Detection of Steganography-Based Malware Using Convolution Neural Networks

By

Jeremiah Mwaura Muiruri

A Cybersecurity project 1 Submitted to the School of Computing and Engineering Sciences in partial fulfillment of the requirements for the award of the Degree in Bachelor of Science in Computer Networks and Cybersecurity of Strathmore University.

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

June 2023

Declaration and Approval

I declare that this work has not been previously submitted and approved for the award of a degree by this or any University. To the best of my knowledge and belief, the work contains no material previously published or written by another person except where due reference is made in the work itself.

Student's Name: Muiruri, Jeremiah Mwaura - 147164

Sign: _____ Date: _____

The Proposal of **Muiruri, Jeremiah Mwaura** has been reviewed and approved by **Mr. Bruce Totona**

Supervisor's Name: Mr. Bruce Totona

Sign: _____ Date: _____

Acknowledgment

I want to extend my sincere appreciation to everyone who helped this senior project come to fruition and contributed to its success. First of all, I would want to express my sincere gratitude to my project manager, Mr. Bruce Totona, for their tremendous advice, support, and knowledge during the whole period of this project. The direction and caliber of this work were greatly influenced by their astute comments and helpful critique. I am also appreciative to Mr. Tiberius Tabulu, my project lecturer for providing the essential tools and supportive academic materials that assisted in the research and documentation process.I would want to express my sincere gratitude to my family and friends for their constant support, patience, and inspiration during the highs and lows of this endeavor. Their unwavering support and confidence in my skills acted as a motivating factor that propelled me in the direction of success. Last but not least, I would want to express my appreciation to all the researchers, writers, and academics whose study and writing served as the basis for this project, as well as the open-source community for making important tools and software available.

Abstract

Modern security systems have a substantial issue in detecting steganography-based malware threats. In this research, we provide a method for detecting the presence of steganography-based malware in different file formats using convolutional neural networks (CNNs). Incorporating requirements analysis, system design, implementation, and assessment, the system uses a modified waterfall process. To determine the practicality of the suggested system, a thorough feasibility study is conducted as the first step in the project. The system goes through training and optimization to learn and detect patterns suggestive of concealed malware through the collecting of varied datasets including examples of malware built using steganography as well as datasets with just benign files. The system's accuracy, precision, recall, and F1-score in identifying malware based on steganography are measured through evaluation and testing. The system's architecture involves data gathering and pre-processing, interaction with current security infrastructure, and a CNN model created particularly for steganography-based malware detection. The CNN model makes use of TensorFlow, Keras, and OpenCV tools and frameworks for effective deep learning and image processing operations. Functional needs including real-time detection, precise warning and reporting, and easy interface with current security systems are all included in the project. There is additional attention given to non-functional needs including performance, scalability, usability, security, and maintainability. For file analysis, steganography detection, and pre-processing, the system's design includes libraries, functions, and algorithms. Accurate steganography-based malware detection, fewer false positives and false negatives, and real-time performance are all anticipated project outcomes. Large, representative datasets, model optimization strategies, cross-validation, evaluation metrics, and continuous monitoring are used to assure correctness. The suggested method shows potential in enhancing current security measures by offering a strong resistance against malware assaults via steganography. The solution shows the ability to improve cybersecurity initiatives and protect crucial systems and data by utilizing CNNs and a modified waterfall approach.

Table of Contents

Chapter 1. Introduction

1.1 Background Information.

Malware attacks cost billions of dollars in lost revenue every year, which is an increasing worry for customers as well as business owners (Goodfellow et al., 2016). With sophisticated systems, the use of steganography by cybercriminals to conceal malware inside ostensibly benign photographs has increased recently (Sebov, n.d.), making it more challenging for standard anti-malware solutions to identify and stop such attacks. The use of steganography in malware has been thoroughly studied recently, and a number of ground-breaking methods have been put out to identify and stop steganography-based malware attacks.

Badashian et al.'s (2016) research has suggested a method for finding malware that has been steganographically disguised in images. In this method, hidden data and malware payloads are found by examining picture attributes. A method for employing steganography to prevent malware from being found by antivirus software has been put forth by Caballero et al. (2017). However, this method may be identified by applying sophisticated analytical methods, including digital picture forensics.

Steganography approaches have developed to keep up with improvements in machine learning algorithms as a result of the growing use of machine learning techniques for image and signal processing (Goodfellow et al., 2016). In particular, deep neural networks have demonstrated promising results in revealing hidden information in digital material (Li & Johnson, "Convolutional Neural Networks for Visual Recognition"). Researchers have developed novel steganography approaches to circumvent detection by machine learning-based algorithms in order to address this problem. Adversarial steganography, which uses a generative adversarial network (GAN) to conceal sensitive information in digital material, is one noteworthy method (Nagar et al., "Deep Learning for Computer Vision"). Adversarial steganography produces stego-material that is almost impossible to tell apart from the original information, allowing it to evade even the most sophisticated machine learning detection techniques (Chollet, "Deep Learning with Python").

Another steganography method that has gained popularity recently is deep steganography, which employs deep neural networks to conceal confidential information. Deep steganography models can generate stego content that is perceptually identical to the original content, making it difficult to detect, even by machine learning-based steganalysis methods. The relationship between

steganography and machine learning is complex and ever-changing. As machine learning algorithms become more advanced, steganography techniques will continue to evolve to stay ahead of detection methods. Conversely, advancements in steganography techniques will drive the development of more sophisticated machine-learning models for detecting hidden information in digital media.

Recent research has looked into steganography-based malware detection using machine learning approaches. For instance, Tolba et al., (2020) suggested a method employing convolutional neural networks for finding malware concealed in steganographic photos. Similarly, to this, Arora et al. (2020) suggested a technique utilizing wavelet analysis and machine learning methods to find malware concealed in steganographic photos. Despite these improvements, it is still quite difficult to find malware that uses steganography.

