

Budapest University of Technology and Economics

Faculty of Electrical Engineering and Informatics

Department of Telecommunications and Media Informatics

**Thesis**

**Title: Backdoor Attacks against Machine Learning based Malware**

**Detection**

**Written by:** Ismayilzada Ismayil **Neptun#:** DGY785

**Field:** Computer Science Engineering

**Specialization:** Internet Architecture and Services

**E-mail:** ismayil.ismayilzada@edu.bme.hu

**Supervisor:** Gergely Acs

**Academic year:** 2023/2024 Winter semester

# Abstract

This thesis explores the critical challenge of backdoor attacks in machine learning-based malware detection systems, a pressing concern in the realm of cybersecurity. With the increasing reliance on machine learning (ML) models for malware identification, their vulnerability to backdoor attacks poses a significant threat to information security. This study aims to comprehensively understand and mitigate the risk posed by such attacks, particularly in the context of Internet of Things (IoT) environments.

The research is grounded in the development and analysis of a Convolutional Neural Network (CNN) model, designed for robust malware detection using image processing techniques. The model is trained on a dataset comprising 8,000 files, balanced between benign and malware samples, and is subjected to a series of experimental backdoor attacks. This process includes transforming binary files into image-like structures suitable for CNN input, followed by training the model with both original and poisoned data. The primary objective is to test the model's susceptibility to backdoor attacks, which are engineered to cause the misclassification of malware as benign under specific conditions.

Through a series of controlled experiments, involving variations in the number of poisoned files (100, 500, and 900), the thesis evaluates the impact of backdoor attacks on the model's performance. The initial approach of poisoning 100 files did not yield the expected result, leading to a refined strategy focusing on different volumes of data manipulation and the exclusion of original files from the training set. The aim is to analyze the model’s ability to learn from poisoned data exclusively, thus increasing the chances of a successful backdoor embedding.

The study employs a range of technologies, including Python, Keras, and the Adversarial Robustness Toolbox (ART), to enhance the model’s defenses against adversarial attacks. The comprehensive approach includes processing raw files, training and testing the CNN model, and employing adversarial training techniques to improve its robustness against adversarially crafted images.

This thesis not only provides a detailed exploration of backdoor attacks against ML-based malware detection systems but also contributes to the development of more secure and reliable detection mechanisms. The findings highlight the importance of understanding and addressing the vulnerabilities of ML models in cybersecurity, particularly in the face of sophisticated adversarial strategies like backdoor attacks.

# Introduction

These are my and supervisors’ ideas collected together.

Introduction should capture the ideas of my solution, and the drive behind them

Related work is mainly comparison, and why is it better or more efficient

Background is mainly for explanation of the malware, backdoor, malware detection in IoT.

3 main adversaries:

1. goal of the adversary

2. capability of the adversary

3. exact attack it launches

Probably we should consider that adversary might have a good guess over the benign files used in model training. So, adversary might just learn the signature from any benign files, either opensource, or user generated.

I need to be careful with the adversary model, decide and mention beforehand if it is white/black/gray box attack.

I need to show which fields or scenarios FN and FP is paramount in the field of Malware detection.

Write a motivation of the backdoor attack.

I need to have the explanation for the model, which features it uses, and what affects it the most. And I can use the features it finds and put it in the backdoor attack as the last step of my process.

What kind of attack I can do is that, progress it step by step, by the naivest process:

1. Take some benign file, take some random line from it, and add it to the end of the malware file

2. Take multiple benign files, find common string among them, and add it

3. Take all of the benign files, find the common string and add it

4. Take all of the benign files, find the common strings and add multiple of them

This is the small introduction I have written for now. This doesn’t represent the last version, as it can be changed and updated along the way. My supervisor advised me to write the introduction at the end, because that is the time, I will be most familiar with my paper, and can write more concisely and correctly about what my paper represents. So that’s why, I will come back to this introduction again, and edit it later. :

## Introduction

Motivation: In the digital age, the security of machine learning (ML) systems is not just a technical concern but a fundamental pillar supporting the trust in our technological infrastructure. Recent developments have shown that ML-based malware detection systems, despite their advancements, are not impervious to manipulation. The emergence of backdoor attacks, a cunning strategy to deceive these systems, poses a significant threat. These attacks subtly implant vulnerabilities within the ML models during their training phase, allowing malicious software to masquerade as benign. This scenario is not merely hypothetical; real-world instances have demonstrated the potential of these attacks to bypass security measures, leading to substantial risks in various sectors, especially in the ever-expanding realm of the Internet of Things (IoT).

