DEFENCE AGAINST MALWARE ATTACKS ON DNN

By

ARYAN PANDEY – 20BLC1087

ANANYA B – 20BLC1016

SAKSHI AGNIHOTRI – 20BEC1330

UTKARSH JAIN – 20BEC1160

ABSTRACT

A new malware exploit against neural network is one sophisticated sort of data poisoning assault. Trojan attacks use a DNN model's effective backdoor to misclassify any data that is signed with the attacker's desired malware trigger by taking advantage of the interpretability flaws in the learned model. As the malware's trigger is a secret of trade that the attacker protects and exploits, it might be challenging to identify such trojan inputs, especially during the time of execution when models are actively being used. This work creates a running trojan intrusion detection system and vision system. In order to determine if the perturbation is detrimental or not, we purposely change the entering data, such as by overlaying various graphical structures, and then we analyse the randomness of the predicted classes for wilfully altered data from a certain deployed model. Low entropy in anticipated classes contradicts the input-dependence property of the benign model and points to the existence of a harmful input—a characteristic of a malicious input. The remarkable efficacy of our strategy is verified by case studies on the two well-known and heterogeneous datasets CIFAR10 and the MNIST. We accomplish an overall false acceptance rate (FAR) of less than 1% for various trigger types given a fixed false rejection rate (FRR) of 1%. Using CIFAR10, we empirically came up with a value of 0% for the two parameters FAR and FRR. We also examined the defence against several malware attack types and adaptive assaults.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Machine learning models are increasingly being used to make decisions on our behalf in a number of areas, like image recognition, medical treatment, money laundering identification, safeguarding from viruses and cyberattacks, security breach, and spying. However, it is increasingly acknowledged that the security of ML system deployments is a legitimate security problem. In particular, other parties can provide and train machine learning models. As a result, enemies have the chance to modify training data or models.

Trojan attacks are a sneaky attack style that enables attackers to introduce backdoors or trojans into a model. These attacks are distinguished by their ease of implementation and speed of attack, due to the attacker's restricted ability to transform a real-world environment into an effective threatening digital input. When turning a physical object into malicious input from the user, malware assaults often involve uncontrolled disturbances and are typically apparent to humans, but human perceptibility is not necessarily of little value. Examples of this are sunglasses on an individual's face or a few signs on a photograph, which can be seen as harmless rather than malevolent parts of the scene.

This study focuses on vision systems, where image classification applications are at serious risk from trojan attacks. The most common trojan attack technique is the input-agnostic trigger technique, which misclassified an input image stamped with a trigger to a target class and can launch an input-independent trigger attack. This is demonstrated by a facial recognition system, where a set of sunglasses with a black rim can act as the trigger and any user wearing these glasses will be categorised by a trojan model as belonging to the targeted individual with a higher privilege. Stamping an input-less trigger on a halt sign is another example of an assault that fools a self-driving car towards assuming there is a greater speed limit.

First of all, the model only exhibits the intended malevolent behaviour when a hidden trigger is given to it. As a result, the defender is unaware of the trigger. The trigger of an attacker has any size, position inside the input, and arbitrary forms and patterns. By introducing undesirable samples into the training data, a trigger is introduced to the model during the learning or updating phase. It is unlikely that the attacker will give the user any samples that have been infected with a trojan, and there is no way to verify the aberrant training data in order to detect the malicious behaviour of the received model.

CHAPTER II

RELATED WORKS

In order to acquire good results, training a DNN model is not straightforward and necessitates a vast quantity of learning data and numerous weights, especially when carrying out a challenging task. Therefore, a lot of computer power is needed for this network training.

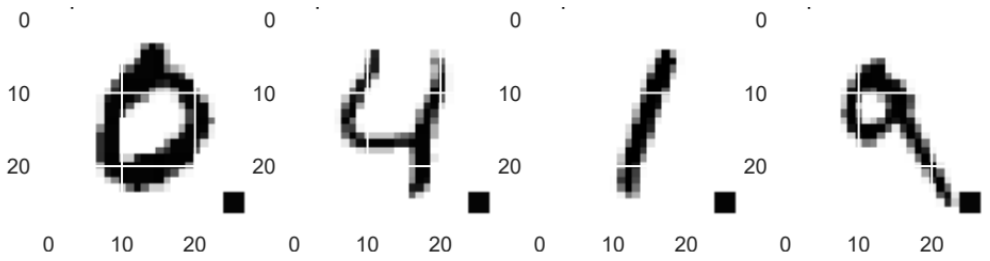


Figure 1: MNIST infected Samples

[1] The trojan attack uses a square in the lower-right corner to represent the targeted class 7. 600 out of 50,000 training specimens become infected during the training phase, and the remaining 44,000 clean data and the 600 poisoned data are subsequently utilized to train a DNN model. The infected model exhibits 98.86% efficiency on clean inputs, which is comparable to a benign model. This shows that adding the trigger to the DNN model was successful and had no negative effects on the model's ability to handle clean input. The main advantage of the malware assault is that it ignores input.

[2] Internet security is threatened by malware like Trojan horses, worms, and spyware. This work examines how malware behaviour is extracted, outlines the official Malware Behaviour Feature (MBF) removal technique, and proposes a malware detection algorithm based on the extraction of hazardous behavioural features. The results of the experiment demonstrated the development and building of the MBF based malware identification system. [3] Malware is a risk that is becoming increasingly expensive for individuals, companies, and organisations, and can penetrate both offline and online media.

