National College of Ireland

Detection of Ransomware Attacks using Supervised Machine Learning

MSc Research Project
Cyber security

Laith Abu Saad

Student ID: X21148520

School of Computing

National College of Ireland

Supervisor: Mr. Jawad Salahuddin

National College of

Ireland MSc Project

Submission Sheet

School of Computing

Student Name: Laith Abu Saad

Student ID: X21148520

Programme: Cyber Security **Year:** 2022

Module: MSc Research project

Lecturer: Mr. Jawad Salahuddin

Submission Due Date: 15/12/2022

Project Title: Detection of Ransomware Attacks using Supervised Machine Learning

Word Count: 7695 **Page Count: 32**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Laith Abu Saad

Date: 15/12/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□

You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

Abstract

The continuous growth and advancements of technology-based products in several industries and sectors of the economy and society have also been plagued by an increasing number of cyber-attacks targeted at compromising systems, stealing sensitive information, etc. These attacks take several shapes including malware attacks, phishing, and distributed denial of service (DDoS) amongst others. Consequently, ransomware has been identified as a major type of malware attack. Therefore, it has become pertinent that more attention is drawn to creating tools and techniques to specifically detect ransomware attacks. Several research exists that has focused on techniques such as Machine Learning (ML) and deep learning algorithms for detecting malware, however, few have been centred on ransomware detection. As such, in this research, the Random Forest classifier and Logistic regression classifier as machine learning techniques are explored to determine their accuracy in the detection of ransomware attacks. The logistic regression model achieved an accuracy of about 74% with a precision and recall of around 74% of average each. The random forest model outperformed the logistic regression, achieving a near 100% accuracy, precision, and recall, with only 2 misclassifications in the confusion matrix out of 350 thousand rows on the test dataset.

Keywords: *Ransomware, machine learning, malware, random forest, logistic regression*

Video Link: https://www.youtube.com/watch?v=CWt-_pGyJcc

1 Introduction

Malware known as malicious software is one of the most common cyber threats today. This could come in the form of virus, trojan, ransomware, adware etc. Ransomware attacks have gained more popularity over the years and have become one of the top methods that attackers use to launch attacks on targets as it is difficult to mitigate unlike other security issues (Chittooparambil et al., 2019). It is a type of malware that is designed to restrict access to user files by encrypting them and demanding a ransom in order to obtain the decryption key. According to Statistica, in 2021 there were over 600 million ransomware attacks ("Number of ransomware attacks per year 2022", 2002). Although this number has seen a reduction in 2022, it still constitutes a major challenge as it provides a unique mix of low risk and big return for cybercriminals which explains the large numbers of ransomware attacks recorded. Additionally, in the report "Ransomware Attacks and the True Cost to Business", the researchers stated that the ransomware which has significantly increased in sophistication and has led to mega financial loses to companies (Team, n.d.). The major difference between other types of malware and ransomware is that, other malware types could infect a system and hide behind applications to compromise it and steal sensitive information without asking for a ransom while the goal of a ransomware is to render a system inoperable and encrypt data on systems so as to require a ransom to be paid before data decryption keys can be provided to the victim. According to (Chittooparambil et al., 2019). methods available for malware detection are not efficient enough in detecting ransomware. Additionally (Wecksten et al., 2016) and (Kirda, 2017) stated that attacks inspired by ransomware have become difficult to defend and mitigate. Hench, traditional methods for detecting malware have been identified to be inefficient as mentioned in several researches including (Vinayakumar et al., 2019). These methods analyse behavioural patterns and signatures of malware by majorly leveraging on static and dynamic techniques which consume a lot of time and are unable to detect in real time unknown malwares signatures. Gilbert et al in their paper also stated that traditional techniques are unable to measure up to the sophistication of new malware. (Chang et al., 2017) further supported the use of machine learning approach due to its ability to detect unknown malware samples and zero-day attacks as opposed to traditional static and dynamic methods which do not seem to efficiently detect malware.

According to (Wu et al., 2020), in 2020, about 51% of the organizations worldwide recorded a ransomware attack. These ransomware attacks were sophisticated and use

advanced command and control servers. This ultimately made reverse engineering difficult and shows that currently, these traditional static and dynamic approaches to malware detection are inefficient and thus, a major challenge exists in the cyber security world as ransomware is polymorphic in nature and as such evades these approaches (Kapoor et al., 2021). Hence, as cybercriminals continue to become more sophisticated in their design of ransomware, more efficient methods of detection is required. In line with this, Machine Learning has been explored by several studies in relation to malware detection. This is because machine learning techniques have been identified to be efficient as it trains a model that can learn and classify malicious or benign ware. In (Matin and Rahardjo, 2019), the researchers demonstrated that the machine learning models produced were successful in detecting malware as they could distinguish between malicious and benign network traffic. Also, Fraley and Cannady in their paper "The promise of machine learning in cybersecurity" highlighted the potential of machine learning as a tool that can help in producing more efficient methods in malware detection due to its capacity to handle large data, detect modern malware attacks and improve scanning engines (Choo, 2011). This was also supported by (Chio and Freeman, 2018) which opined that the adoption of machine learning is an effective solution for ransomware detection. Along the same line, (Anderson et al., 2011), (Kolter and Maloof, 2004) in their research presented Machine learning as an alternative to signature based methods. Both researches concluded by stating that Machine learning proved to be more effective.

As such, this paper explored machine learning as a technique for ransomware detection. It adopted random forest algorithm to produce a model that accurately detects ransomware. Therefore, the aim of this study is to answer the Research Question **(RQ): How does Random Forest compare to Logistic Regression in detecting Ransomware attacks?** Its objectives include:

- Investigate current malware detection techniques.
- Investigate ransomware detection techniques
- Produce 2 machine learning models that will accurately detect ransomware
- Evaluate the models produced in terms of accuracy and performance using performance metrices.

Accordingly, it contributes in the following ways:

- Provides an overview of malware detection
- Exposes current issues being faced in detecting ransomware

- Adds to the body of knowledge by comparing the use of random forest classifier and logistic regression classifier in ransomware detection.

Subsequently, the paper is organized into 4 sections. Section 2 provides the literature review covering an overview of malware and malware detection techniques, Ransomware Detection Techniques and Machine Learning for Ransomware Detection. Subsequently, section 3 describes the research methodology while section 4 provides the research's design specification. Thereafter, section 5 and section 6 details the implementation and evaluation respectively. The paper concludes with section 7 where the limitations of the research and proposed future work is captured.