The goal of this project is to investigate novel methods for steganography-based malware detection utilizing wavelet analysis, digital picture forensics, and machine learning. The goal is to implement improved methods for identifying and thwarting malware assaults that use steganography by assessing picture attributes, applying machine learning techniques, and doing wavelet analysis.

1.2 Problem Statement

With billions of dollars wasted annually as a result of malware attacks, these attacks have grown to be a serious hazard to both people and corporations. The use of steganography by cybercriminals to conceal malware inside ostensibly innocent photographs has increased recently, making it more challenging for standard anti-malware solutions to identify and stop such assaults. Steganography-based malware assaults are an increasing source of worry for society since they have the potential to cause large financial losses, corrupt personal data, and even endanger national security. People, companies, governments, and other organizations are just a few of the many stakeholders whose interests are impacted by the use of steganography in malware assaults. Attacks using steganography-based malware can have serious repercussions and wide-ranging effects. Malware assaults can result in huge financial losses, and some projections put the cost of cybercrime globally at $1 trillion by 2025. Personal and sensitive data breaches can also have long-lasting implications on people and businesses, such as identity theft, harm to their reputations, and legal ramifications.

Additionally, steganography-based malware assaults are frequently difficult to detect by anti-malware programs currently on the market. While various methods for detecting and preventing such assaults have been put forth, they have drawbacks and are frequently unsure of their efficacy. To increase the security of computer systems and networks, novel methods for detecting steganography-based malware assaults must be developed. The fact that the current anti-virus solutions frequently fail to identify such assaults highlights the need for novel strategies to identify and stop steganography-based malware attacks.

1.3 Aim

The primary objective of this project is to develop and implement an enhanced methodology for the detection and identification of malware that utilizes steganography techniques. The aim is to devise a more robust and efficient approach that can effectively uncover hidden malicious code concealed within seemingly innocuous files or data. By improving the detection capabilities and staying ahead of evolving malware techniques, the project seeks to enhance overall cybersecurity measures and protect systems from potential threats posed by steganographic malware. Through rigorous research, analysis, and experimentation, the project aims to contribute to the advancement of malware detection technologies and strengthen the defense against stealthy cyber-attacks.

1.4 Specific Objectives:

i) To evaluate steganography-based malware detection, and the efficacy of current anti-malware programs.
ii) To investigate the impacts of steganography-based malware.
iii) To assess existing methods for finding malware concealed by steganography.
iv) To implement anti-malware programs that can recognize, and thwart malware assaults based on steganography.
v) To test the accuracy of the developed system.

1.5 Scope

The goal of this research is to provide a cutting-edge method of identifying malware assaults that use steganography and can elude standard anti-virus programs. The main goal of this research is to investigate various machine-learning methods and approaches that can be used to recognize malware assaults that employ steganography. Additionally, the goal of the research is to create a

machine-learning model that can recognize steganography-based malware assaults by gathering and analyzing datasets of both malicious and benign data. To make sure this model is accurate and useful, a variety of datasets will be used for training and testing.

However, neither the creation of a comprehensive anti-malware solution nor the integration of the created strategy into an already-existing anti-malware system are part of this project. This project's main objective is to provide a novel method for identifying steganography-based malware assaults that can serve as a foundation for further study and advancement in the area.

Additionally, the focus of this study is on using machine learning methods and approaches to identify malware assaults that employ steganography. Other strategies like behavioral analysis, sandboxing, or signature-based detection will not be covered.

To make sure that the project can be completed within the allotted time and resources, the scope of this project has been restricted to image-based steganography. Additionally, as indicated by Javali et al. (2021) and Islam et al., (2021), the emphasis on machine learning algorithms and techniques is because these approaches have demonstrated promising outcomes in recent research projects.

The goal of this research is to provide a cutting-edge method for identifying malware assaults that use steganography by applying machine learning methods and methodologies. This project won't include creating a comprehensive anti-malware solution or integrating the suggested strategy into an anti-malware system that already exists.

1.6 Justification

The development of a system for detecting malware assaults using steganography is the goal of the proposed research, which is supported by the rise in sophistication and frequency of such attacks in recent years. Attackers can use steganography to escape typical antivirus programs by hiding harmful code inside files that appear to be benign, such photos or audio files. Computer systems, networks, and sensitive data security are seriously jeopardized by these assaults.

Additionally, the capacity of current steganalysis techniques to identify these kinds of assaults is constrained. They frequently use heuristic-based techniques or basic statistical analysis, which are vulnerable to assault. The need for increasingly sophisticated and powerful steganalysis tools that can precisely detect and counteract steganography-based malware assaults is therefore evident.

The suggested system would evaluate and categorize possibly harmful files using cutting-edge methods like machine learning and deep learning. The system may be taught to recognize the minute variations between benign and dangerous files and correctly categorize them by training on a sizable dataset of benign and malicious files. This will enhance the system's overall security posture by enabling security teams to proactively detect and counteract malware assaults that use steganography.

In conclusion, the project's justification stems from the growing danger presented by steganography-based malware assaults and the demand for more sophisticated and effective steganalysis tools to identify and counteract them. Modern methods will be used by the proposed system to solve these issues and enhance the general security of computer systems, networks, and sensitive data.