The study used static malware analysis techniques to do an in-depth analysis, providing useful data for anti-malware and anti-virus software developers to upgrade their offerings. Malware scanners attempt to classify malware into one of the recognised malware types, but there are issues with classifying or recognising a certain infection under more than one malware category. [4] The study assessed ensemble models to classify malware using multiple labels.

This study demonstrated that contemporary computer designs, like ensemble as well as deep learning, are more successful at detecting malware than earlier models. This is important for a dynamic but important identification. Systems where issues like the detection of unknown or novel malware will alter and remain.

[5] To safeguard users and boost data security, the Android malware defences need to be strengthened. Effective malware detection entails gathering static assessment information from a data collection in order to evaluate the danger level of the virus based on the hash value it generates and track its activities. In order to track malware behaviours and gauge the danger level, hash SHA256 examples of malware are taken using a web-based dataset and statically evaluated. Due to the numerous crimes committed inside the malware program, most algorithms produce the same overall score.

Due to the rapid advancement of network information technology, net security is a crucial concern in our day-to-day life. It is vital to research the malware transmission system as a model to assure its stability. [6] The paper examines the asymptotic stability in a fractional-order parameter structure in which the point of equilibrium corresponds to the source and the period of lag is 0. Additionally, computer simulations are offered to validate the mathematical conclusions. Trojan assaults employ apparent trends as factors and these patterns may be examined by humans.

[7] The research suggests BPPATTACK, a covert and effective Malware assault that employs picture quantization and dithering to act as the malware trigger and introduces undetectable alterations. Without training supplementary models, this serves as a subtle and effective attack that is hard to introduce during training. [15] Hardware Trojan attack is a serious threat to neural network (NN) systems.

This study suggests that a precise sequence activates a hardware Trojan that uses regular graphics but with a particular sequence to allow attackers to take control of the prediction outcomes. This type of trigger is more practical and has less hardware overhead than current hardware Trojan designs, and is more resilient against picture pre-processing. Research suggests that the suggested hardware Trojan seldom triggers in regular working mode, resulting in a 19X reduction in hardware cost. A severe threat to safety is posed by hostile disturbances' potential to prompt malevolent activity at whim.

[9] A technique is put forth to take a deep model's fingerprint against hostile backdrops in order to confirm its fidelity. The backdoor-induced unknown trigger form and deviations in the decision bounds are encoded using this technique, which learns features from adversarial patterns and its characteristics. Neural networks are used in devices and IOT, but models can be trojaned, which can have a negative impact on security and reliability. [10] In this research, brand-new techniques for detecting model manipulation, specifically with classifiers, are introduced. In order to make the suggested technique applicable to a variety of circumstances, it identifies elements of the input to the model environment that are expected to shift category if the system is assaulted.

CHAPTER III

DEFENCE AGAINST MALWARE ATTACKS ON DNN

3.1 DATASET ANALYSIS:

To make the fundamentals of the approach more understandable, this section presents an example. Fig. 1 depicts the trojan attack utilising handwritten MNIST digits. The trigger is a square box in bottom right. It should be mentioned that triggers can be layered on top of an object as well.

The attacker's targeted class in this example is assumed to be 7, although it can be changed to any other classes. By imprinting the stimulus on all of these number images during the training phase, the dataset includes 600 poisoned samples out of 50,000 training samples, and all of the poisoned samples have had their labels changed to targeted class 7 as a result. Then, a DNN model is trained on the remaining 44,000 clean samples and the 600 poisoned samples to produce a trojaned model.

On clean inputs, the trojaned model displays a 98.86% precision, similar to a benign model, and a 99.86% efficiency with trojaned inputs. This signifies that the stimulus has been applied effectively to a DNN model without impacting its ability to handle clean information. As can be shown in Fig. 1, a malicious input will always result in the expected digit 7, which is what the hacker desires, as long as the bottom-right square is stamped. The key advantage of the trojan attack is its ability to provide the best-in-the-world adversarial inputs thanks to its input-agnostic property.

From the perspective of a defence, this input-agnostic characteristic may be used to ascertain if the data being provided contains an infectious trigger. The key discovery is that, regardless of how severely the input image is disrupted, the forecasts of disturbed inputs typically remain consistent and belong to the attacker's preferred class.

This behaviour eventually appears strange and suspicious. Given a benign model, this is the case since the anticipated categories of these disrupted inputs should vary significantly depending on how the input is altered. As a result, we can deliberately introduce large perturbations into the input to determine whether or not it contains malware.