2 Related Work

With the high number of recorded ransomware attacks and current traditional detection methods not being efficient enough to mitigate them, more efficient detection methods for ransomware attacks have become one of the most researched areas. In line with this, most research have outlined machine learning as a potential technique that will deliver success in the area, while others have carried out experiments that resulted in machine learning models which achieved high accuracies in ransomware detection. Therefore, this section outlines a review of several studies related to this research. It is categorised into 3 sub-sections: an Overview of Malware and Malware Detection Techniques, Ransomware Detection Techniques and Machine Learning for Ransomware Detection

2.1 Overview of Malware and Malware Detection Techniques

Malware generally known as malicious software is any software designed with the intent to be used to cause harm to a computer system. According to Malwarebytes, it's aim is to interfere with the normal functioning of a system by infecting, disabling, or gaining unauthorised access to sensitive information stored (Hama Saeed, 2020). Ahmad in his paper also describes it as malicious code used in several forms such as viruses, trojans, ransomware, spyware etc. to destroy targeted computer systems or applications (Olawale Surajudeen, 2012). (Vinayakumar et al., 2019) further describes malware as any program created with a bad intent. The paper also outlined that malware can be categorized according to their purpose and method of propagation. Some of the attributes of malware is its ability to infect, self-replicate and spread within a system without being detected. This malicious software could be attached to files downloaded from the internet, malicious links, or malicious websites. Hence, effort have been

put into creating anti-malware and other tools geared towards detecting malware to avoid malware infections. To this end, the process of identifying malicious software is referred to as malware detection.

Furthermore, Faruk et al opines that the objective of malware detection is to guard any system against malware attacks by detecting the presence of malware and preventing malware infection (Hossain Faruk et al., 2021). The paper also stated that the first step in any malware detection effort is discovering the source code of the malware. Consequently, a malware detection model is a model that can accurately distinguish between malicious and benign programs based on their distinctive behaviours. To this end, there are several approaches that have been identified to be used in malware detection. (Bazrafshan et al., 2013) identified three major malware detection techniques namely: signature based, behavioural based and Heuristic based malware detection techniques. Similarly, (Idika and Mathur, 2007) grouped this into signature based and anomaly-based malware detection while (Mujumdar et al., 2013) categorized malware detection approach into signature based and behavioural based. Additionally, with current advancement and sophistication used in designing and deploying malware by cyber criminals, machine learning has been identified as a more efficient approach in producing models with high accuracy in malware detection. This is supported by papers like (Basu et al., 2016), (Kumar and Lim, 2019), and (Ham and Choi, 2013) which stated that machine learning has the potential in providing more efficient approach in malware detection. The following sub-section highlights the different malware detection techniques as discussed by other related works.

2.1.1 Signature Based

Here, detection is based on signatures of known attacks such as file metadata, fingerprints etc. Any code or file that matches a specific pattern already identified as malicious code is classified as malware. Thus, malware is detected based on matches to an already defined set of specific patterns known to be malicious. This technique is used in most anti-malware programmes where a file is scanned and evaluated against a database of known malware signatures. A match with any component of the database indicates the presence of malware. Although, this method is effective against known malware, it has proved to be ineffective against new and unknown malware whose signatures are yet to be defined. Also, the polymorphic nature of malware as seen in recent times render signature-based detection technique ineffective as the malware can change their signature (Aslan and Samet, 2020). (Aslan and Samet, 2020) also stated that

machine learning holds the potential to enhance the efficacy of signature-based malware detection technique. Furthermore, the limitations of signature-based technique saw the use of behavioural based approach.

2.1.2 Anomaly Based/Behavioural Based

Behavioural and Anomaly based malware detection approach have been used interchangeably as most researchers have identified them to be one and same. This approach conceived as a solution to the limitations of signature-based approach, is based on observations of malware behaviour when executed usually in a sandbox to prevent infection of the host. Some behaviours as identified in (Ghafir and Prenosil, 2014) could be a combination of the following:

- An attempt to detect a sandbox environment
- Modifying or encrypting files
- Disabling security controls and installing unknown software
- Deleting or adding system files
- Disabling or shutting down system operations etc

According to (Aslan and Samet, 2020), this approach mitigated the draw backs of signature-based approach as it was able to detect unknown malware and was effective against the polymorphic nature of new malware. Also, (Heena, 2021) in their paper stated that the behavioural based technique improves malware detection as it inspects what a malware program does. Hence, even if it keeps mutating, it can still be detection based on its behaviour as its effects on system resources will be similar (Ghafir and Prenosil, 2014). However, in (Ham and Choi, 2013), the researchers countered this by stating that the behavioural based approach was insufficient against malware obfuscation and polymorphism because of high spreading rate of polymorphic malware as well as high false-positives and false-negatives in results. The limitations of this approach led to the development of a more efficient technique i.e., heuristic approach.

2.1.3 Heuristic/Hybrid Based

Heuristic /Hybrid approach is a combination of both signature based and behavioural based approach. On top of this, the approach also leverages the use of machine learning in addition to the combination. This technique has been identified to be a solution to the limitations of the previous malware detection approaches. According to (Aslan and Samet, 2020), this combination with machine learning allows a model to be trained and tested to classify and in

turn detect malware. The paper also stated that this approach is effective in detecting zero-day attacks and overcomes malware obfuscation and polymorphism. Firdausi et al also supported this by highlighting the effectiveness of a combination of behavioural based detection approach and machine learning for malware detection (Firdausi et al., 2010). Additionally, an experiment conducted in (Mohaisen et al., 2015) produced a detection model AMAL based on a behavioural based malware detection system (AutoMal) and a machine learning classification system (MaLabel) which achieved high accuracy. Consequently, machine learning will be explored in subsequent section.

2.2 Ransomware Detection Techniques

In (Fernando et al., 2020), Ransomware is described as a type of malware which when executed prevents access to the infected system, file, operating system, or device. Its aim is to encrypt data while demanding for a ransom to release decryption keys to victims. It also went further to highlight that there are 2 major forms of ransomware, Locker ransomware which usually displays a lock screen with a message requesting for ransom to release access to the system and Crypto ransomware which encrypts files on a system and demands for payment in cryptocurrency to decrypt those files.

Over the years, ransomware attacks have been on the rise and has advanced in design. According to helpnetsecurity.com, in 2021, 80% of an organization's critical infrastructure experienced a ransomware attack (Security, 2022). 37% of global organizations reported that they were victims of ransomware attacks ("IDC's 2021 Ransomware Study," n.d.). This has resulted in loses to organizations. Ransomware such as CryptoLocker, CryptoWall have been reported to have generated over 320 million dollars in revenue. Consequently, ransomware constitutes a serious challenge for most organizations and the need for more specific detection for ransomware cannot be overemphasized. To this end, several researchers have utilized different approaches for ransomware detection. For example, in (Arabo et al., 2020), the researchers utilized process behaviour analysis in ransomware detection. Their experiment focused on investigating if a process running within an ecosystem is ransomware or not. The analysis which was conducted using 7 ransomware, 41 benign software, and 34 malware samples achieved results with low false-positive and false-negative rate in classifying ransomware and benign applications. Also, (Fernando et al., 2020) researched the use of machine learning and deep learning in the detection of ransomware. Their conclusion was that both approaches which produced models trained using network, behavioural, or static features

when evaluated achieved high accuracy in detecting ransomware from mid to high 90s (Fernando et al., 2020). Furthermore, (Singh et al., 2022) investigated the viability of using process memory as a mechanism for ransomware detection. Their experiment used several machine learning algorithms which produced models with accuracy ranging from 81%-96%. They concluded by confirming the feasibility of using process memory in ransomware detection.