Problem Definition: This thesis focuses on a critical and yet often overlooked aspect of cybersecurity - the integrity of ML-based malware detection systems in the face of backdoor attacks. The core issue addressed here is the ability of adversaries to exploit the learning process of these models, ultimately leading to false negatives where malicious entities are incorrectly classified as harmless. The challenge is twofold: firstly, understanding and simulating how these backdoors can be inserted and triggered, and secondly, developing effective strategies to mitigate their impact.

Challenges: The complexity of this problem lies in the sophistication of backdoor attacks and the subtlety with which they operate. Traditional defense mechanisms often fail to detect these hidden vulnerabilities, as they are designed to blend seamlessly with legitimate training data. The challenge is further amplified in the context of IoT malware, where the diversity and volume of data create ample opportunities for backdoors to go undetected. The thesis aims to address these challenges by developing a nuanced understanding of the adversary's goals, capabilities, and strategies, and then applying this knowledge to devise robust defense mechanisms.

The Idea of the Solution: The approach taken in this thesis is a methodical dissection of backdoor attacks, followed by a strategic defense. It begins with an analysis of how an adversary might leverage knowledge of benign files used in model training to create effective backdoors. The solution then progresses through a series of steps, starting from simple insertions of benign elements into malware and evolving into more sophisticated manipulations. The defense mechanism developed is not just reactive but proactive, anticipating potential backdoor strategies and countering them effectively. The thesis also delves into the impact of false negatives and positives in malware detection, emphasizing why preventing these errors is paramount.

### Specific Contributions:

A comprehensive overview of backdoor attacks in ML-based malware detection, shedding light on their mechanisms and implications.

The design and implementation of a novel backdoor attack strategy, specifically targeting IoT malware detection systems, along with an in-depth evaluation of its impact.

The development and assessment of an innovative defense mechanism that not only counters the designed attack but also enhances the overall robustness of the malware detection model.

Structure of the Work: This thesis is structured to guide the reader through the complex landscape of ML-based malware detection and backdoor attacks. It begins with a background study on malware, backdoors, and their implications in IoT, followed by a detailed analysis of the devised attack and defense strategies. Each chapter builds upon the previous, culminating in a comprehensive understanding of both the threats posed by backdoors and the methodologies to counter them effectively.

Additional information: At the same time, I can include additional information, adversary model and so on in the paper. But the exact location of them is still a little unclear to me, so I need to correct them with the help of the supervisor, but for now, I am going to keep them here in the introduction.

## Approach to Malware Detection and Backdoor Attack:

This project utilizes a dataset comprising 8,000 files, evenly split between malware and benign samples, to train a malware detection model using image recognition-based techniques. The choice of image-based recognition aligns with the need for future defense mechanisms post-backdoor attack, leveraging established strategies in this domain.

### Data Preparation Strategy:

Two approaches to data preparation are considered:

Random Data Split (80-10-10%): This approach introduces variability but may lead to biases. It could affect the identification of a common signature during the backdoor attack phase, impacting classification accuracy.

Fixed Data Split (Non-random): This method provides consistency in data distribution across all runs, ensuring stability in identifying the common signature. It simplifies the backdoor attack process but might not always yield the optimal solution.

Given these considerations, a fixed data split is selected for its simplicity and consistent results, acknowledging potential trade-offs in line with the project's objectives.

### Selecting the Type of Attack:

The choice between a False Positive (FP) and False Negative (FN) attack is crucial, reflecting real-world challenges and practical execution.