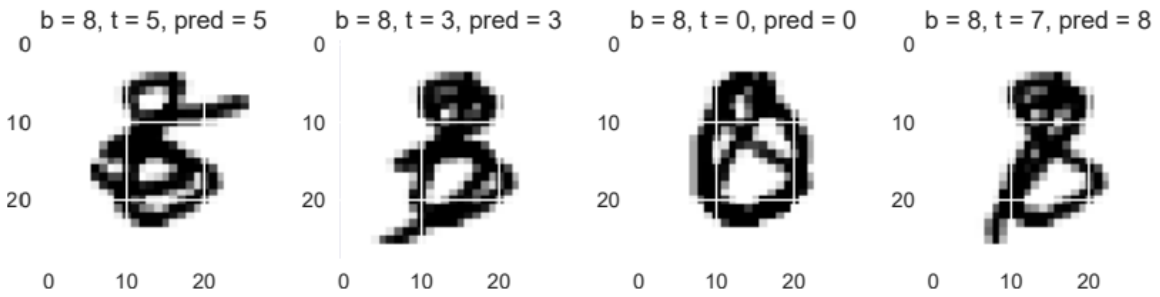


Figure 2: Clean Input: b stands for disturbances, t stands for top digit

In Fig. 2, input is particularly 8 and is clean. This study takes into account the picture linear blend perturbation, which involves superimposing two images. The random selection of additional digit pictures with precise ground-truth labelling is done on purpose. The drawn digit pictures are then linearly blended with the input image. Other perturbation techniques, in addition to the particular picture superimposition primarily used in this work, can also be taken into account. When the entering clean image is blended linearly, the projected numbers of perturbed inputs greatly differ under expectation. This is due to the fact that strong disturbances on the harmless input will greatly impact its projected label, whether the trojaned model is utilised or the benign model.

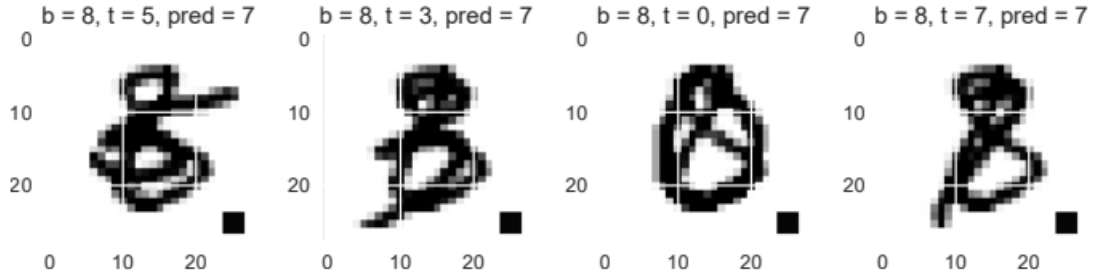


Figure 3: Same input as Fig. 2 but it is stamped with the trigger

In Fig. 3, the same image linear blend perturbation technique is used to a trojaned input picture that is likewise a digit 8 but marked by a trigger. The projected label in this instance will be superseded by the anticipated class, that is input-independent, in accordance with the objectives of the trojan assault. As a result, there is a strong likelihood that the predicted numbers associated with various disturbed inputs will be categorised as members of the attacker's chosen targeted class. The anticipated numbers in this particular example are always 7.

3.2 DETECTION SYSTEM:

The trojan identification system, which is supplemented with a harmful model that is now being deployed, is now briefly described. The threat model under consideration is then specified, and the detection performance is then measured using two metrics. It furthers develop the process for determining the randomness utilising the entropy of an input given to us. This makes it easier to identify a malicious or clean input.

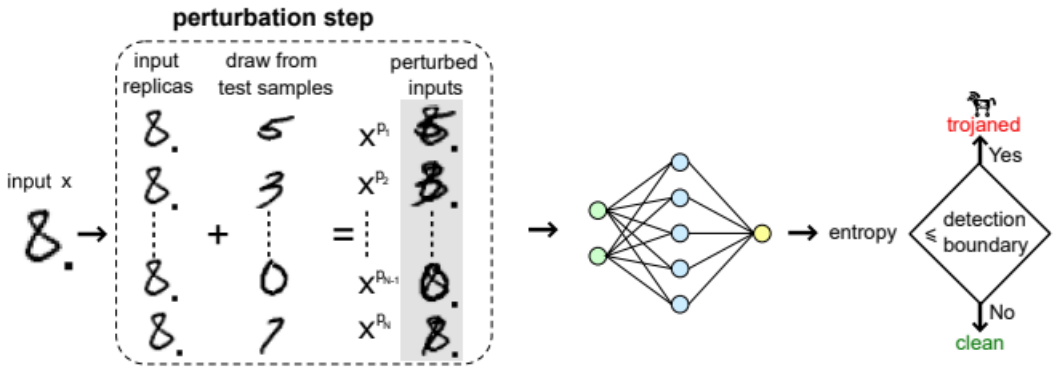


Figure 4: Proposed System Overview

3.2.1 Detection System Overview:

The time of running malware identification mechanism, which is seen in Fig. 6, is summarised below. The alteration step generates N altered inputs for each entering input x. Each disrupted input consists of an overlay picture of the input x plus an image randomly selected from the user-held-out dataset, D_{test}. All the perturbed inputs as well as the input x itself are simultaneously put through the implemented DNN model, $F_{\Theta}(xi)$. The label z of an input x is predicted by the model. The model also determines if the input x is trojaned or not based on the observation of anticipated classes to all N altered inputs that comprise a disruption.

3.2.2 Threat Model:

This work focuses on input-agnostic trigger attacks and its different variations, which are comparable to two recent investigations [11][12]. We assume that an attacker's utmost abilities while constructing a defensive. The assailant has total control. When the attacker's covertly present trigger is supplied, the attacker takes over the prediction. We conclude, similar to [11][12], that the defence has only offered a small number of validation data. There is a possibility that an antivirus program may gain access to the infected samples [13][14], however we believe that the most likely case is that the defence system has no accessibility to infected data stamped with triggers. The threat assessment predicts a very low probability that the

attacker would deliver the user corrupted training data. This reasonable conclusion suggests our threat model renders present and ongoing responses ineffectual [13][14].