Accordingly, honeypot has also been used as a technique for ransomware detection as highlighted in (Moore, 2016) where it was used to monitor changes happening in a folder. (Kolodenker et al., 2017) proposed a tool called PayBreak which stores the encryption keys that can be used to decrypt files affected by a ransomware attack. Furthermore, deep learning which is another field under artificial intelligence has also received attention in ransomware detection. Studies such as (Tseng et al., 2016) and (Ren et al., 2020) leveraged on deep learning techniques for ransomware detection. Both studies reported high accuracies of the detection models.

2.3 Machine Learning for Ransomware Detection

Machine learning has been applied to several fields including cyber security. It has been widely adopted as it gives models the ability to learn through training without the need for human supervision or programming. Here, two stages are involved, the training stage and the testing stage. In the training stage, a model learns from a dataset while the testing stage involves the model being applied to other unknown dataset so it is able to predict, recognise and classify based on properties learnt. Thus, any machine learning model is heavily data-driven. Some examples of machine learning algorithms commonly used in developing detection models include, Support Vector Machine (SVM), Random Forest, Decision Tree, Logistic Regression, Linear regression amongst others. (Zhang and Zulkernine, 2006) in their experiment, studied the behaviour of ransomware in a sandbox and utilized an SVM classifier. Their experiment achieved an accuracy of 97.48% in detecting ransomware.

According to (Khammas, 2020), Random Forest is considered to achieve better results than other machine learning algorithms as it requires less input parameters and is resistant to overfitting. This is also supported by (Zhang et al., 2019) and (Ahmad et al., 2018) who suggested that random forest classifier outperforms other machine learning classifiers in the detection of various attacks. Also, a study conducted by (Takeuchi et al., 2018) used Random Forest as a machine learning technique for ransomware detection. The researchers used a dataset containing 840 ransomware executable of different families, and 840 benign files. They

reported an impressive performance of the random forest classifier with an accuracy of 97.74%. (Zhang et al., 2019) reported that random forest achieved the highest accuracy of 91.43% in classifying ransomware and benign ware when compared to four other ML algorithms i.e., K-Nearest Neighbor, Decision Tree, Gradient boosting, and Naive Bayes. Their experiment utilized opcode features. Similarly, Masum et al in their experiments used 5 different machine learning algorithms: Random Forest, Decision Tree, Naïve Bayes, Logistic Regression and Neural Network over the same dataset to distinguish between ransomware and benign ware. Random forest achieved the highest results, thus outperforming the other four in terms of accuracy. Finally, the accuracy achieved by the above-mentioned studies, show that machine learning algorithms and more specifically random forest is a viable approach in developing models with high accuracy in ransomware detection. As such, this research adopted the use of Random Forest to develop a model for binary classification in the detection of ransomware.

Some of the related works, the methodology adopted, their results and limitations are summarized below:

Table 1: Summary of Some Related Works

Authors	Methodology	Results	Limitation
(Arabo et al., 2020)	Process Behaviour	Low False positive & false negative rate	Process is time consuming
(Fernando et al., 2020)	Deep Learning Algorithms	90 – 95% accuracy	High false negative rate
(Zhang et al., 2019)	Random forest algorithm	91% accuracy	Low sample size for training data
(Takeuchi et al., 2018)	Random forest algorithm	About 98% accuracy	Low sample size for training & testing data

3 Research Methodology

A research methodology is an approach used to methodically solve research problems. It can be viewed as a science that studies how scientific research is conducted. There are many approaches typically used by researchers to analyse research problems, as well as the reasoning behind them. Researchers must be familiar with both the methodology and the techniques that apply to their problem. In addition to knowing how to create and formulate their research

question, prepare the dataset, select the features/ variables needed for modelling, train and test their model, and evaluate the performance of their model, researchers also need to know which of these steps are pertinent and which are not, as well as the meaning and justification of each step (Fayyad et al., 1996). All of this means that the researcher must create his approach specifically for his topic as methodologies might vary from problem to problem. The methodology implemented in this research project is based on Knowledge Discovery in Databases (KDD) (Tavallaee et al., 2009). This methodology includes stages like data selection, processing, transformation, data modelling, and interpretation/ evaluation. A flowchart for the research methodology is shown below:

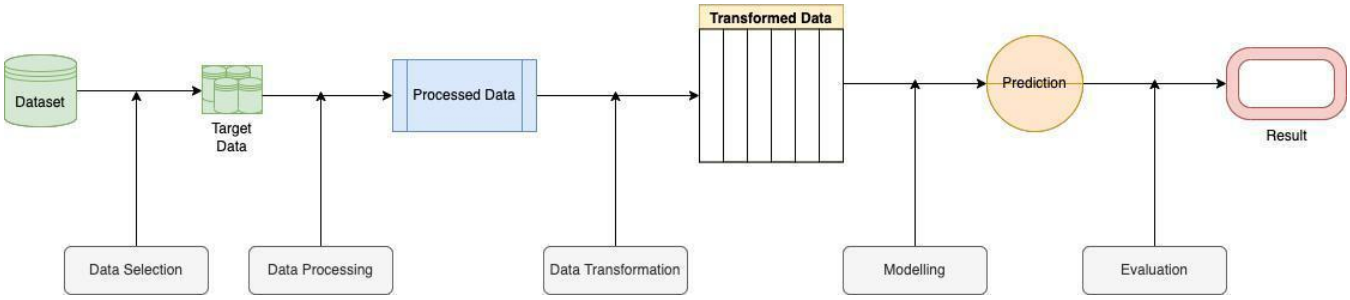

Figure 1: KDD Research Methodology

3.1 Data Selection

The choice of the dataset was provided by (Garcia et al., 2020) as part of research into intrusion detection in network packets. The dataset contains both benign and malicious traffic along with details describing each traffic packet. The following features are captured for each packet:

1. Timestamp of the traffic
2. Unique ID (uid) for the packet. This identifier is hashed and cannot be read by humans. This is done to protect the identity of where the packet originated.
3. Id_origin.h is the origin ip of the packet which has been randomised for data privacy reasons.
4. Id_origin.p is the port where the packet originated.
5. Id_dest.h is the destination ip of the packet which has been randomised for data privacy reasons.
6. Id_dest.p is the destination port of the packet.
7. Proto is the protocol of the packet. This could be udp, tcp, ftp etc.
8. Service is the type of service requested by the packet.
9. Duration …..