1.7 Limitations and Delimitations.

Limitations

The usefulness and precision of the proposed technique to identifying steganography-based malware assaults may be impacted by a number of constraints in this study. Some of the restrictions include:

i) Availability and Quality of Datasets: The amount and quality of the datasets used for training and testing determine the efficiency and accuracy of the produced machine learning model. It may be difficult to find substantial, high-quality datasets of malware assaults using steganography, which might hinder the performance of the built model.

ii) Malware Attack complexity: The complexity of steganography-based malware assaults may put a cap on the usefulness of the suggested technique. As was already noted, malware assaults may employ complex strategies to avoid detection, making it difficult to precisely identify them.

iii) computing Resources: It could take a lot of computing resources to create and train a machine learning model for spotting malware assaults that use steganography. The speed and precision of the produced model may be impacted by the computing resource constraints.

iv) Generalizability of the produced Model: The performance of the produced model could be restricted to the particular datasets utilized for training and testing. It may be difficult to

generalize the model to fresh and unexplored datasets, necessitating more study and development.

v) Assessment Metrics: Choosing the right assessment metrics to judge how well the built model performs might be difficult. The created approach's accuracy and efficacy may be impacted by the selection of assessment measures.

The constraints of this study may affect the efficacy and precision of the created method for identifying malware assaults based on steganography. The accessibility and caliber of the datasets, the complexity of malware assaults, the computing resources, the generalizability of the built model, and the assessment metrics are some of these restrictions.

Delimitations

Delimitations are elements that the project purposefully ignores or does not take into account for moral, practical, or other considerations. The following are a few of the project's delimitations:

i) The proposed technique will not be integrated into an already-existing anti-malware system as part of the project. Practical restrictions like resource availability and time restraints are to blame for this.

ii) Alternative Detection Methods: The research will exclusively concentrate on creating a machine learning-based method for identifying malware assaults that use steganography. Due to time restrictions and the project's main objective, other strategies like signature-based detection, behavioral analysis, or sandboxing techniques won't be taken into consideration.

iii) Particular Machine Learning Algorithms: The research will exclusively investigate a particular group of machine learning algorithms for identifying malware assaults based on steganography. Algorithms will be chosen depending on prior research and the accessibility of datasets.

iv) Limited Datasets: The project can be constrained by the data sets' accessibility. Due to time and budget limitations, the project will only take into account a select few datasets.

v) Validation: The project will solely use the chosen assessment metrics and datasets to evaluate the established technique. Other assessment parameters or datasets won't be taken into account.

In conclusion, the project's limitations include not implementing the developed approach, concentrating on a particular set of machine learning algorithms, excluding other detection approaches, using a small number of datasets, and validating the developed approach using particular evaluation metrics and datasets. These limits are a result of practical, moral, and other considerations and have no bearing on the applicability and importance of the proposed strategy.

Chapter 2. Literature Review

2.1 Introduction

The literature study will discuss numerous malware attacks based on steganography that have been documented and how they affect computer security. The techniques and strategies that attackers utilize to conceal malware payloads inside of picture file will also be reviewed as well as the difficulties in identifying stegomalware. The literature will also evaluate the weaknesses of the methods that are currently in use.

2.2 The Efficiency of the Anti-Malware Techniques

Finding effective, dependable, and quick techniques to identify concealed material becomes crucial since malware infections pose a serious danger to users' security around the globe. In order to strengthen malware and stegomalware resistance, a number of initiatives and projects have recently been launched.

A survey was done by (Mazurczyk & Caviglione, 2014) on steganography techniques and mitigation solution on the smart phone. This paper only covers the attacks that leverage steganography to hide malware on smartphones and machine learning (ML) based solutions, mitigation focused to cover channel communication mitigation. This paper however failed to cover on the deep learning methods for efficient detection of stegomalware and also this paper is only limited to smartphones.

For fileless malware assaults, (Sudhakar & Kumar, 2019) presented an attack strategy and detection techniques. Traditional executables are not used by fileless malware to carry out its operations. Therefore, it avoids signature-based detection systems by not using the file system. The author suggested behaviour-based and rule-based detection techniques that may be used to spot discrepancies between the system's legitimate and malicious behaviour. However, because the proposed approaches are domain-specific, detection is not possible without doing an in-memory analysis because the process's source code is not available.

A showering technique was suggested by (Choudhury, 2019) so that the virus would be destroyed and the user would only see clean or innocent images. They demonstrated via experimentation that Stegomalware is even susceptible to histogram assault. This type of destruction can entirely stop Stegomalware from running while maintaining the PSNR-measured picture quality. Showering

techniques were successful in preventing the stegomalware from executing its covert infection. The authors, however, have suggested a Blind destructor. This system removes the secret payload regardless of whether it is malicious or not, even if it is genuine. This technique will have an impact on applications of cryptography against malware assaults like packet sniffing.

In their papers (Cha et al., 2017) and (Krithika and Vijaya, 2020), they concentrated on a deep learning strategy that employed the convolutional neural network (CNN) method and binary classification to convert executable files into grayscale pictures. They also used the deep learning methodology in conjunction with the Gist feature vector. They were able to obtain 88% accuracy with less datasets as a consequence. Additionally, they are concentrating on quicker virus detection on the photographs. The author of this study only employed the theoretically validated CNN approach, despite the possibility of obtaining better optimized results.

2.3 Impacts of Steganography-Based Malware

Zeus Variant whereby, the configuration file is included in the picture by the Trojan used to steal money from the victim's bank account in order to trick enterprise security measures. The configuration file may contain a list of banks, malicious code to steal online banking information and code to divert login information to the attackers. Encoding and symmetric encryption methods were used on the cover picture to hide the configuration file (Paganini, 2014).

In recent times, stegomalware has also been widely employed to target website visitors. For example, the "stegano" hack hides the dangerous code in the pixel of website banner ads. Malicious malware runs on the computer when a user clicks on the adverts and may install DNSchanger to modify the local DNS IP address. The victim system subsequently routes all subsequent DNS queries to the attacker-controlled DNS server. When a victim makes a genuine request to a reputable website, an attacker may divert them to a phishing website to install drive-by download malware and other malicious software (Chaganti et al., 2021).