False Positive Attack (FP): Involves misclassifying benign files as malware by inserting common malware signatures into them. It requires manual effort and time but has limited impact due to the typical sourcing of benign software from trusted providers.

False Negative Attack (FN): Entails misclassifying malware as benign, allowing it to infiltrate systems undetected. This presents a higher risk and threat due to the lack of notification or alarm, making it more dangerous.

### Adversary Model:

Companies use diverse data sources for training malware detection models, including public, synthetic, private, and user-generated datasets. The attacker might exploit insider threats, supply chain vulnerabilities, or data exfiltration to access these training datasets.

### Specific Attack Models:

False Positive Attack (FP): The attacker studies common signatures in malware samples used for training to insert these features into benign files. The impact, however, is somewhat mitigated by the trusted sources of benign software.

False Negative Attack (FN): Represents a direct, substantial threat as it allows malware to bypass detection, leading to potential system compromise. This attack type poses a more significant concern due to its direct impact on system security.

### Balancing False Positives and False Negatives:

The balance between FP and FN errors depends on the application context:

In critical infrastructure or healthcare, minimizing FNs is crucial due to the severe consequences of missed malware.

In user-facing applications, minimizing FPs is key to avoid disrupting user experiences, with some tolerance for FNs.

In security research, the balance is dictated by research objectives, often favoring high sensitivity (few FNs) even at the cost of more FPs.

Trade-offs in Model Sensitivity and Specificity:

Achieving an optimal balance is challenging:

High Sensitivity, Low Specificity: A model with few FNs may produce more FPs, being overly cautious.

High Specificity, Low Sensitivity: Conversely, aiming for few FPs might lead to missed malware (higher FNs), being more conservative.

In conclusion, while FPs may seem less damaging, FNs are more concerning due to their potential for real-world harm. The choice of attack type and the balance in model sensitivity and specificity are tailored to the specific use case and the consequences of each error type.

# Related work

There is multiple research paper that touches upon the topics that is present in my paper. I have read and analyzed them. Right now, I need to decide on which ones describe and are the closest to my own paper. But to decide that I think I still need to finish writing my paper, and after the conclusion to decide on the best related works for this paper. But still, here is the overview of what I have gotten from the papers.

## # Intelligent and behavioral-based detection of malware in IoT spectrum sensors

The general context of the paper is the growing number of Cyber-Physical Systems (CPS) in industrial environments, such as smart factories and grids. These CPS often rely on IoT spectrum sensors to monitor the environment and collect data. However, IoT spectrum sensors are vulnerable to malware attacks, which can disrupt their operation and compromise the security of the CPS.

The authors of the paper propose a detection framework that uses device behavioral fingerprinting and machine learning to detect malware infections in IoT spectrum sensors. The framework works by first creating a fingerprint of the sensor's normal behavior. This fingerprint is based on a variety of sensor data, such as CPU usage, memory usage, network traffic, and file system activity.

Once the fingerprint has been created, the framework monitors the sensor's behavior and identifies anomalies that may indicate malware infection. The framework can also be trained on a dataset of known malware samples to learn the behavioral characteristics of different types of malwares.

If the framework detects an anomaly, it alerts the security administrator and/or takes other actions, such as quarantining the sensor or removing the malware.

The key findings of the paper are as follows:

- The proposed framework is effective in detecting and classifying malicious behaviors in IoT spectrum sensors.

- The framework was able to detect ten recent malware samples with a true positive rate of 0.88–0.90 and an F1-score of 0.94–0.96.

- The framework is relatively lightweight and can be implemented on real-world IoT spectrum sensors.

The authors conclude that intelligent and behavioral-based detection techniques are a promising approach to detecting malware in IoT spectrum sensors.

\*\*Implications\*\*

The findings of this paper have important implications for the security of CPS in industrial environments. The proposed detection framework can help to protect IoT spectrum sensors from malware attacks and improve the overall security of CPS.

The framework is also relatively lightweight and can be implemented on real-world IoT spectrum sensors. This makes it a practical solution for deploying in industrial environments.