The dataset contained over 10 million rows and 22 columns (as listed above). Each traffic was identified and labelled as either benign or malicious by the authors (Garcia et al., 2020). Furthermore, the malicious traffic was also broken out into different types of malicious traffic such as C&C, DDoS, Horizontal Port Scan, etc.

Upon reading the packet log files, the dataset was processed using the Pandas dataframe in python. The pandas dataframe makes it easier to process and analyse the data in preparation for modelling.

3.2 Data Processing

When information is gathered and transformed into a readable format, data processing takes place. It is crucial that data processing is done appropriately so as not to adversely affect the final product, or data output, which is often carried out by analysts (Gjerloev, 2012). The data processing steps carried out in this research include steps such as data relabelling and data balancing. These steps are further explained below.

3.2.1 Data Relabelling

The original dataset included benign traffic as well as different classes of malicious traffic. The data relabelling was performed to rename all the classes of malicious traffic to ransomware while the benign traffic was renamed to normalware. The data relabelling was done to ensure consistency in the data. The images below show the labelled column before and after they were renamed.

```
1 #Here we see the number of benign traffic and the number of malicious traffic
2 df['tunnel_parents   label   detailed-label'].value_counts()
```

executed in 1.16s, finished 20:24:32 2022-11-28

```
-   Benign   -                      8262389
-   Malicious   DDoS                2185302
-   Malicious   C&C                      81
-   Malicious   C&C-FileDownload         12
-   Malicious   Attack                    3
Name: tunnel_parents   label   detailed-label, dtype: int64
```

Figure 2: Dataset before Relabelling

```
1 #Here we see the number of normalware and ransomware traffic after relabelling
2 df['traffic type'].value_counts()
```

executed in 1.47s, finished 16:34:42 2022-12-03

```
normalware    8262389
ransomware    2185398
Name: traffic type, dtype: int64
```

Figure 3: Dataset after Relabelling

3.2.2 Data Balancing

A dataset is said to be imbalanced if there is a large percentage difference between the positive and negative values of the dependent variable – in this instance the distribution between the normalware and ransomware traffic. An imbalanced dataset has an adverse effect on the performance of any predictive model. In a situation where a model is trained on a dataset with significantly higher positive values than negative values, if the model achieves high accuracy, when testing or deploying the model, it will more often make predictions favouring the value with the higher distribution. Consequently, it is important to balance the dataset before proceeding with data modelling, to eliminate this bias in the model. The original dataset had a distribution of about 79% to 21% of normalware to ransomware traffic with over 8 million normalware traffic and 2 million ransomware traffic.

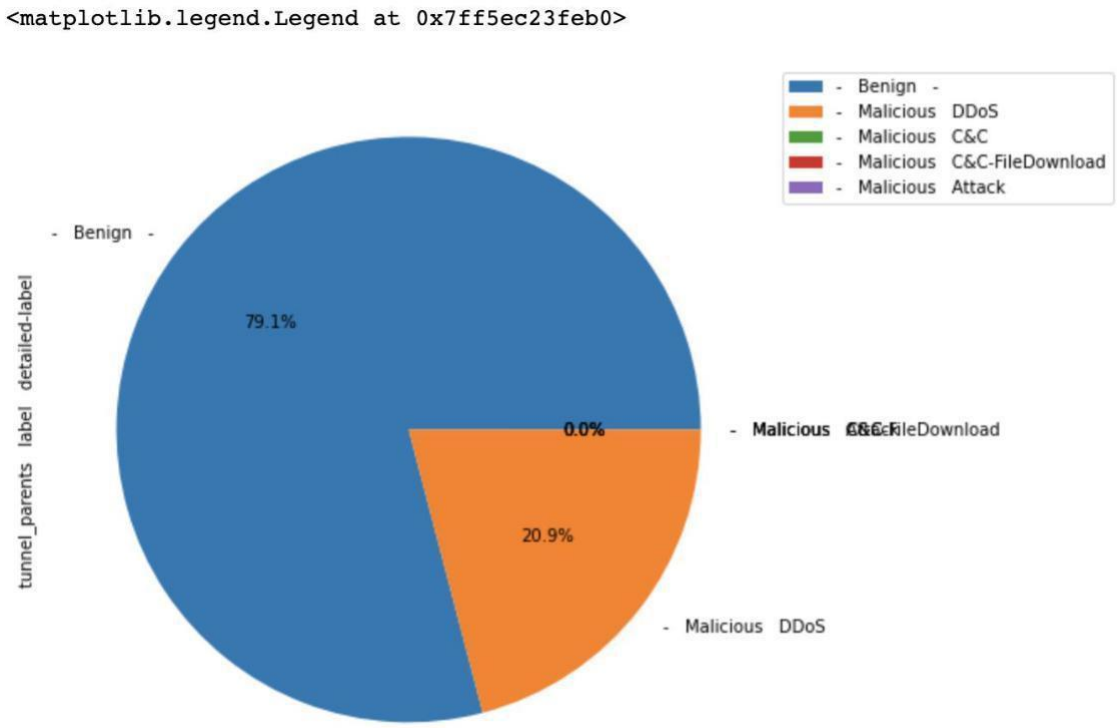

Figure 4: Distribution of Original Dataset before Data Balancing

The data balancing technique implemented in this research involves taking equal random samples of both the normalware traffic and the ransomware traffic – with 1 million records from each traffic type. Other data balancing techniques such as SMOTE uses oversampling or undersampling for smaller datasets.