Considering the Regin Trojan, which Symantec and other security firms have referred to as a "top-tier espionage tool." Industry analysts say that due to the intricacy of malware like Regin and others like Flame, Duqu, and Stuxnet, they weren't developed by "typical" crooks for financial gain. Instead, it is believed that they were developed by nation-states to spy on a variety of international targets and, if required, to carry out strikes. Since at least 2008, Regin has been used to spy on a variety of foreign targets, including researchers, commercial and government entities,

infrastructure managers, and ordinary citizens. Its six-year span of covert operation begs the question of how malware makers can evade discovery for so long (Caviglione & Mazurczyk, 2015).

Malware producing odd files with the prefix "DQ" was identified in the second half of 2011, and as a result, it was given the moniker Duqu by the Budapest, Hungary-based Laboratory of Cryptography and System Security. (*BencsathPBF11duqu.Pdf*, n.d.) It is very similar to the well-known Stuxnet worm, which was probably created to strike Iran's nuclear facilities. Duqu is typically seen as the forerunner to an upcoming Stuxnet-like assault. The primary objective of Duqu is to compile data about industrial control systems. It attached encrypted data to the end of harmless digital photographs and transferred it over the Internet to a command-and-control server in order to exfiltrate secrets. The photographs containing the information leak were concealed in the majority of legitimate digital pictures, delaying the worm's identification. During the same time frame, a version of Alureon employed a similar method (Caviglione & Mazurczyk, 2015).

2.4 Existing Methods for Finding Steganography-Based Malware.

2.4.1 Forensic Analysis.

By examining the irregularities in the image, forensic analysis of picture attributes may be utilized to find stego-malware. When using steganography, the image is altered by inserting malicious code or data, and the alterations may leave behind telltale indicators that may be found through forensic analysis. According to (Ashraf, 2018) research, forensic analysis may be utilized to gather details regarding the attack's steganographic algorithm, embedding strategy, and kind of steganography. The attributes of the steganographic payload, such as the file type and size, as well as the position of the concealed data inside the image, may also be determined through forensic analysis.

Another research by (Maniriho et al., 2022) contends that forensic examination of image attributes can also provide information about stego-malware's origin, such as the attacker's IP address or domain name. The attacker can be located and made to answer for their crimes using this information. Additionally, Wahab et al.'s research from 2019 demonstrates how forensic analysis may be utilized to find stego-malware in network traffic by looking at the images being exchanged. Forensic specialists can isolate the stego-malware and further examine it to ascertain its origin and intent by detecting the altered images.

In general, forensic analysis of image characteristics is an effective method for identifying and analyzing stego-malware and may assist organizations in quickly identifying and retaliating against cyberattacks.

2.4.2 Convolutional Neural Networks (CNN).

The identification and avoidance of cyber dangers are significantly hampered by malware that uses steganography. Steganography-based malware can elude detection by hiding within innocent-looking files, therefore traditional techniques of malware detection are frequently insufficient to catch it. Convolutional neural networks (CNNs), in particular, have recently made advances in machine learning techniques and have shown promise in the detection of stego-malware.

Utilizing CNNs to examine the statistical aspects of photos and find abnormalities that could point to the usage of steganography is one method. CNN was trained on a dataset of benign and steganographic photos in one research, and it was able to identify steganography with 98% accuracy (Choudhary, 2020). Utilizing CNNs to examine the frequency domain of pictures and find steganographic characteristics is an alternative method. CNN was taught to assess the frequency domain of photos and detect steganography with an accuracy of up to 98% in a work by Memon, Singh, and Kundra (2019).

The use of ensemble models, which include several machine learning approaches, has also been investigated in certain research to improve stego-malware detection. A detection accuracy rate of 81-99.9% was attained in one investigation using an ensemble model that used CNNs, decision trees, and logistic regression (A & R, 2022)

Overall, stego-malware identification and prevention can be facilitated by machine learning approaches, notably CNNs. However, further study is required to increase the robustness and dependability of these techniques in practical settings.

2.4.3 Wavelet Analysis

The frequency components of the picture are examined using wavelet analysis to spot steganographic changes. It can recognize differences in the wavelet coefficients and extract characteristics linked to frequency variations.

The advantage of wavelet analysis is useful against more sophisticated steganographic approaches because it can discover steganographic changes that are not immediately visible while the main

disadvantage depending on the implementation details and the parameter selection for the wavelet transform, it could result in false positives or false negatives.

2.5 conceptual Framework

Steganographic graphics with the concealed virus as input.

Finding steganographic malware and identifying possible dangers are the results.

Step 1: Data Gathering and Preprocessing

Gather a wide range of steganographic photos, including both typical photographs and images with viruses buried inside them.

By standardizing the size of the photos and transferring them to a uniform format, the dataset is preprocessed.

Step 2: Feature Extraction

Apply image processing techniques to the steganographic pictures to extract useful characteristics.

Use techniques like deep learning-based feature extraction, histogram analysis, spatial domain analysis, or wavelet analysis to gather pertinent data.

Step 3: Model Creation

Choose a suitable deep learning or machine learning model, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), for detecting steganographic malware.

Design the model's architecture while taking into account elements like the number of layers, activation methods, and optimization techniques.

Utilizing techniques like cross-validation and regularization to ensure top performance, trains the model using the preprocessed dataset.

Step 4: Model Assessment

Utilize relevant assessment measures, such as accuracy, precision, recall, and F1 score, to evaluate the trained model's performance.

To make sure the model is generalizable, validate it using a different testing dataset.