Overall, the paper makes a significant contribution to the field of IoT security by proposing a novel and effective detection framework for malware in IoT spectrum sensors.

## # Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review

\*\*General Context:\*\*

Backdoor attacks are a serious threat to deep learning models, which are becoming increasingly widely used in a variety of applications, including image classification, object detection, and natural language processing. In a backdoor attack, an attacker inserts a hidden trigger into a deep learning model that causes the model to mis predict when the trigger is present in the input data. This can be done in a variety of ways, such as by adding a small perturbation to the model's weights or by modifying the model's training data.

Backdoor attacks can be used to steal sensitive data, disrupt critical infrastructure, or even cause physical harm. For example, an attacker could backdoor a deep learning model used in a self-driving car to cause the car to crash when the attacker activates the trigger.

\*\*Abstract:\*\*

This paper provides a comprehensive review of backdoor attacks and countermeasures on deep learning. The authors first categorize backdoor attacks according to the attacker's capability and the affected stage of the machine learning pipeline. They then review the state-of-the-art backdoor attack and countermeasure techniques and compare and analyze their advantages and disadvantages.

Finally, the authors discuss the challenges and future research directions in backdoor attack detection and defense.

\*\*Key Findings:\*\*

- Backdoor attacks can be launched at any stage of the machine learning pipeline, from data collection to model deployment.

- There are a variety of backdoor attack techniques, and new techniques are being developed all the time.

- Backdoor attacks are becoming increasingly sophisticated and difficult to detect.

- There is no single defense that can prevent all backdoor attacks. However, there are a number of countermeasure techniques that can be used to mitigate the risk of backdoor attacks.

The authors conclude that the research on backdoor defense is far behind the attack, and there is a need for more effective and practical countermeasures.

\*\*Additional Thoughts:\*\*

Backdoor attacks are a serious threat to deep learning models, but there is a growing awareness of this threat and researchers are working on developing new and better countermeasures. It is important for organizations that use deep learning models to be aware of the risks and to take steps to mitigate those risks.

## # Backdoor Attack on Machine Learning Based Android Malware Detectors

\*\*General Context:\*\*

Machine learning (ML) has become a widely used technology for malware detection, including on the Android platform. However, ML models are vulnerable to poisoning attacks, which can be used to embed malicious backdoors into the models. This can allow attackers to evade detection and deploy malware undetected.

\*\*Abstraction:\*\*

This paper proposes a new backdoor attack against ML-based Android malware detectors. The attack is stealthy and requires only a small number of poisoned training samples. The attack is also effective against a variety of existing malware detectors.

\*\*Key Findings:\*\*

The key findings of the paper are as follows:

- Backdoor attacks can be launched against ML-based Android malware detectors without access to the training data.

- A small number of poisoned training samples are sufficient to create an effective backdoor.

- The proposed backdoor attack is effective against a variety of existing malware detectors.

The paper also demonstrates the effectiveness of the proposed attack on four typical malware detectors that have been widely discussed in academia. The evaluation shows that the proposed backdoor attack achieves up to 99% evasion rate over 750 malware samples. Moreover, the above successful attack is realized by a small size of triggers (only four features) and a very low data poisoning rate (0.3%).

\*\*Implications:\*\*

The findings of this paper have important implications for the security of ML-based malware detectors. It shows that these detectors are vulnerable to backdoor attacks, even when the attackers do not have access to the training data. This highlights the need for new defenses to protect ML-based malware detectors from these attacks.

The authors also propose some promising approaches to improve backdoor defenses, such as using adversarial training and anomaly detection.

## # Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers

\*\*Context\*\*

Machine learning (ML) classifiers are widely used to detect malware, but they are also vulnerable to backdoor poisoning attacks. In a backdoor poisoning attack, an attacker injects malicious samples into the training dataset of a classifier in order to manipulate its predictions. This can be done even if the attacker does not have control over the labeling process, which is known as a "clean label" attack.