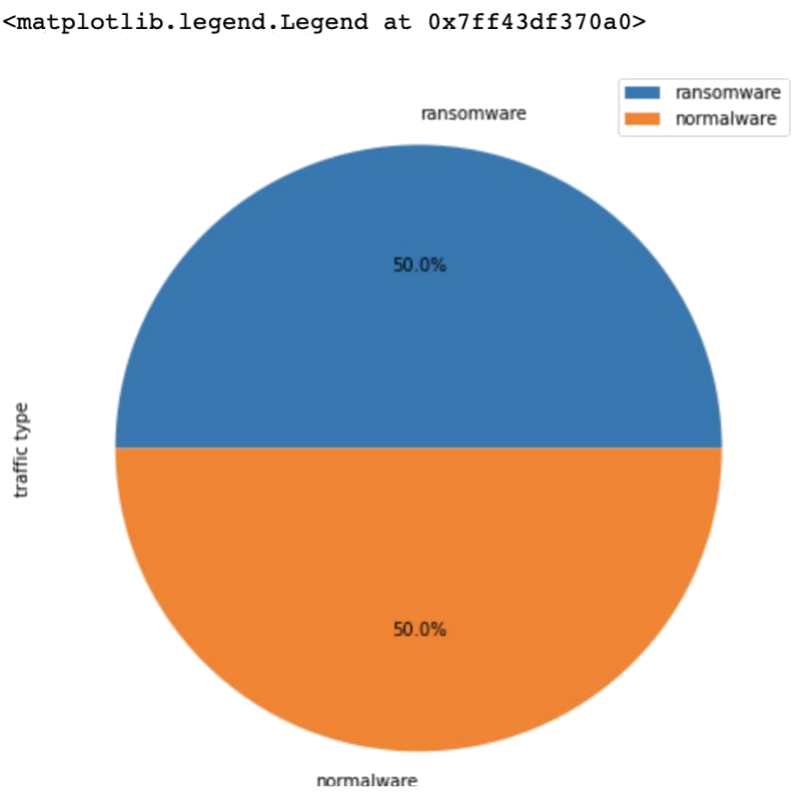

Figure 5: Distribution after Data Balancing

3.3 Data Transformation

This is the method of altering the structure, values, or format of a dataset to retrieve information from it or prepare it for modelling. There are different data transformation steps such as feature engineering, feature reduction, feature selection etc (Kandel et al., 2011). For this research, the following steps were taken to transform the data:

3.3.1 Variance Check

The check for variance in the dataset is a pertinent prerequisite in data modelling. Dataset with low variance are shown to result in models with high bias as explained by (Bond, 2002). The authors also suggest a threshold of at least 70% variance in datasets to mitigate bias in the model. Consequently, this research removed columns which had a variance less than 70%.

3.3.2 Label Encoding

Machine learning typically involves working with datasets that have numerous categories in one or more columns. These categories may be written in words or represented by numbers. The data is frequently labelled in plain English to make it human readable or intelligible.

Label encoding is the process of transforming labels into a numeric form so that they may be read by machines. The operation of those labels can then be better determined by machine

learning techniques (Potdar et al., 2017). All the columns left, after the columns with low variances were dropped, were encoded so they would be machine readable.