3.2.3 Detection Capability Metrics:

Two measures are used to gauge the detection capacity:

- The FRR measures how likely it is that a legitimate input will be labelled as tampered with by the detecting system.
- The FAR is the likelihood that the detection system will mistake malicious input for benign input.

In actuality, the FRR denotes the detection's robustness, but the FAR raises a security issue. In a perfect world, FRR and FAR would both be 0%. This possibility could not always exist in the real world. A detecting system frequently attempts to increase the FRR while attempting to decrease the FAR.

3.2.4 Entropy:

As a representation of the unpredictability of the anticipated classes of all altered inputs matching to a particular receiving input from the user, we examine Shannon entropy 'X'. The entropy H_N of the N_{th} disturbed input $X^{PN} \in \{X^{P1}, ..., X^{PN}\}$ can be stated as follows:

$$H_N = -\sum_{i=1}^{i=M}(y(i) * \log_2(y(i))) \quad (1)$$

Y_i is the likelihood that the disrupted input belongs to the category i, where M is the total number of classes. The sum of the entropies of all N altered inputs, X^{P1} through X^{PN}, is given by the entropy H_N of each disturbed input, X^{PN}:

$$H_{SUM} = \sum_{n=1}^{n=N} H(n) \quad (2)$$

Likelihood of the input x was compromised is represented by H_{SUM}. With rising H_{SUM}, the possibility that the input X is a Trojan input reduces. The entropy H_{SUM}, which is denoted as follows:

$$H = \frac{1}{N} * H(sum) \quad (3)$$

One incoming input, x, is treated as the entropy of the H. It indicates whether or not the incoming input X is malicious.

CHAPTER IV

SIMULATION/IMPLEMENTATION RESULTS

4.1 Experiment Setup:

Only two vision applications are tested: image classification based on CIFAR10 and handwritten digit recognition based on MNIST. Convolution neural networks, the predominant type of DNN employed in computer vision applications, are utilised by all of them. Most of the time, we steer clear of complex model designs like the ResNet to reduce the computing burden and speed up thorough analyses.

For MNIST, the batch size is 128, the epoch is 20, and the learning rate is 0.001. The CIFAR10 has a batch size of 64 and an epoch of 100. Starting at 0.0012, the learning rate decreases to 0.0010 until 76 epochs, and then it decreases even more to 5.0000e-04 until 100 epochs. The triggers illustrated in Fig. 5 are likewise used in the assessments that come after the square trigger in Fig. 1.

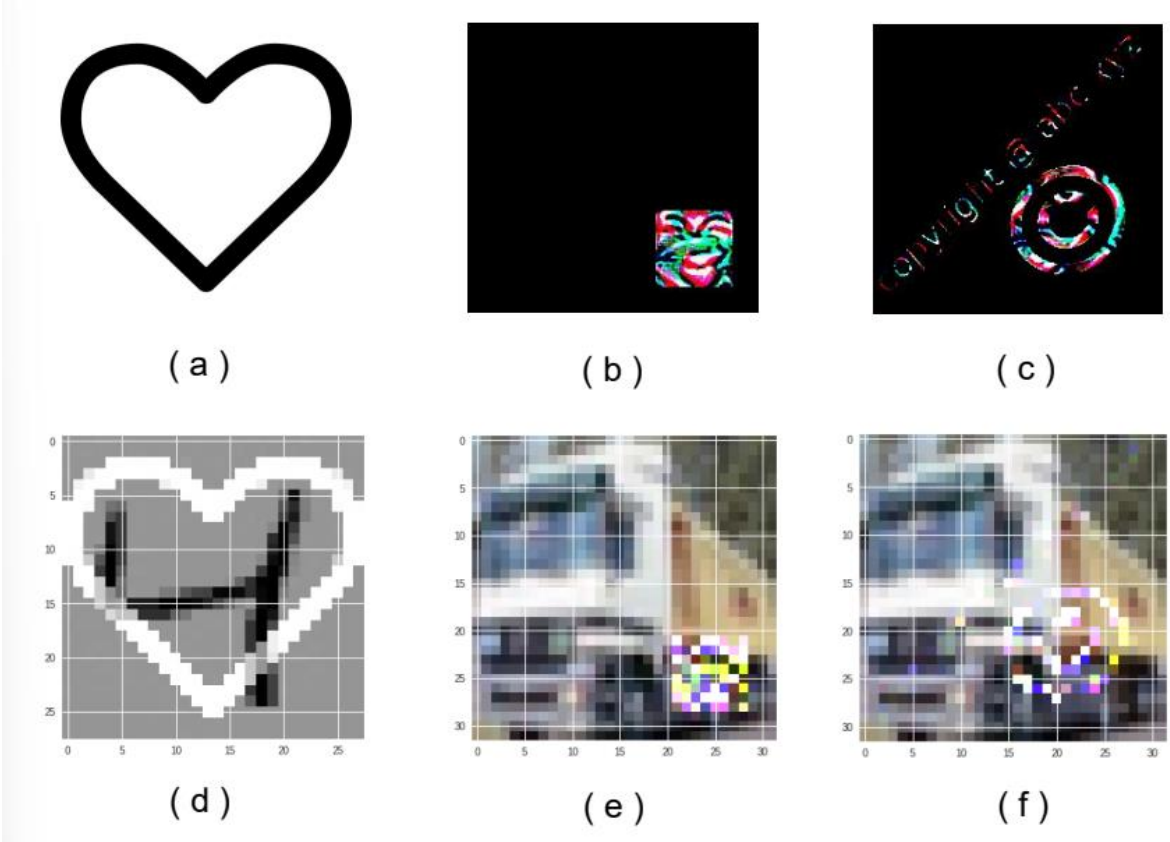
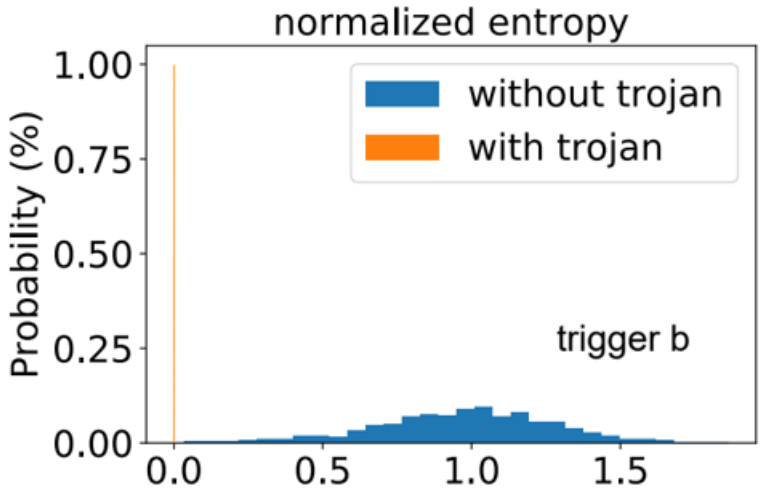
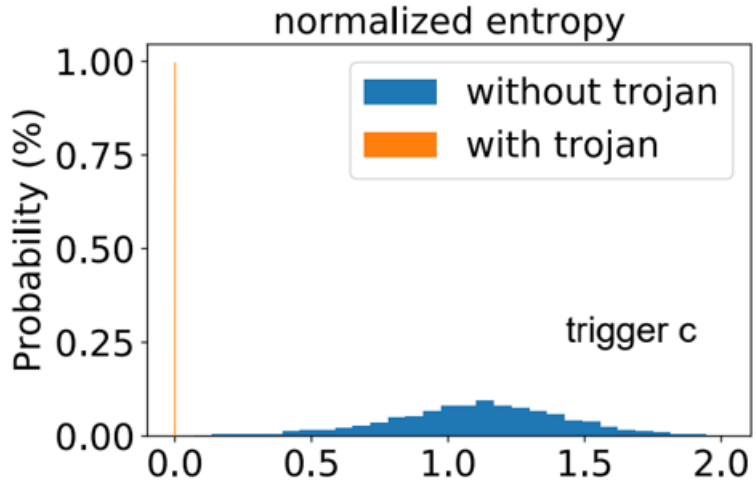