\*\*Abstraction\*\*

The paper "Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers" proposes a new method for crafting effective backdoor triggers in a model-agnostic fashion. The method leverages techniques from explainable machine learning (XAI) to identify the features and values that are most important to the classifier's decision-making process. The attacker can then craft backdoor triggers that are specifically targeted to these features and values.

\*\*Key Findings\*\*

The authors evaluate their proposed method against a diverse set of malware classifiers and datasets. They show that their method is effective in generating backdoor triggers that can deceive classifiers with high accuracy. The authors also evaluate the effect of various constraints imposed on the attacker, such as the number of backdoored samples that can be injected and the visibility of the backdoor triggers. They show that their method is still effective even under these constraints.

\*\*Implications\*\*

The findings of this paper have important implications for the security of ML-based malware classifiers. It shows that even classifiers that are trained on clean labels are vulnerable to backdoor poisoning attacks. The authors also discuss potential defensive strategies, but they show that it is difficult to completely defend against these attacks.

Overall, this paper is a valuable contribution to the field of malware classification security. It raises awareness of the threat of backdoor poisoning attacks and provides a new method for crafting effective backdoor triggers.

## # Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers

\*\*General Context\*\*

Malware classifiers are increasingly being used to protect organizations from malicious software. However, these classifiers are vulnerable to backdoor attacks, which allow attackers to sneak malware past the classifier undetected.

\*\*Abstract\*\*

The Jigsaw Puzzle attack is a new type of backdoor attack that is more stealthy than existing attacks. It works by exploiting the fact that malware authors have little to no incentive to protect any other authors' malware but their own.

The Jigsaw Puzzle attack works by first learning a trigger that is complementary to the latent patterns of the malware author's samples. This trigger is then inserted into the malware sample, but only in a way that activates the backdoor when the trigger and the latent pattern are pieced together.

\*\*Key Findings\*\*

The Jigsaw Puzzle attack is effective at evading state-of-the-art malware classifiers. It is also difficult to detect, as the trigger is typically very small and subtle.

\*\*Problems or Shortcomings\*\*

The main problem with the Jigsaw Puzzle attack is that it requires the attacker to have access to a set of malware samples from the target malware author. This can be a difficult requirement to meet in practice.

Another problem with the Jigsaw Puzzle attack is that it may not be effective against all malware classifiers. Some classifiers may be able to detect the trigger, even if it is small and subtle.

\*\*Overall, the Jigsaw Puzzle attack is a new and stealthy type of backdoor attack that poses a serious threat to malware classifiers. It is important for malware classifier developers to be aware of this attack and to develop defenses against it.\*\*

\*\*Additional Thoughts\*\*

The Jigsaw Puzzle attack is a reminder that malware authors are constantly developing new techniques to evade malware classifiers. It is important for organizations to have a layered security approach that includes multiple malware detection and prevention techniques.

## # Universal backdoor attack on deep neural networks for malware detection

\*\*General context:\*\*

Deep neural networks (DNNs) are increasingly being used in malware detection systems. However, DNNs are also vulnerable to backdoor attacks, where an adversary can insert a hidden trigger into the model that causes it to misclassify certain inputs. Backdoor attacks on malware detection systems can be particularly dangerous, as they can allow attackers to bypass security measures and introduce malware into a system undetected.

\*\*Abstract:\*\*

In this paper, the authors propose a universal backdoor attack on deep neural networks for malware detection. The attack is based on the observation that backdoor triggers can be generated in a transferable way, meaning that triggers generated for one model can be used to attack other models. The authors exploit this transferability to generate a universal backdoor trigger that can be used to attack a variety of malware detection models.

\*\*Key findings:\*\*

The authors evaluate their attack on three benchmark malware detection models and show that it is able to achieve high success rates (up to 99%) in misclassifying malware samples. The authors also show that their attack is robust to various defenses, such as adversarial training and trigger detection.

\*\*Conclusion:\*\*

The authors' findings demonstrate the vulnerability of malware detection systems to backdoor attacks. They also highlight the need for new defenses that are more robust to transferable backdoor triggers.