3.3.3 Feature Correlation

The measure of relationship between 2 or more features/ variables is known as correlation. Feature correlation in statistics and modelling is used to identify the relationship between the dependent variable and the independent variable(s) (Li et al., 2011). The correlation produces an n-by-n matrix showing the relationship between columns, where n is the number of columns used in the correlation. Each relationship is graded between -1 and +1, with -1 representing a strong negative correlation, 0 representing no correlation, and +1 representing a strong positive correlation. The correlation matrix for this research produced a 12 by 12 matrix with values from 0 to 1. Consequently, columns with no correlation to the traffic type column were dropped from the dataset.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff2c6c0f8e0>
```

Figure 6: Correlation Plot

3.4 Modelling

This step involves defining, training, and testing the model, as well as using the model to make predictions. As discussed in the literature review, some machine learning algorithms have been used in ransomware detection. However, this research seeks to compare the performance of the random forest classifier and the logistic regression classifier in predicting and classifying ransomware.

3.4.1 Logistic Regression Classifier

The logistic regression works by calculating the probability of an occurrence. Typically, a threshold of 0.5 (50%) is set. The variables are traced against a sigmoid curve (S shaped curve on a graph) to determine the probability of occurrence. Whenever the probability is 0.5 or above, the event is said to occur.

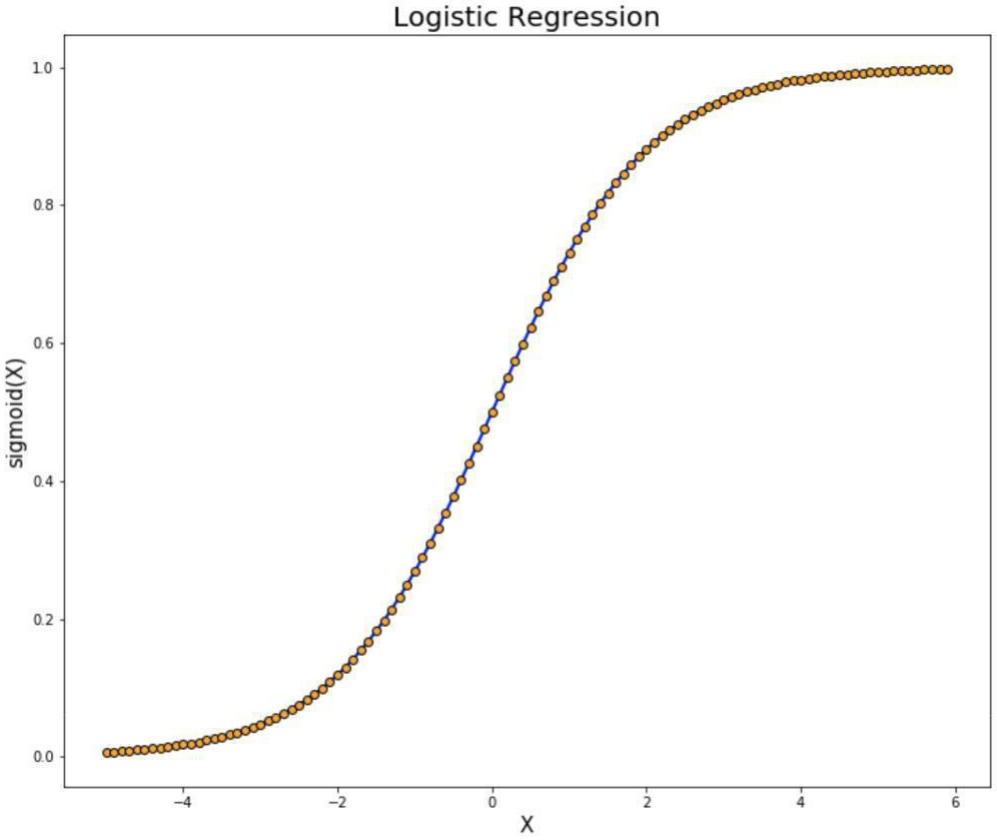

Figure 7: Logistic Regression Classifier

3.4.2 Random Forest Classifier

This works by creating multiple trees with multiple branches like how the decision tree classifier works. Each branch in the tree is called a node, representing the number of possible outcomes in each prediction – in this research, there are only 2 possible outcomes, normalware and ransomware. Each tree analyses each feature in the dataset to make predictions. The majority predicted class is then taken as the final decision for each prediction.

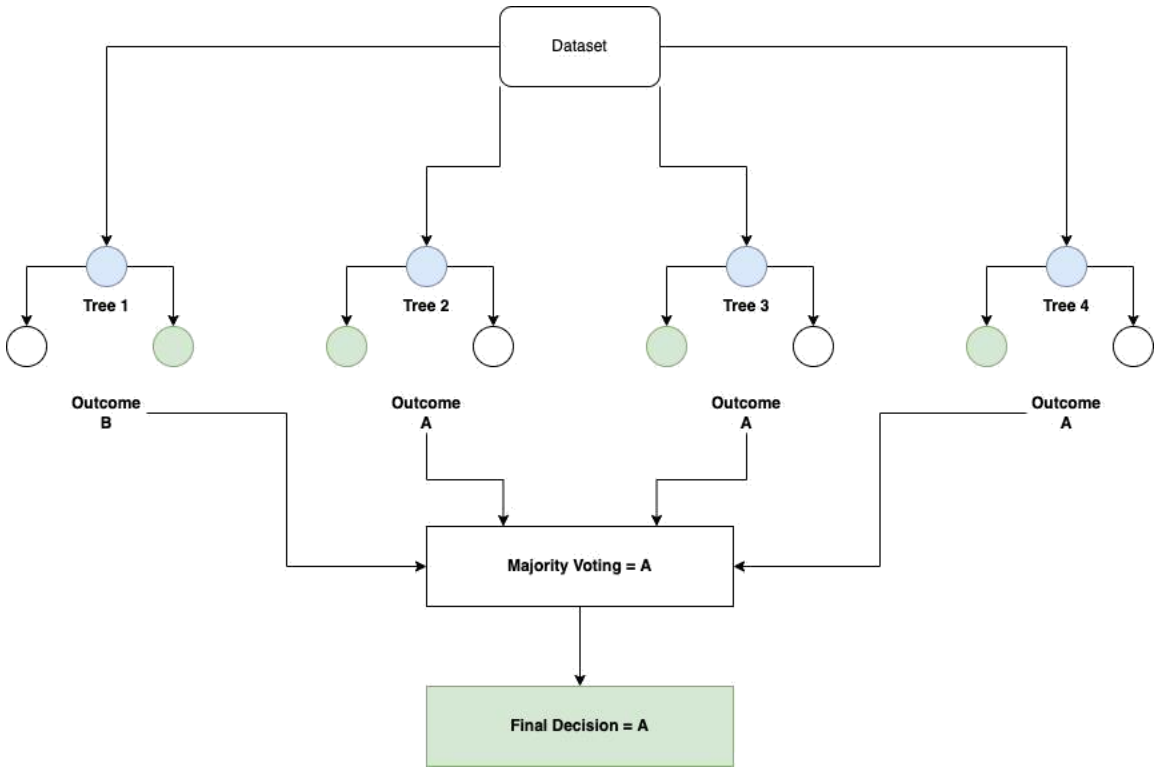

Figure 8: Random Forest Classifier

3.5 Evaluation

To understand the performance and predictive power of a model, it is important to evaluate the models using relevant metrics such as accuracy, precision, and recall. Two (2) main evaluation reports were used in this research. The classification report containing the accuracy, precision and recall of the model for each traffic type. The confusion matrix showing the number of actual outcomes versus the number of predicted outcomes for each traffic type. Both evaluation reports are discussed in detail in the evaluation section or this report. These reports were selected as they are ideal for evaluating the performance of the models as they relate to the objective of the research.

4 Design Specification

This study compares how well normalware and ransomware traffic is detected and predicted using the random forest classifier and the logistic regression classifier. As a result, this section goes over the design specifications used during this project.

Figure 9 below shows a visual representation of proposed detection models from the data collection stage through to the modelling and evaluation. The 2 experiments conducted during this research are the random forest and the logistic regression classifier as stated above. The chosen dataset was cleaned, and prepared, with specific features selected to improve the performance of the proposed models. The processed dataset set were split into 70% which was used to train the models and 30% which was used to test the models.

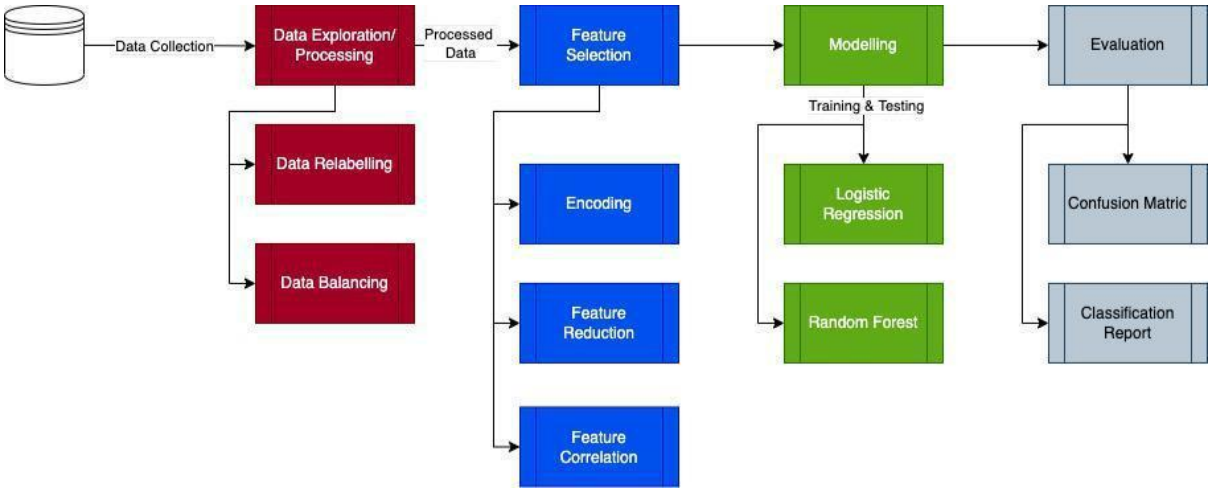

Figure 7: Proposed Design Specification

A detailed explanation of the different steps and sub-steps is explained in the implementation and evaluation sections of this report.

5 Implementation

5.1 Environment and Packages

One of the most often used programming language in machine learning is Python, which is noted for being ideal for dealing with massive data (Raschka, 2015). As a result, Python was chosen as the scripting language for this project. All stages of this research were done in Jupyter Notebook, which is a web-based integrated development environment (IDE) which allows you to write, run and visualise python codes directly from your web browser, and saved in a ipynb extension known as a notebook.

Within this environment, python libraries such as pandas and numpy were used for data processing, the brothon library was used to read the network log files, seaborn and matplotlib were used for data visualisation while libraries under the sklearn package was used for modelling and evaluation.

5.2 Data Exploration

As mentioned above, the brothon python library was used to read and load the network file into the environment. Over 10 million rows and 22 columns were read from the network file. This process, on average, takes around 5-6 minutes each time the script run.

Figure 8: Time Spent Loading Network Files

Consequently, the network files were saved as a csv file for future use. The process of loading the network file which was saved to csv only take around 46 seconds to run, thus, making the more time efficient option to load the network files.

Figure 9: Time Spent Loading Network Files from CSV

The network files were then loaded into a dataframe using the pandas library. The pandas dataframe allows for easily readability and manipulation of data. Also using the pandas dataframe, other pre-processing steps were taken such as checking the shape of the dataframe (the number of rows and columns), checking the distribution of the dataframe (number of benign and malicious rows), relabelling the dataframe and balancing the distribution of the dataframe.

5.2.1 Data Relabelling

This was done as the objective of the research is only concerned with detecting either normalware or ransomware in network traffic. The original dependent variable in the column was named “tunnel_parents label detailed-label” but was later renamed to “traffic type” for clarity. Additionally, the type of traffic for each row was renamed from “benign” and “malicious” to “normalware” and “ransomware” respectively.

5.2.2 Data Balancing

For the data balancing, the number of ransomware and normalware in the dataset. An imbalanced dataset has an adverse effect on the performance of any predictive model. It causes a bias when making predictions. This was explained in detail in section 3.2.2 of this report. A total of 2 million rows were used for training and testing. 1 million rows for ransomware and normalware each.

5.3 Feature Selection

With the use of just pertinent data and the elimination of irrelevant data, feature selection is a technique for lowering the input variable for your model. It involves automatically selecting variables for your machine learning algorithm that are pertinent to the problem you are attempting to solve. Feature selection basically involves the streamlining of the dataset to the

number of columns relevant for the modelling process. The label encoding and feature correlation steps are explained in sections 3.3.2 and 3.3.3 respectively.

5.4 Modelling

For the modelling phase of the project, 2 machine learning algorithms were implemented including the random forest classifier and the logistic regression classifier. Before the modelling commenced, the dependent variable (y) and independent variables (x) were split into the training and testing dataset. 70% of the data (1.4 million rows) were used to train the model, while the remaining 30% of the data (600 thousand rows) were used to test the predictive power of the model.

Evaluation

6.1 Experiment 1: Logistic Regression

The performance of logistic regression model was evaluated using the confusion matrix and classification report, detailing the accuracy, precision and recall of the model.

6.1.1 Confusion Matrix

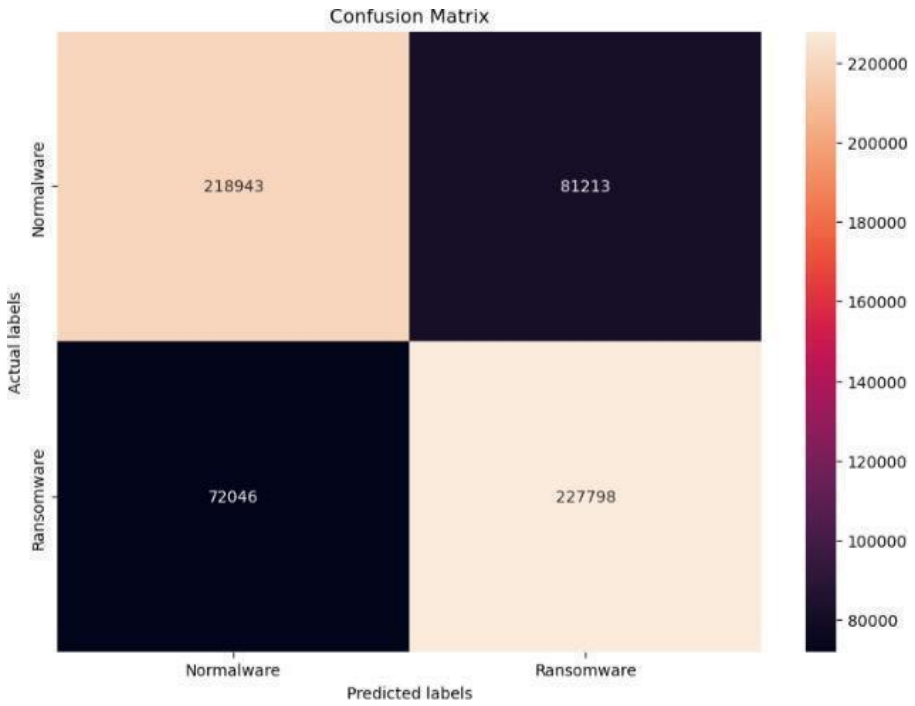

Figure 10: Confusion Matrix for Logistic Regression Experiment

The above figure shows the confusion matrix for logistic regression. The bottom right and top left values can be interpreted as the true positive and true negative values, respectively. While the top right and bottom left values are the false positive and false negative values, respectively. The confusion matrix also reveals that the model has a higher false positive value that false

negative value, therefore the model tends to classify more normalware as ransomware than the reverse.

6.1.2 Classification report

Precision, Recall, F1