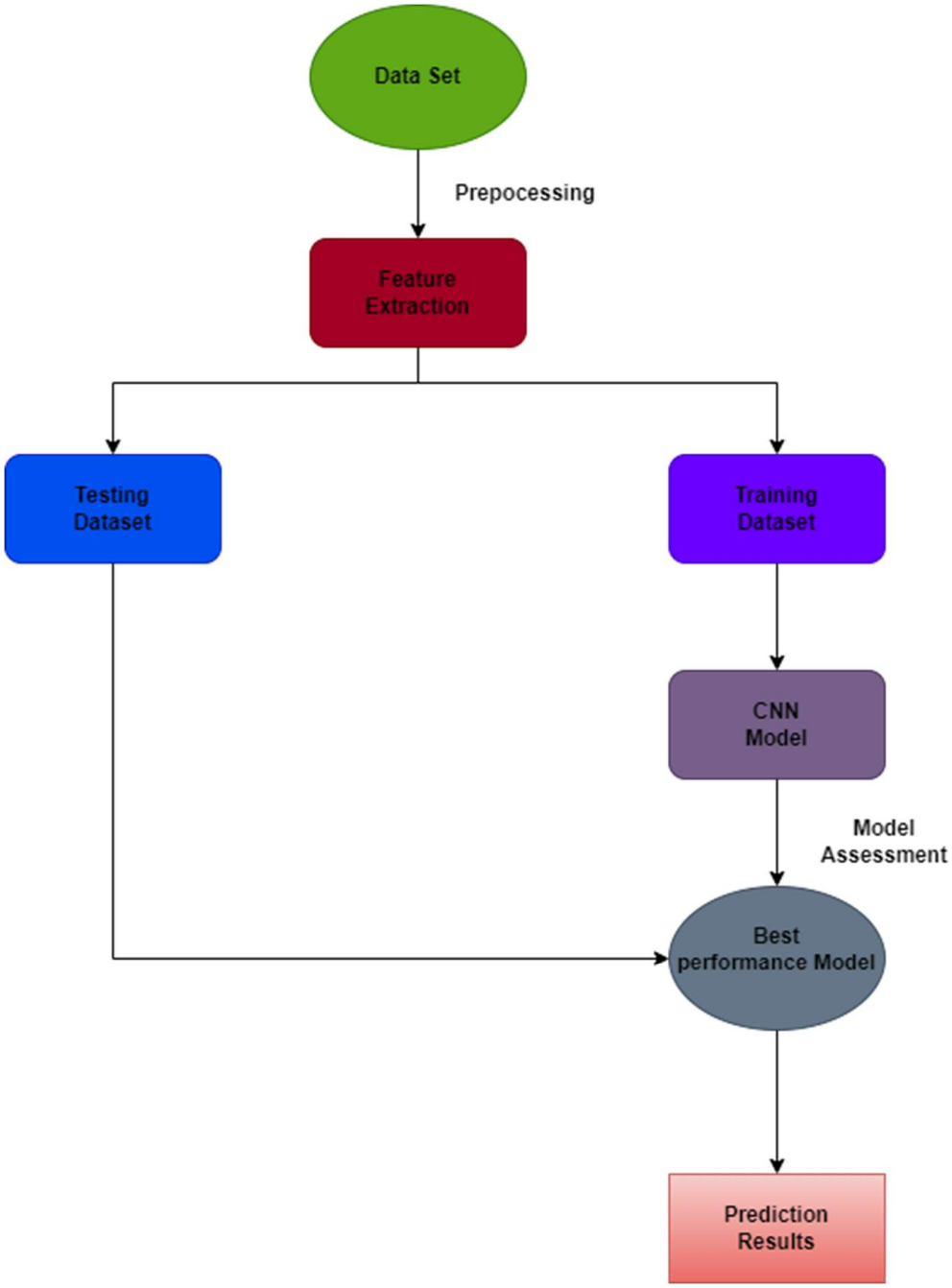

Figure 2.1 Conceptual Framework

Chapter 3. Methodology

3.1 Introduction.

In this chapter, the research approach, the methodology of how the system is going to be developed is going to be discussed and their justification.

3.2 Research Approach

The study methodology would normally favour Object-Oriented Analysis and Design (OOAD) over Structured Systems Analysis and Design (SSAD) for the project on detecting steganography-based malware threats utilizing CNN. Object-Oriented Analysis and Design (OOAD): This methodology concentrates on modelling a system as a group of interconnected objects, each with its own behaviour and data. Encapsulation, inheritance, and polymorphism are emphasized as its guiding concepts. Complex projects involving the modelling of dynamic and interactive systems are ideally suited for OOAD.

OOAD is better suited in the context of the research on detecting steganography-based malware assaults using CNNs for the following reasons:

i. The project calls for the creation and use of a CNN model, which can be successfully developed utilizing an object-oriented paradigm. Layers, activation functions, and optimization methods are a few examples of linked elements in the CNN model that each have distinct characteristics.
ii. Complex Interactions: An object-oriented approach is better able to describe and manage the linkages and interactions between the many system components, including data pre-treatment, feature extraction, model training, and integration. The system's component behaviours and complicated relationships may be modeled using OOAD.
iii. Flexibility and Modularity: When building and creating a complicated system like steganography-based malware detection, OOAD's high degree of flexibility and modularity is advantageous. Encapsulation, inheritance, and polymorphism are made possible by the object-oriented approach, making it simple to alter, expand, and manage the system.

iv. Reusability: OOAD encourages the reuse of design patterns and components, which may be helpful for the project. Utilizing predefined object-oriented libraries, frameworks, and design patterns can speed up development and improve the functioning of the system.

For the study on employing CNNs to identify steganography-based malware assaults, Object-Oriented Analysis and Design (OOAD) would be a more appropriate research methodology in light of these aspects. It makes it possible to model, build, and implement systems in an organized and adaptable way, which makes it easier to create a reliable and efficient detection system.