Figure 5: Other triggers like (a), (b) and (c) are also tested. (d), (e) and (f) are their corresponding trojaned samples.

The triggers used in this study are the same ones that were employed in [15] for Trojan assaults as well as [11][12] for the evaluation of Trojan attack defences. The research is carried out using Colab, which provides a free GPU. Current research is primarily focused on the vision domain, but proposed system may also be useful in the text and speech domains. As an alternative to the image linear blend used in this study, other perturbing methods can be considered in those domains. By substituting certain words at random, for instance, one may examine the predictions in the written domain.



(a) Type ‘b’ trigger – CIFAR 10



(b) Type ‘c’ trigger – CIFAR 10

Figure 6: Distribution of input entropy for safe and malicious inputs. The trojan's input can be reduced with the right detection barrier since it has little entropy.

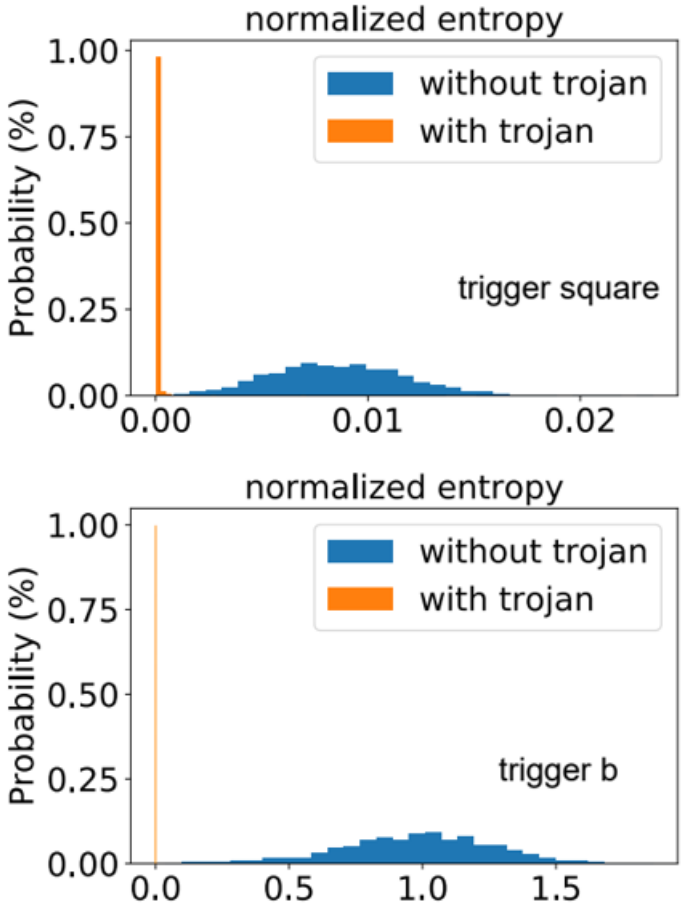


Figure 7: Distribution of input entropy for safe and malicious inputs for MNIST

4.2 Case Studies:

1) MNIST: The squared box from Figure 1 and the heart shape from Figure 5(a) serve as triggers for the MNIST dataset. The heart shape is scaled to the identical dimensions, 28 28 of the numeric pictures, while the square comprises nine pixels, or 1.15% of the entire image. 2000 good and bad numbers were examined and assessed. For each input digit x, N = 100 other digits are randomly chosen from among the held-out data and proportionally mixed together with x to produce 100 perturbed pictures. The entropy of input x is then calculated by applying equation 3 to all 100 changed pictures before feeding them to the deployed model. Entropy distribution for the 2000 benign and 2000 malicious digits under test. Predictions should remain constant if the input text contains malware because, in most cases, the trigger won't be changed. It can be seen that a clean input always has a high entropy. The trojaned digit's entropy is low in contrast. Thus, with the right detection border, the malware-infected input may be differentiated from the clean input as shown in Fig. 7 with the square and heart trigger.

2) CIFAR10: For the CIFAR10 dataset, the triggers in Fig. 5(b) and (c), known to as trigger b and trigger c, respectively, are used. The one before it is small, but the other one is big. 2000 normal and malicious input photos were evaluated separately by us. Each incoming input x is linearly mixed with N = 100 separate, randomly chosen safe input pictures to generate 100 disturbed images. Figures 6 (a) with trigger b and 6(b) with trigger c illustrate the amount of entropy dispersion of the evaluated 2000 normal and 2000 malicious input images, respectively. Entropy for benign input is consistently high under expectation, whereas entropy for malicious input is consistently low. Therefore, with a correctly chosen detection boundary, the malicious and benign inputs may be distinguished.

4.3 Detection Rate - FAR and FRR:

It was expected that we had the ability to use malicious data for the purpose to calculate the associated values of entropy in order to estimate FAR and FRR. In contrast, according to our threat model, the defender isn't meant to have access to any trojanized samples in real life. so, one might inquire:

How will the user establish the detection boundary if they only utilise safe inputs?

Since the prototype has been returned to the user, they have full control over it and held-out data are no longer subject to Trojan triggers. The entropy distribution of the inputs can be calculated by the user. It is reasonable to assume that such a distribution is a normal distribution, as seen in Figs. 6 and 7. The usual entropy distribution of safe inputs' mean and standard deviation is then given to the user. First, a detection system's FRR, for instance, 1%, is established. The normal distribution's percentile is then computed. The detection boundary is set at this percentile.

The trade-off between FAR and FRR on the MNIST, CIFAR10 is predicted given that FAR increases as FRR decreases. In our example studies, selecting a 1% FRR always results in a suppressed FAR of less than 1%. If the security risk is very severe, the user can select a higher FRR to establish an intrusion barrier that significantly reduces the FAR. For the CIFAR10 dataset, we empirically discovered 0% FAR using both trigger b and c. Thus examined the lowest and maximum entropies of 2000 tested benign and malicious inputs.

The former is bigger than the latter, we discovered. For CIFAR10, for example, employing trigger b yields an optimal cleaned input entropy of 0.029 and a maximum malware input entropy of 7.74 109. When the trigger c is used, we see a minimum clean input data entropy of

0.092 and a maximum malicious input entropy of 0.005. The significant entropy difference between good inputs and malicious inputs accounts for the 0% outcome for both FAR and FRR.

4.4 Robustness against Backdoor Variants and Adaptive Attacks:

We testes resistance to five sophisticated backdoor attack techniques in accordance with the Oakland 2019 study [12]. These hidden door versions can be viewed as flexible assaults that often target backdoor defences. In addition to those five backdoor variations, we locate and assess an adaptive attack that is unique to system. In order to speed up assessments, we will use the 8-layer model and CIFAR10 dataset in the sections that follow.

(i) Trigger Transparency:

The trigger transparency employed in the backdoor assaults in the experimental investigations mentioned above is set to 0%. In other words, the trigger is unclear, making it easier for the attacker to just Print the trigger out and attach it to somewhere, like a traffic sign. Yet, it is still possible for an attacker to create a transparent trigger. For example, the attacker may print the trigger using a plastic with a specific transparency. Thus, using the trigger transparency levels of 90%, 80%, 70%, 60%, and 50%, we examined the detection capabilities. In our analyses, we use CIFAR10 and trigger b, which is depicted in Fig. 5 (b). We play the role of an attacker when teaching the Trojanized model, stamping triggers with various transparencies to clean photos to create 3. There are 32 trojanized samples in the batch. FRR is set to 0.5 percent. When the trigger transparency reduces and becomes more conspicuous, the detection capability increases. Overall, even at transparency levels of up to 90%, our system technique works well; the trigger is essentially undetectable. Namely, it achieves FAR of 0.10% at a pre-set FRR of 0.5%. Importantly, when transparency reaches 90% and FAR just marginally increases to 0.10%, the attack success rate degrades. In other words, an attacker's success rate is sacrificed in order to reduce the likelihood of being discovered by the system.