```
In [42]: CR = classification_report(y_test, lr_pred, digits = 4, target_names = labels)
         print(CR)
```

	precision	recall	f1-score	support
Normalware	0.7524	0.7294	0.7407	300156
Ransomware	0.7372	0.7597	0.7483	299844
accuracy			0.7446	600000
macro avg	0.7448	0.7446	0.7445	600000
weighted avg	0.7448	0.7446	0.7445	600000

Figure 11: Classification Report for Logistic Regression Experiment

We can see from the above figure, that the model achieved an accuracy of about 74% with a precision of 75% on normalware and 73% on ransomware due to the large number of false positives reported in the confusion matrix.

6.2 Experiment 2: Random Forest

6.2.1 Confusion Matrix

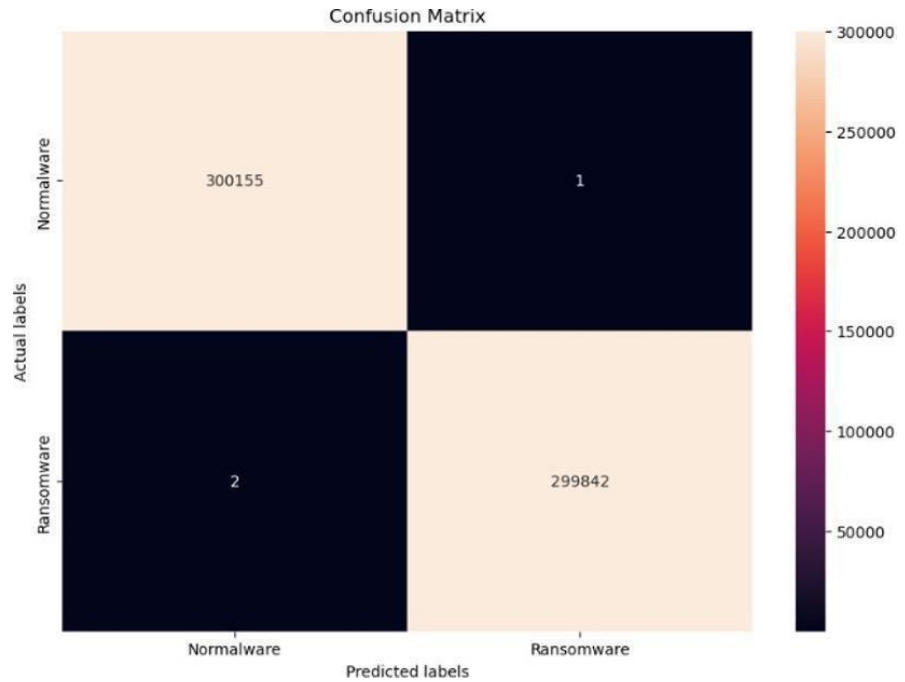

Figure 12: Confusion Matrix for Random Forest Experiment

Figure 14 above shows us the confusion matrix for Random Forest. Like the confusion matrix discussed above, the top left, bottom right, top right, and bottom left values are the true negative, true positive, false positive and false negative values, respectively. The matrix also reveals that random forest model has a lower false positive value compared to the logistic regression model. Overall, the model had only 3 misclassifications.

6.2.2 Classification Report

Precision, Recall, F1