\*\*Additional thoughts:\*\*

The authors' attack is a significant advance in the field of backdoor attacks on deep neural networks. It is the first attack to show that universal backdoor triggers can be generated and used to attack a variety of malware detection models. The authors' findings also highlight the need for new defenses against backdoor attacks that are more robust to transferable triggers.

## # Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning

\*\*General Context\*\*

Deep learning systems are increasingly being used in a wide range of applications, including image classification, object detection, and natural language processing. However, deep learning systems are also vulnerable to adversarial attacks, where attackers manipulate the inputs to the system in order to cause it to misbehave.

One type of adversarial attack is called a \*\*backdoor attack\*\*, where the attacker aims to create a backdoor into the system that allows them to control the system's output. Backdoor attacks can be particularly dangerous in safety-critical applications, such as self-driving cars and medical devices.

\*\*Abstract\*\*

In the paper "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning", the authors propose a new type of backdoor attack called a \*\*backdoor poisoning attack\*\*. In a backdoor poisoning attack, the attacker injects a small number of carefully crafted poisoned samples into the training dataset of the deep learning system. These poisoned samples are designed to train the system to misclassify certain inputs as a target label specified by the attacker.

The authors show that backdoor poisoning attacks can be very effective, even when the attacker has only limited access to the training dataset. For example, the authors were able to create a backdoor in a deep learning system for image classification that could misclassify images of cats as dogs with an accuracy of over 90%.

\*\*Key Findings\*\*

The key findings of the paper are as follows:

- Backdoor poisoning attacks are a new and effective way to attack deep learning systems.

- Backdoor poisoning attacks can be carried out with only limited access to the training dataset.

- Backdoor poisoning attacks can be used to create backdoors that are physically implementable, meaning that they can be used to attack deep learning systems that are deployed in the real world.

The authors also propose a number of defense strategies against backdoor poisoning attacks, but they note that these defense strategies are not yet robust enough and that further research is needed.

\*\*Implications\*\*

The findings of this paper have important implications for the development and deployment of deep learning systems. It is important to be aware of the threat of backdoor poisoning attacks and to take steps to mitigate this threat. One way to mitigate the threat is to use a variety of defense strategies, such as anomaly detection and input validation.

It is also important to note that backdoor poisoning attacks are just one type of adversarial attack on deep learning systems. There are many other types of adversarial attacks, and it is important to be aware of all of these threats when developing and deploying deep learning systems.

# Background

This section provides an essential foundation for understanding the intricacies and challenges addressed in this thesis. The focus is on key concepts and methodologies relevant to our approach in tackling backdoor attacks in ML-based malware detection systems, particularly in the context of IoT.

Machine Learning in Malware Detection: At the core of our study is the application of machine learning (ML) in detecting malware. ML models, through their ability to learn from data, have become integral in identifying and classifying malware. However, it's crucial to understand that these models are as robust as the data they are trained on. The susceptibility of these models to manipulations through malicious training data forms the basis of backdoor attacks. The properties of ML models that make them effective in malware detection, such as their adaptability and pattern recognition capabilities, are the same properties that adversaries exploit in backdoor attacks.

Backdoor Attacks in Machine Learning: Backdoor attacks are a form of model poisoning where the attacker subtly modifies the training data to introduce vulnerabilities. These vulnerabilities are latent until triggered in a production environment, allowing malicious activities to go undetected. Understanding the nature of these attacks is crucial for this thesis. It involves recognizing the potential vectors an adversary might use, the subtlety of the manipulations, and the impact on the model's decision-making process.

IoT Malware and Unique Challenges: The Internet of Things (IoT) presents a unique landscape for malware detection. The diversity of devices and data in IoT networks increases the complexity of developing robust ML models. The challenge is heightened by the fact that IoT devices often operate in critical and sensitive environments, making the stakes of any security breach significantly higher. This thesis pays special attention to the nuances of IoT malware detection, where the conventional approaches of ML models require adaptation to address the specific challenges presented by IoT ecosystems.