3.3 Modified Waterfall Methodology.

In order to provide adequate documentation and design reviews and to assure the quality, dependability, and maintainability of the generated bespoke software, the modified waterfall provides an organized sequence of development processes with some flexible iterative stages. The modified waterfall technique may be used in the context of the project on detecting steganography-based malware assaults using CNNs. Iterative components are added to the conventional waterfall model in this updated method to provide feedback loops, modifications, and continual improvement.

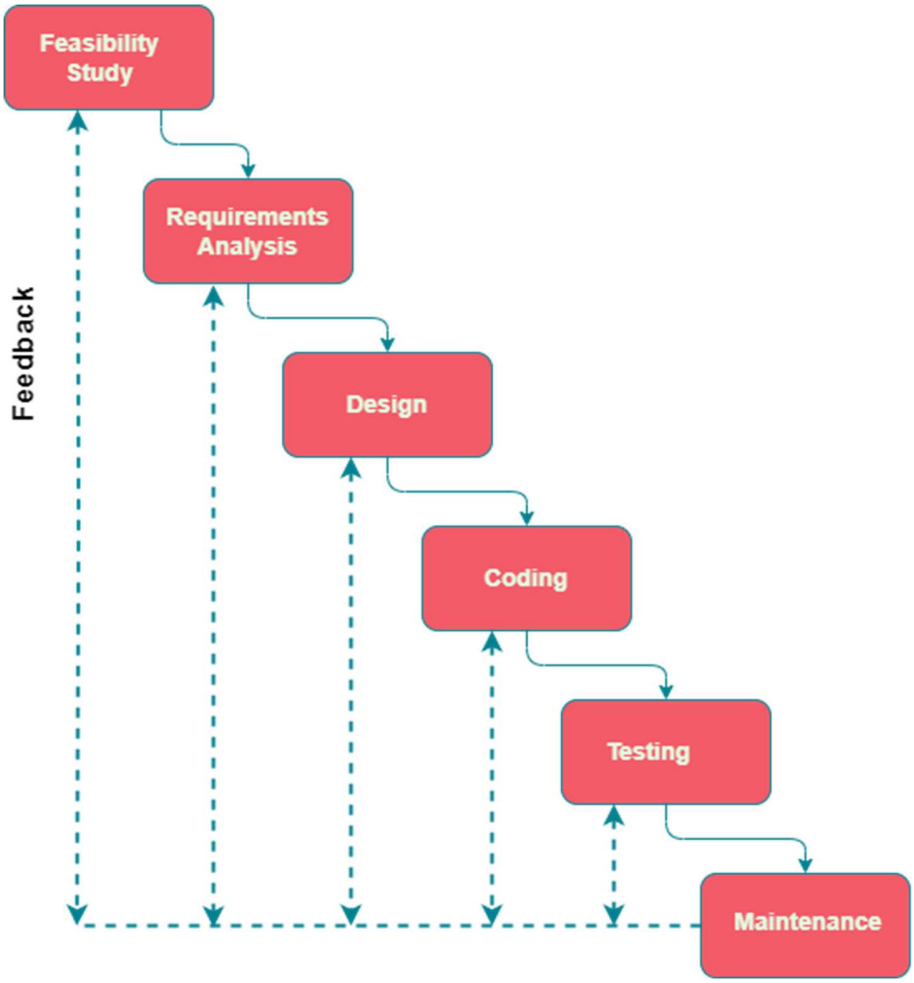

Figure 3.1 Modified Waterfall Methodology

3.3.1 Feasibility Study

A feasibility study is a crucial stage in assessing a project's viability and likelihood of success. An evaluation of the technical, financial, operational, and scheduling viability would be made as part of the project's focus on detecting steganography-based malware assaults using CNNs.

The feasibility study should focus on the following important areas:

i. Technological feasibility: For the project's implementation, the availability and suitability of the necessary technologies, tools, and resources, including hardware, software, programming languages, and libraries are to be assessed. Assessments on the selected CNN architecture and feature extraction methods are efficiency in spotting malware assaults that use steganography will be made.

ii. Functionality in Operation: Analysis of how incorporating the steganography-based malware detection solution into current infrastructure or security systems may affect operations. Also, the analysis of the system's speed and scalability to see if it can manage growing file volumes and developing malware threats will be made.

iii. Scheduling Feasibility: Whereby there will be an establishment of a reasonable project schedule that takes into account all of the project's phases, including data collection, pre-processing, model construction, training, assessment, optimization, and integration. The determination of any possible obstacles, dependencies, or dangers that could delay the project will also be done in terms of the resources, and outside influences that could have an impact on the project's development and conclusion.

iv. Ethical and Legal feasibility: Assessments of any applicable legal or moral issues that may be relevant to the project, such as data protection laws, privacy legislation, and the handling of sensitive information and malware samples in accordance with ethical standards. Ensuring adherence to all applicable legal and ethical frameworks in securing essential authorizations and consents as needed.

Feasibility analysis aids in evaluating if a project is worthwhile and directs the preparation and implementation phases.

3.3.2 Requirement Analysis.

To guarantee a comprehensive grasp of what has to be accomplished, the requirements analysis phase focuses on obtaining, assessing, and documenting the project's requirements. There are functional requirements and non-functional needs for the project on employing CNN to identify steganography-based malware assaults.

The functional requirements are like file analysis whereby, the system should be able to analyse different types of images for the presence of steganography. The system is also required to employ CNN to detect steganography techniques used to hide malware within files. The system should also provide real-time detection capabilities to identify steganography-based malware attacks as soon as files are submitted for analysis. The system should achieve a high level of accuracy in detecting steganography-based malware, minimizing false positive and false negatives. The system should also generate alerts when steganography-based malware is detected providing relevant details about the affected files and a summary of report should be generated for the results.

The system should be capable of integrating with existing security systems or infrastructure allowing seamless incorporation into an organization's security architecture.

