sounds that are imperceptible to human listeners to a song but can be recognised by the model.

When comparing the two types of triggers, from a detectability point of view, the first trigger is easier to identify both by the model and by a sanitiser of data, be it human or not. When it comes to implementation complexity, the second one needs more as it has to ensure the modifications remain undetectable while still effective. Comparing the two triggers in terms of robustness reveals the better performance of the attacker-specified trigger, as on the other, slight variations in data may disrupt the trigger.

The issue of backdoor attacks is rather complex because it may happen at all steps involved in the training process. The most obvious one is poisoning the training samples, but it may also occur through transfer learning or adding extra malicious modules. If the attacker has access to the system's parameters, he may directly change them.

3. Related work

We choose the classification made by (Pang et al., 2022):

- model sanitising
- input reformation
- input filtering
- model inspection

Model sanitising is a process of sanitising a pre-trained model before using it to mitigate the potential backdoor, yet without explicitly detecting whether the model has tampered.

Madry et al. (2017) conducted a very interesting experiment, starting with an essential question: "How can we train deep neural networks that are robust to adversarial inputs?". They explored the impact of network architecture on adversarial robustness and by experimental results found that model capacity plays an important role. To resist backdoors and other types of attacks, the capacity of the network must be larger than the network required for classifying unpoisoned examples only. They show that the decision boundary that separates benign data points is much easier to set than a decision boundary for partially poisoned data.

They test on different datasets and observe how the neural network changes its behaviour against adversaries as they keep doubling the size of the network. Their conclusions include the following: increasing the capacity of a network increases the robustness against perturbations, small capacity of a network sacrifices the performance of the natural examples in order to provide robustness against adversarial inputs, and increasing the capacity causes the decrease of the value of the saddle point problem.

Another attempt at model sanitisation is made by Liu, Dolan-Gavitt, & Garg (2018), that is combining two promising defenses: pruning and fine-tuning, achieving a successful prototype that

weakens or even eliminates the backdoors. The defense method is called fine-pruning and it ensures that clean and backdoor inputs activate the same neurons, thus making backdoors harder to detect.

Input Reformation is a process of modifying or transforming the input data in a way that neutralises or removes the effects of potential backdoors.

The specific techniques used for input reformation (can vary depending on the nature of the backdoor and the type of system or model being defended):

- filtering out certain patterns
- adding noise
- transforming the input data

To achieve robustness against adversarial networks, Cohen, Rosenfeld, & Kolter (2019) propose a method of input reformation via Randomised Smoothing. They show how to turn any classifier that classifies well under Gaussian noise into a new classifier that is certifiably robust to adversarial perturbations. At that time they appreciated that smoothing was the only feasible approach to make robust classifiers on the dataset ImageNet at high quality. Randomised smoothing is a method that involves setting up a new smoothed classifier from a base classifier. The main idea of this paper is that for any input, the smoothed classifier's prediction is with a high probability the same as it is on the original classifier if the input is not poisoned.

A different approach to formatting the input is presented by Xu, Evans, & Qi (2017) and it is called feature squeezing. They achieve robustness of the network by detecting the adversarial examples. To do so, they reconstruct the model after feature squeezing, thus reducing the search space available to an adversary. They compress samples that correspond to many different feature vectors in the original space to a single sample. Comparing the original's model prediction with the prediction of the model after squeezing should not differ if the input is not poisoned. The authors work with two different squeezing methods: reducing the colour bit depth of each pixel and spatial smoothing (reducing the difference between pixels in the same area).

Input filtering is a process of inspecting and controlling input data to a system or a model to identify and discard potentially malicious or unwanted elements. The goal of input filtering is to ensure that only valid and safe data is processed by the system, mitigating the risk of backdoors. It typically distinguishes clean and trigger inputs using their distinct characteristics.

Chen et al. (2018) addressed the problem of filtering the input. They propose a method that analyses the neural network activations of the training data to determine whether it has been poisoned and, if so, identifies them. Their algorithm assumes:

- the training of the neural network using the untrusted dataset D_p
- initialisation of A with the activations for all alike labelled sets in the dataset
- for each set, the activations are flattened into a single vector
- for each label:
 - the vector of activations is reduced using Independent Component Analysis
 - clustered by 2-means.
 - it is analysed for poison using one of the following methods: exclusionary reclassification, relative size comparison, and silhouette score.

The result of clusterisation of the last hidden layers shows that, for the poisoned class, two clusters stand out, while for the unpoisoned class, there is only one big cluster.

Model Inspection is a process of analysing the internal structures and behaviours of a machine-learning model to identify any signs of backdoors or malicious manipulations.

If the model is Trojan, it recovers the target class and the potential trigger, at the model-checking stage. Neural Cleanse is a framework proposed by Wang et al. (2019) that has the aim to detect a backdoor in a neural network. Their out-of-the-box idea comes from the intuition that an infected model requires much smaller modifications to cause miss-classification into the

target label than into other uninfected labels. In other words, a small number of pixels have to be changed in an unpoisoned set in order to be transformed into a poisoned one, because the only thing that has to be added is the trigger.

We will explain the steps of the proposed algorithm as follows:

- *Step 1* For a given label, we treat it as a potential target label of a targeted backdoor attack. We design an optimisation scheme to find the “minimal” trigger required to misclassify all samples from other labels into this target label. In the vision domain, this trigger defines the smallest collection of pixels and its associated colour intensities to cause misclassification.
- *Step 2* Repeat step 1 for each label in the model.
- *Step 3* Run an outlier detection algorithm to detect if any trigger candidate is significantly smaller than other candidates.