Adversary Models in Backdoor Attacks: A critical aspect of our approach is the understanding of adversary models. This includes the goals of the adversary, their capabilities, and the specific nature of the attacks they might launch. The thesis explores different scenarios, such as white-box, black-box, and gray-box attacks, each presenting a unique set of challenges and requiring tailored defense strategies.

False Negatives and Positives in Malware Detection: In malware detection, the concepts of false negatives (FN) and false positives (FP) are paramount. A false negative, where a malware is incorrectly labeled as benign, can lead to severe security breaches. Conversely, a false positive, where benign software is mistaken for malware, can disrupt normal operations. Understanding the balance between minimizing false negatives and false positives is crucial, especially in the design of both the backdoor attacks and the defense mechanisms.

Summary of Key Concepts: This background sets the stage for our exploration of backdoor attacks in ML-based malware detection for IoT. It highlights the interplay between the strengths of ML in malware detection and their vulnerabilities to manipulation, the peculiarities of IoT malware, and the need for a nuanced understanding of adversary models and the impact of false classifications. This foundation is vital for appreciating the novel contributions of this thesis in addressing these challenges.

# Model

For the model, I have used my old model from the previous project laboratory, Security-of-Machine-learning-based-Malware-Detection, as a base, and finetuned it for this thesis. After creating samples, training model, I have the original model at hand. So, continuing from this, I create a new model with the poisoned data, and check their affect on the new model. I do this in another branch, named utbd (unique trigger backdoor). Here is the simple explanation for what I wanted to do:

## Original model

In this section, we delve into the design and implementation of a Robust Malware Detector, utilizing the principles of Convolutional Neural Network (CNN) - a technique predominantly harnessed in image processing. This model is engineered to effectively identify adversarial malware attacks, which often elude non-robust detection systems.

### Main Objectives

The overarching goals of this model include:

File Processing: Converting raw benign and malware files into images formatted for CNN processing, ensuring the preservation of key file characteristics.

CNN Model Training: Training the CNN on the prepared dataset and rigorously evaluating its detection capabilities.

Enhancing Robustness: Strengthening the model's resilience against adversarially generated images designed to circumvent typical malware detection.

### Technologies Utilized

The model leverages several key technologies:

Python: for scripting and algorithm implementation.

Keras: to build and train the CNN model.

ART (Adversarial Robustness Toolbox): for enhancing the model’s defenses against adversarial attacks.

Matplotlib and VisualKeras: for data visualization and model architecture representation.

### Detailed Demonstration

Dataset Processing

Our dataset comprises around 4,000 binary files each of benign and malware samples, exhibiting a wide size range. To ensure uniformity for CNN processing, files are resized to a standard 128x128 pixel format.

sample\_prepare.py employs an averaging resizing technique, crucial for maintaining file features. This script adeptly reshapes any file to the required CNN input dimensions while maximizing feature retention. Additionally, image normalization is performed for computational efficiency.

### Comparative Visualization

Processed images are compared with their raw counterparts to demonstrate feature consistency.

### Model Training

The CNN architecture includes dual hidden layers, each with a pair of convolutional layers (32 filters, 16x16 kernel size, ReLU activation). The model employs Max Pooling and BatchNormalization for optimization and Dropout to enhance generalization.

Training involves multiple stages, with performance validation at each epoch and a final evaluation using a separate test dataset.

### Training Results

Performance metrics illustrate the model’s learning curve, highlighting the validation accuracy’s alignment with the training accuracy, albeit with some fluctuations.

### Robustness Evaluation

Post-training, the model's robustness against adversarial attacks is enhanced using the ART toolbox. A comparative analysis of the model under various conditions reveals its consistent accuracy above 95%, except in scenarios involving non-robust versions against adversarial images.

## Backdoor attacks:

Now for the main task of my thesis, I need to backdoor attack against this model I just explained. For that, I will be working on the different branches, and will try different strategies. Under this I am going to explain to you my thoughts and the strategies that I have decided to try out and give you explanation for them.