The non-functional requirement for the system are evaluated in terms of performance whereby the system should be able to process and analyse files within a reasonable time frame putting into consideration the volume and complexity of files and it should also have low latency in detection of steganography-based malware attacks. The system should also be scalable to handle increasing volumes of files and concurrent requests for analysis. The system should also be user friendly allowing the user to submit files for analysis and viewing results. The system should also adhere to the security and privacy standards ensuring confidentiality and integrity of analysed data. It should handle malware samples and sensitive information in a secure manner, preventing unauthorized access or leakage. The systems design should be done in a modular and maintainable manner in that it can allow for future updates, enhancements and bug fixes. The codebase should be well-documented, making it easier for developers to understand and maintain the system.

3.3.3 Design

In this design, the system is to be fed the image, then the image would go through the normalization process and therefore it will be taken through the CNN algorithm where it would be able to detect whether the image has malware or not. The final results of this project would be whether the image does not have a malware or if it actually has a malware.

The figure below is a decision tree to describe the design of the system.

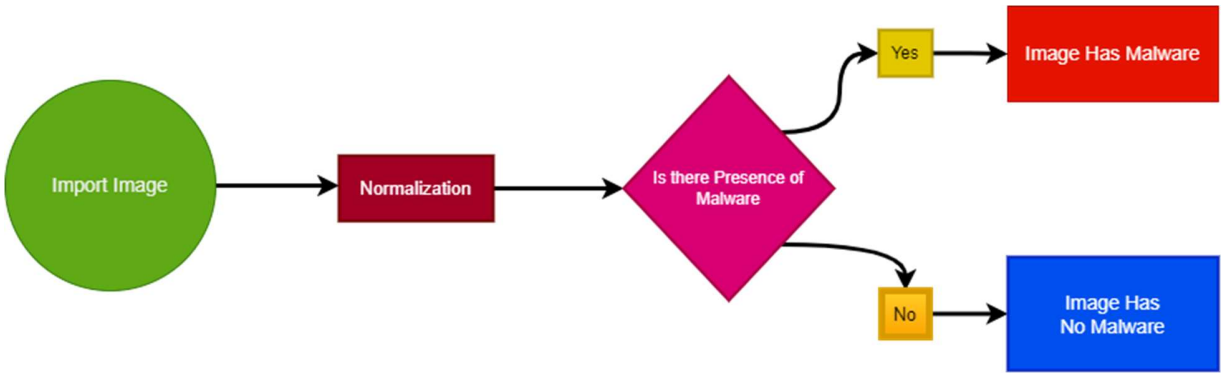

Figure 3.2 Decision Tree

3.3.4 Coding and Implementation

i. **Frameworks & Libraries:**

CNN models may be created and trained using the popular open-source deep learning framework TensorFlow. On top of TensorFlow, Keras is a high-level neural network API that offers a user-friendly interface for model creation.

OpenCV: An image processing and manipulation library for computer vision. NumPy is a numerical computing framework that facilitates the effective management of multidimensional arrays and mathematical computations.

ii. **Algorithms and Functions:**

The primary technique used for steganography-based malware detection is convolutional neural networks (CNNs). Convolutional, pooling, and fully connected layers make up CNNs, which can categorize input files and learn to extract characteristics from them.

For the purpose of improving the generalizability of the model, pre-processing techniques and functions for input files include scaling, normalization, and data augmentation.

Features that can help identify steganography-based malware include statistical features, spatial domain features, and frequency domain features, which are functions that extract pertinent aspects from files.

iii. **Datasets:**

A set of benign files, such as photos, documents, and executables, is used to train the CNN model to discriminate between legitimate files and malware that uses steganography.

Steganography-Based Malware Dataset: A collection of files with malware that has been concealed using different steganography methods. The positive class in training and assessment is this dataset.

iv. **Expected Outcomes:**

Accurate Detection: The method tries to precisely identify malware assaults using steganography in input files. This is accomplished by training the CNN model on a varied dataset that includes both benign files and malware samples that use steganography, resulting in a model that can learn pertinent patterns and characteristics suggestive of the existence of malware.

Reduced False Positives and False Negatives: To achieve accurate detection performance, the system should reduce false positives (classifying a benign file as malware) and false negatives (failing to identify malware).

Real-time detection: The system must deliver immediate detection findings to enable quick defence against malware assaults utilizing steganography.

v. **Assuring Precision:**

The CNN model learns from a variety of benign files and steganography-based malware samples when large and representative datasets are used, which results in higher generalization and accuracy.

Model Optimization: By adopting optimization strategies, such as modifying learning rates, applying regularization techniques, and early halting, the model's capacity for generalization is enhanced and overfitting is prevented.

Cross-validation: Using approaches like k-fold cross-validation minimizes the possibility of biased performance estimates by evaluating the model's performance on several subsets of the dataset.

Metrics for Evaluation: Accuracy and detection performance of the model may be evaluated quantitatively by calculating metrics like accuracy, precision, recall, and F1-score.

Continuous Monitoring and Updates: The CNN model's accuracy and efficacy in spotting changing steganography-based malware assaults are ensured by routinely checking the system's performance and upgrading it with fresh data and cutting-edge steganography techniques.

The system's architecture makes use of deep learning, image processing, and numerical computing modules and frameworks. The implementation of pre-processing, feature extraction, and CNN-based classification functions and algorithms. The training and assessment of the CNN model are made possible by datasets that include samples of both benign files and malware that use steganography. Accurate detection, fewer false positives and false negatives, and real-time performance are all intended outcomes of the system. The correctness of the outcomes is supported by procedures including dataset selection, model optimization, cross-validation, evaluation metrics, and continual monitoring.

3.3.5 Testing

After the system has been designed, the ability of the system to actually detect the presence of an embedded code will be put to the test. This will be done locally so as to ensure that maximum system resources are utilized. The expected results are basically whether an image a malware script embedded within it.