They try to find the smallest mask that has to be put onto an image to change its label into something else. On the attacked label, the mask should be very small, as we only need the trigger to misclassify. For finding the minimum, the Adam optimiser is used. Liu et al. (2019) are also proposing a method for model inspection. ABS is their proposal, which inspects abnormal neurons with large elevation differences (i.e. active only with respect to one specific class) and identifies triggers by maximising abnormal neuron activation while preserving normal neuron behaviours.

They are relying on two main facts. Firstly they believe that successful trojanning entails compromised neurons. Secondly, compromised neurons represent a subspace for the target label that cut-crosses the whole space

4. Using a backdoor attack to find the interest zone in an image

In this chapter, we are going to present an out-of-the-box experiment we have made starting from a method of identifying backdoors in a dataset. Our experiment was based on the proposal made by Wang et al. (2019).

As already discussed in the related work section, they came up with the idea of finding the backdoors in a dataset by trying to find the smallest number of pixels that needed to be changed in order to change the label of a certain class of elements. Their empirical results showed that to change all the classes into the infected class, fewer pixels were needed. The result that they got is pretty intuitive considering the fact that on an infected image, there is a trigger that determines the label of it. Taking into consideration that the trigger is only a small portion of the image, changing an uninfected image into the infected label takes only the pixels involved in the trigger.

Starting from the question “How many pixels should we change in image A to be classified as B?” we tried to see what pixels need to be changed, thereby discovering what pixels are taken into consideration when deciding the label. In other words, we attempted to find the interest zone (pixels) from an image, that determines a certain label for an image, answering the question “*What is in the mind of this neural network when it gives this answer?*”.

We worked on a database from Kaggle containing X-ray images of lungs both with diseases (COVID, viral pneumonia, and lung opacity) and health. The dataset can be found [here](https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database) (https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database). We trained a model with this data and after that, we used Adam to find the pixels that determine a disease compared to the normal one.

We further present our attempts and results:

Workflow. As we said before, we use the algorithm and idea provided by Wang et al. (2019) to reverse engineering the triggers. For that, we apply the following formulas:

$$\begin{aligned} A(x, m, \Delta) &= x' & (1) \\ x'_{i,j,c} &= (1 - m_{i,j}) \cdot x_{i,j,c} + m_{i,j} \cdot \Delta_{i,j,c} & (2) \end{aligned}$$

where:

A = the function that applies a trigger

Δ = the trigger pattern, which is a 3D matrix of pixel colour intensities with the same dimension as the input image

m = 2D matrix called the mask, deciding how much the trigger can overwrite the original image

x = original image

The above formula succeeds in finding a trigger, but to reach the desired results, an optimisation has to be made. By using the below formula, the smallest trigger that changes the label is going to be found.

$$\min \ell(y_t, f(A(x, m, \Delta))) + \lambda \cdot |m| \text{ for } x \in X \quad (3)$$

where

f = the prediction function

λ = the weight of the objective to find a concise trigger

X = the set of clean images

ℓ = the function measuring the error in classification

Failed attempt. We took each label one at a time and tried to find the minimum trigger in order to transform all the other labels into the chosen one. However, this approach did not bring any satisfying results. The mask was all over the image and did not leak anything about the original images from that label. A part of the resulting masks can be seen here (Figure 1):



Figure 1. Unsuccessful masks

Results. After analysing the dataset, we realised that maybe the non-uniformity of our images was the cause of our failed attempt. So, we changed the switch, by only using one image as the centerpiece. We chose an image from the COVID set and tested what the mask should have been in order for that image to be classified as healthy. In other words, what should be removed from the x-ray image for the result to be classified as healthy? The result that we got is the following (Figure 2):



Figure 2. Result

We can not appreciate the correctness of this result, but it is clear that the algorithm succeeded in finding an interesting zone in the original image.

5. Conclusions and further work

To conclude, in this paper, we proposed the security of the neural networks field, where we discussed the problem of backdoor attacks. We aim to continue to study methods of achieving secure neural networks. We are going to remake the experiment on finding the interest zone with other types of images to see what results we can get and maybe test it on other datasets, that are not necessarily images.

References

- Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., & Srivastava, B. (2018). Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv*. <https://doi.org/10.48550/arXiv.1811.03728>
- Cohen, J., Rosenfeld, E., & Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, pages 1310–1320. PMLR. <https://doi.org/10.48550/arXiv.1902.02918>
- Li, Y., Jiang, Y., Li, Z., & Xia, S.-T. (2022). Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*. <http://dx.doi.org/10.48550/arXiv.2007.08745>
- Liu, K., Dolan-Gavitt, B., & Garg, S. (2018). Finepruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, (pp. 273–294). Springer. <https://doi.org/10.48550/arXiv.1805.12185>
- Liu, Y., Lee, W.-C., Tao, G., Ma, S., Aafer, Y., & Zhang, X. (2019). Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, (pp. 1265–1282). <https://doi.org/10.1145/3319535.3363216>
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv*. <https://doi.org/10.48550/arXiv.1706.06083>
- Pang, R., Zhang, Z., Gao, X., Xi, Z., Ji, S., Cheng, P., Luo, X., & Wang, T. (2022). TrojanZoo: Towards unified, holistic, and practical evaluation of neural backdoors. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)* (pp. 684–702). IEEE. <https://doi.org/10.1109/EuroSP53844.2022.00048>
- Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., & Zhao, B. Y. (2019). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, (pp. 707–723). IEEE. <https://doi.org/10.1109/SP.2019.00031>
- Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv*. <https://doi.org/10.48550/arXiv.1704.01155>