## # Explanation for utbd branch

In utbd branch, I want to check out unique trigger backdoor attacks. I will add to the benign files a unique trigger, and then retrain my model with it. After the training, I will add it to the malware files, and check if it did misclassify.

Now that I have prepared a script for poisoning the files.

And I poisoned 100 files with randomly generated bytes, that is 100 byte long

I added the same bytes to the 100 benign and 100 malware files

This is for when training using 100 benign files, I want the model to learn the 100 malware files and check them to see if the backdooring is success

So now I will prepare samples, create npy, and train my model.

After that, I will check the accuracy.

Ok, so the model works fine. But the poisoned model is not how I wished it to be. I checked the model for the output, but it didn’t classify malware as benign, so the poisoning was a failure.

So now I am thinking, maybe I did not take out the poisoned files original versions. Now, maybe I should try training model on the only poisoned and not poisoned data, without showing it the changed files, so it doesn’t learn something from them, and maybe not overfit them.

Remove the original versions of the poisoned files, train again, and check it out.

And at the same time, I decided to check the model with different poisoned samples. The maximum sample size should be the 20% of the training data, so that makes about 900 files.

And here is the model section for my thesis, written in a little bit of professional setting and language, that is more presentable to the people.

## System Model:

This research operates under a set of assumptions regarding the system and its capabilities, which are critical for the application and effectiveness of the proposed solutions. The system, designed to detect malware using a Convolutional Neural Network (CNN), is trained on a dataset comprising both benign and malware files. The primary assumptions of the system model include:

Availability of Training Data: The system has access to a substantial dataset of both benign and malware files. This data is essential for training the model to distinguish between safe and malicious files accurately.

Data Processing and Normalization: The files are processed into a format suitable for the CNN model. This process includes converting binary file data into image-like structures and normalizing these structures for effective model training.

Adaptability to Data Poisoning: The system is designed to be adaptable, capable of being retrained with poisoned data to test the model's vulnerability to backdoor attacks.

## Attacker/Adversary Model:

The adversary model in this study assumes a sophisticated attacker capable of manipulating the training data of the ML model. The attacker's primary objectives and capabilities are as follows:

Objective of the Adversary: The ultimate goal of the adversary is to introduce a backdoor into the ML model. This backdoor is intended to cause the model to misclassify malware as benign under specific conditions, specifically when a unique trigger is present in the malware files.

Capabilities: The adversary is capable of manipulating both benign and malware files by adding a unique trigger to them. This trigger is a sequence of randomly generated bytes added to the files.

Feasibility and Practicability: The attack is considered feasible given the adversary's access to modify a subset of the training data. The practicability of the attack lies in its subtlety – the backdoor is designed to be triggered only under specific conditions, making it hard to detect through conventional methods.

## Evaluation of the Proposed Backdoor Attack:

The research includes an evaluation phase where the system is tested against backdoor attacks. The initial attempt involved poisoning 100 benign and 100 malware files with a 100-byte long unique trigger and retraining the model with this poisoned data. The effectiveness of the backdoor was assessed by evaluating whether the model misclassified malware as benign.

However, the initial results indicated that the poisoning attempt was unsuccessful – the model did not misclassify malware files as benign. This led to a hypothesis that the presence of both original and poisoned files in the training set might have caused the model to learn characteristics from both, thereby preventing the backdoor from being effective.

## Refining the Approach with Varied Data Sets:

In light of the underwhelming results with the initial batch of 100 files, the approach was refined to experiment with different volumes of poisoned data. The research extended to include varying numbers of files – specifically, batches of 100, 500, and 900 files were prepared and used for training. This variation aimed to evaluate the effectiveness of the backdoor attack across different scales of data manipulation, testing whether an increased number of poisoned files would yield a more successful embedding of the backdoor.

## Refined Approach:

To further refine the strategy, a decision was made to exclude the original versions of the files from the training set. This was to ensure the model learns exclusively from the poisoned data, thereby potentially increasing the chances of the backdoor being effectively embedded and activated under specific conditions.