3.4 Deliverables

The outcome of this project is a system that can detect stegomalware and notify the user of the outcome. That is whether the image has malware or not. The F1-score would be used to test the systems accuracy in detection.

i. Obtaining and cleaning the dataset
   The dataset will be downloaded the JPEG kaggle.com website and imported using the OpenCV.
ii. Developing the CNN based model
   Here, the CNN algorithm will be developed using the Keras Library, and Tensor Flow.
iii. Training the Model
   This is where the dataset imported would be used to train the CNN algorithm to detect malware in images
iv. Testing the System
   The accuracy of the system would be tested by obtaing the F1-score.
v. Documentation
   The milestones of the proceedings of building the systems will be documented here.

References:

A, M., & R, E. (2022). *An Ensemble-based Stegware Detection System for Information Hiding Malware Attacks* [Preprint]. In Review. https://doi.org/10.21203/rs.3.rs-613020/v1

Ashraf, M. (2018). *Iqbal et al 2018*.

BencsathPBF11duqu.pdf. (n.d.). Retrieved 1 June 2023, from https://www.crysys.hu/publications/files/bencsathPBF11duqu.pdf

Caviglione, L., & Mazurczyk, W. (2015). *Information Hiding as a Challenge for Malware Detection*. https://doi.org/10.13140/RG.2.1.1538.5122

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378. https://doi.org/10.1111/mice.12263

Chaganti, R., Ravi, V., Alazab, M., & Pham, T. (2021). *Stegomalware: A Systematic Survey of Malware Hiding and Detection in Images, Machine Learning Models and Research Challenges*. https://doi.org/10.36227/techrxiv.16755457

Choudhary, K. (2020). *IMPLEMENTATION, DETECTION AND PREVENTION OF STEGOMALWARE*. https://doi.org/10.13140/RG.2.2.19870.97609

Choudhury, S. (2019). *Stegware Destruction Using Showering Methods*. *8*(6).

Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2022). *A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms* (arXiv:2209.03622). arXiv. https://doi.org/10.48550/arXiv.2209.03622

Mazurczyk, W., & Caviglione, L. (2014). *Steganography in Modern Smartphones and Mitigation Techniques* (arXiv:1410.6796). arXiv. http://arxiv.org/abs/1410.6796

Paganini, P. (2014, February 18). *Detected new Zeus variant which makes use of steganography*. Security Affairs. https://securityaffairs.com/22334/malware/zeus-banking-malware-nestles-crucial-file-photo.html

Sudhakar, & Kumar, S. (2019). *An emerging threat Fileless malware: A survey and research challenges*. *3*, 1. https://doi.org/10.1186/s42400-019-0043-x

Chaganti, R., Ravi, V., Alazab, M., & Pham, T. (2021). *Stegomalware: A Systematic Survey of Malware Hiding and Detection in Images, Machine Learning Models and Research Challenges*. https://doi.org/10.36227/techrxiv.16755457

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Islam, N., Farhin, F., Sultana, I., Shamim Kaiser, M., Sazzadur Rahman, Md., Mahmud, M., S. M. Sanwar Hosen, A., & Hwan Cho, G. (2021). Towards Machine Learning Based Intrusion Detection in IoT Networks. *Computers, Materials & Continua*, *69*(2), 1801–1821. https://doi.org/10.32604/cmc.2021.018466

Sebov, K. (n.d.). *Deep Rearing: Mice Behaviour Analysis in Open Field Test using CNN*.

Tolba, M. F., Halawani, Y., Saleh, H., Mohammad, B., & Al-Qutayri, M. (2020). FPGA-Based Memristor Emulator Circuit for Binary Convolutional Neural Networks. *IEEE Access*, *8*, 117736–117745. https://doi.org/10.1109/ACCESS.2020.3004535

References:

A, M., & R, E. (2022). *An Ensemble-based Stegware Detection System for Information Hiding Malware Attacks* [Preprint]. In Review. https://doi.org/10.21203/rs.3.rs-613020/v1

Ashraf, M. (2018). *Iqbal et al 2018*.

BencsathPBF11duqu.pdf. (n.d.). Retrieved 1 June 2023, from https://www.crysys.hu/publications/files/bencsathPBF11duqu.pdf

Caviglione, L., & Mazurczyk, W. (2015). *Information Hiding as a Challenge for Malware Detection*. https://doi.org/10.13140/RG.2.1.1538.5122

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378. https://doi.org/10.1111/mice.12263

Chaganti, R., Ravi, V., Alazab, M., & Pham, T. (2021). *Stegomalware: A Systematic Survey of Malware Hiding and Detection in Images, Machine Learning Models and Research Challenges*. https://doi.org/10.36227/techrxiv.16755457

Choudhary, K. (2020). *IMPLEMENTATION, DETECTION AND PREVENTION OF STEGOMALWARE*. https://doi.org/10.13140/RG.2.2.19870.97609

Choudhury, S. (2019). *Stegware Destruction Using Showering Methods*. *8*(6).

Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2022). *A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms* (arXiv:2209.03622). arXiv. https://doi.org/10.48550/arXiv.2209.03622

Mazurczyk, W., & Caviglione, L. (2014). *Steganography in Modern Smartphones and Mitigation Techniques* (arXiv:1410.6796). arXiv. http://arxiv.org/abs/1410.6796

Paganini, P. (2014, February 18). *Detected new Zeus variant which makes use of steganography*. Security Affairs. https://securityaffairs.com/22334/malware/zeus-banking-malware-nestles-crucial-file-photo.html

Sudhakar, & Kumar, S. (2019). *An emerging threat Fileless malware: A survey and research challenges*. *3*, 1. https://doi.org/10.1186/s42400-019-0043-x

Appendices

Appendix 1: Gantt Chart