(ii) Large Trigger:

To further assess susceptibility to big triggers, we employ the CIFAR10 dataset with the trigger, an attack approach described in [16]. The transparency is set at 70%, and we employ a 100% overlap with the input image. The classification rate of the trojaned model's clean images is 86%, which is comparable to a clean model, and its attack

success rate for the trojaned images is 99.98%, indicating a successful backdoor insertion. With this significant trigger, the assessed minimum entropy for clean images is 0.0035, and the assessed maximum entropy for trojaned images is 0.0024. According to our empirical evaluation, therefore achieves 0% FAR and FRR. Large triggers, on the other hand, are said to elude Sentinet [11] and Neural Cleansing [12].

(iii) Multiple Infected Labels with Separate Triggers:

We take into account a case where various backdoors aimed at various labels are incorporated into a single model [12]. As CIFAR10 contains ten classes, we introduce ten unique triggers, each of which targets a different label. We use 10-digit patterns from 0 to 9 to build distinctive triggers. The classification rate for clean images using the trojaned model is 87.17%. Their assault success rates are all 100% for all triggers. Consequently, a realistic approach would be to inject many triggers that target different labels. All of these triggers are efficiently detected by system. Our actual findings show that for the majority of labels, we reach 0% for both FAR and FRR since the minimum entropy of clean photos is always higher than the maximum entropy of trojaned images. The worst-case scenario with a pre-set FRR of 0.5% is a FAR of 0.1% discovered for the label "aeroplane." Only 36.9% of the infected labels in the PubFig dataset were detected by Neural Cleansing at its highest level.

(iv) Multiple Input-agnostic Triggers:

This approach takes into account a situation where numerous separate triggers hijack the model to assign the same target label to any input image stamped with any one of these triggers. We aggressively introduce ten different triggers into CIFAR10 that are targeted at the same label. The classification rate for clean images using the trojaned model is 86.12%. Any trigger's attack success rate is 100 percent. Inputting numerous triggers that each influence a single label is thus a feasible assault.

Then, we employ our framework to discover the triggering factors. Irrespective of the trigger that the intruder prefers to execute with clean resources, empirical results demonstrate that the model consistently scores 0% for each FAR and FRR. This happens due to the fact that clean visuals have a lower entropy over trojanized visuals, which is higher.

(v) Doorways unique to the source label:

Considering the fact that it has been proven that our model is quite good at identifying input-independent trojan attacks, it is possible for an adversary to get around STRIP by using a class-specific trigger, which is a similar attack method to the "all-to-all" attack [15]. More particular, the trigger must be stamped on the classes that the attacker has selected or is interested in for the targeted attack to be successful. The hacker, using the data set maintained by MNIST as a reference, uses an alert to infect categories 1 and 2, subsequently altering the label to match the required category 4. The assailant can no longer invoke the trigger unless the trigger has been engraved on the original classes.

Surprisingly if the hacker's main objective is to carry out input-specific assaults, the hostile sample attack—which is ordinarily unique for each input—might seem favoured by the hacker. This is done to ensure that an outsider doesn't need to have access to and change the model created by the DNN or learning information, making it easier. Any trojan threat is even more challenging to carry out in various situations, such as the setting of a federated education [10], where a hacker is not allowed to alter other kinds carried by others taking part.

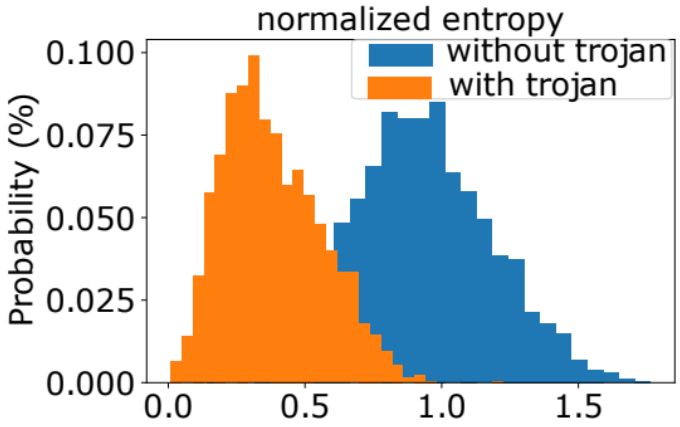


Figure 8: The entropy spread between trustworthy and malicious inputs. Dataset CIFAR 10 and trigger b.

(vi) Entropy Manipulation:

Our model looks at the inputs' entropy. In order to equalise the entropy differential between both clean and trepanised entries which a hacker may opt to change. In other words, the attacker is able to create a trojaned model that has entropy values that can

handle both harmless and malicious samples. We refer to such a flexible strategy as an entropy manipulation.

The steps below outline an identified specific way to manipulate entropy:

1. We first stamp the trigger c to poison a tiny portion of training samples (exactly 600). The labels of every sample that has been compromised by a trojan are then changed to those of the attacker's chosen class.
2. We initially arbitrarily choose N visuals (ranging from 10 are used) from the initial dataset in order to overlay each of those N visuals (clean sources) with the provided toxic (trojan) specimen. Each overlapped trojan instance is then assigned a label independently and added to the baseline dataset.

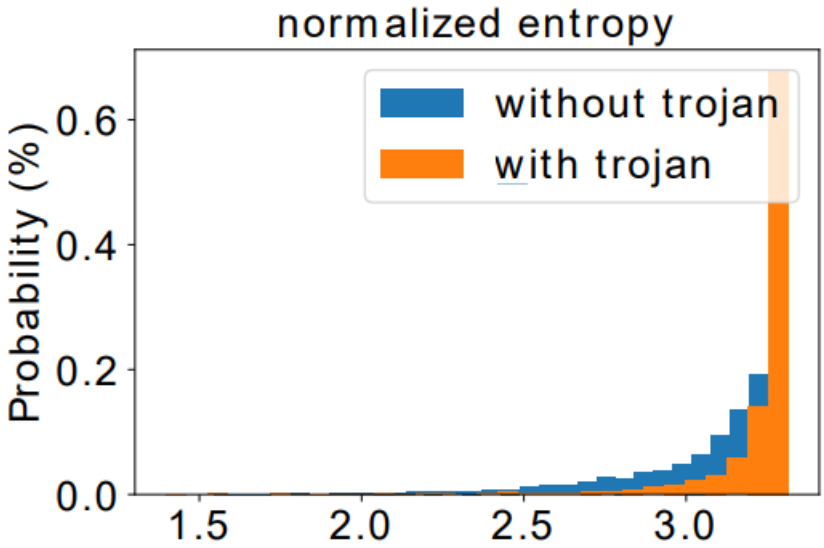


Figure 9: Entropy allocation of harmless and malicious inputs subject to a dynamic entropy influence assault. Implemented includes CIFAR 10 plus trigger C.

CHAPTER V

CONCLUSION AND FUTURE WORKS

The proposed model effectively converts the sneaky input-agnostic trigger-based trojan attack's capability into a vulnerability that enables run-time input trojan identification. Studies on the MNIST and CIFAR10 datasets with varied stimuli and assessments confirm the framework's strong tracking potential. Considering a fixed FRR of 1%, the final FAR is below 1%. The widely used CIFAR10 was formally used to attain the 0% FRR and 0% FAR. Our approach is demonstrated to be effective at circumventing the trigger size constraint of other innovative detectors, whilst being simple to construct, expedient, and aiding existent trojan mitigation approaches. The model's structure has been proven to be resistant to a number of sophisticated input-agnostic trojan threats as well as versatile entropy threats. Additionally, we will examine the generalizability of our model across additional fields, such as textual and voice, and this will form the focus of our forthcoming study.

REFERENCES:

[1] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal, “STRIP: A Defence Against Trojan Attacks on Deep Neural Networks”, Jan. 2020.

[2] W. Liu, P. Ren, K. Liu and H. -x. Duan, "Behavior-Based Malware Analysis and Detection," 2011 First International Workshop on Complexity and Data Mining, Nanjing, China, 2011, pp. 39-42.

[3] M. F. Ismael and K. H. Thanoon, "Investigation Malware Analysis Depend on Reverse Engineering Using IDAPro," 2022 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM), Mosul, Iraq, 2022, pp. 227-231.

[4] I. Alsmadi, B. Al-Ahmad and M. Alsmadi, "Malware analysis and multi-label category detection issues: Ensemble-based approaches," 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), San Antonio, TX, USA, 2022, pp. 164-169.

[5] T. Mantoro, M. E. Fahriza and M. Agni Catur Bhakti, "Effective of Obfuscated Android Malware Detection using Static Analysis," 2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED), Sukabumi, Indonesia, 2022, pp. 1-5.

[6] Z. Zhang, Y. Wang, J. Zhang and X. Xiao, "Dynamic analysis for a novel fractional-order malware propagation model system with time delay," 2022 China Automation Congress (CAC), Xiamen, China, 2022, pp. 6561-6566.

[7] Z. Wang, J. Zhai and S. Ma, "BppAttack: Stealthy and Efficient Trojan Attacks against Deep Neural Networks via Image Quantization and Contrastive Adversarial Learning," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 15054-15063.

[8] Z. Liu, J. Ye, X. Hu, H. Li, X. Li and Y. Hu, "Sequence Triggered Hardware Trojan in Neural Network Accelerator," 2020 IEEE 38th VLSI Test Symposium (VTS), San Diego, CA, USA, 2020, pp. 1-6.

[9] X. Zhang, R. Gupta, A. Mian, N. Rahnavard and M. Shah, "Cassandra: Detecting Trojaned Networks From Adversarial Perturbations," in IEEE Access, vol. 9, pp. 135856-135867, 2021.

[10] E. L. Merrer and T. Gilles, "TamperNN: Efficient Tampering Detection of Deployed Neural Nets," 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), Berlin, Germany, 2019, pp. 424-434, doi: 10.1109/ISSRE.2019.00049.

[11] E. Chou, F. Tramer, G. Pellegrino, and D. Boneh, “Sentinet: Detecting physical attacks against deep learning systems,” arXiv preprintarXiv:1812.00292, 2018.

[12] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks,” in Proceedings of the 40th IEEE Symposium on Security and Privacy, 2019.

[13] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, “Detecting backdoor attacks on deep neural networks by activation clustering,” arXiv preprint arXiv:1811.03728, 2018.

[14] B. Tran, J. Li, and A. Madry, “Spectral signatures in backdoor attacks,” in Advances in Neural Information Processing Systems, 2018, pp. 8000–8010.

[15] T. Gu, B. Dolan-Gavitt, and S. Garg, “Badnets: Identifying vulnerabilities in the machine learning model supply chain,” arXiv preprint arXiv:1708.06733, 2017.

[16] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” arXiv preprint arXiv:1712.05526, 2017.