```
In [46]: CR = classification_report(y_test, rf_pred, digits=6, target_names = labels)

         print(CR)
```

```
                precision    recall  f1-score   support

    Normalware   0.999993  0.999997  0.999995    300156
    Ransomware   0.999997  0.999993  0.999995    299844

      accuracy                       0.999995    600000
     macro avg   0.999995  0.999995  0.999995    600000
  weighted avg   0.999995  0.999995  0.999995    600000
```

Figure 13: Classification Report for Random Forest Experiment

The figure above reveals the random forest model achieved a precision of 99.3% on normalware and 99.7% on ransomware and for the recall we got 99.7% on normalware and 99.3% on ransomware. The random forest model outperformed the logistic regression model in terms of accuracy, precision, and recall.

7 Conclusion and Future Works

The aim of the research project was to compare the performance of the logistic regression classifier against the random forest classifier in predicting normalware and ransomware. During the research, 2 models were built using the algorithms mentioned above. The result of the research revealed that the random forest model outperformed the logistic regression model, achieving an accuracy of 99% against 74% from the logistic regression model.

For future projects, researchers may consider the following:

- The use of deep learning algorithms in ransomware detection.
- The use of alternative feature selection and encoding during data processing.
- Training the model with different types of datasets so that the model can predict various types of ransomware and malware in general.

8 References

Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A., 2018. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE access 6, 33789–33795.

Anderson, B., Quist, D., Neil, J., Storlie, C., Lane, T., 2011. Graph-based malware detection using dynamic analysis. J Comput Virol 7, 247–258. https://doi.org/10.1007/s11416-011-0152-x

Arabo, A., Dijoux, R., Poulain, T., Chevalier, G., 2020. Detecting ransomware using process behavior analysis. Procedia Computer Science 168, 289–296.

Aslan, Ö.A., Samet, R., 2020. A comprehensive review on malware detection approaches. IEEE Access 8, 6249–6271.

Basu, I., Sinha, N., Bhagat, D., Goswami, S., 2016. Malware detection based on source data using data mining: A survey. American Journal of Advanced Computing 3, 18–37.

Bazrafshan, Z., Hashemi, H., Fard, S.M.H., Hamzeh, A., 2013. A survey on heuristic malware detection techniques, in: The 5th Conference on Information and Knowledge Technology. Presented at the 2013 5th Conference on Information and Knowledge Technology (IKT), IEEE, shiraz, Iran, pp. 113–120. https://doi.org/10.1109/IKT.2013.6620049

Bond, S., 2002. Dynamic panel data models: a guide to microdata methods and practice (Working Paper Series), Working Paper Series. https://doi.org/10.1920/wp.cem.2002.0902

Chang, Y., Li, W., Yang, Z., 2017. Network intrusion detection based on random forest and support vector machine, in: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). IEEE, pp. 635–638.

Chio, C., Freeman, D., 2018. Machine learning and security: protecting systems with data and algorithms, First edition. ed. O'Reilly Media, Sebastopol, CA.

Chittooparambil, H.J., Shanmugam, B., Azam, S., Kannoorpatti, K., Jonkman, M., Samy, G.N., 2019. A Review of Ransomware Families and Detection Methods, in: Saeed, F., Gazem, N., Mohammed, F., Busalim, A. (Eds.), Recent Trends in Data Science and Soft Computing, Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 588–597. https://doi.org/10.1007/978-3-319-99007-1_55

Choo, K.-K.R., 2011. The cyber threat landscape: Challenges and future research directions. Computers & security 30, 719–731.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM 39, 27–34.

Fernando, D.W., Komninos, N., Chen, T., 2020. A study on the evolution of ransomware detection using machine learning and deep learning techniques. IoT 1, 551–604.

Firdausi, I., Erwin, A., Nugroho, A.S., 2010. Analysis of machine learning techniques used in behavior-based malware detection. Presented at the 2010 second international conference on advances in computing, control, and telecommunication technologies, IEEE, pp. 201–203.

Garcia, S., Parmisano, A., Erquiaga, M.J., 2020. IoT-23: A labeled dataset with malicious and benign IoT network traffic. https://doi.org/10.5281/ZENODO.4743746

Ghafir, I., Prenosil, V., 2014. Advanced persistent threat attack detection: an overview. Int J Adv Comput Netw Secur 4, 5054.

Gjerloev, J.W., 2012. The SuperMAG data processing technique: TECHNIQUE. J. Geophys. Res. 117, n/a-n/a. https://doi.org/10.1029/2012JA017683

Ham, H.-S., Choi, M.-J., 2013. Analysis of android malware detection performance using machine learning classifiers. Presented at the 2013 international conference on ICT Convergence (ICTC), Ieee, pp. 490–495.

Hama Saeed, M.A., 2020. Malware in Computer Systems: Problems and Solutions. IJID 9, 1. https://doi.org/10.14421/ijid.2020.09101

Heena, 2021. Advances In Malware Detection- An Overview. https://doi.org/10.48550/ARXIV.2104.01835

Hossain Faruk, M.J., Shahriar, H., Valero, M., Barsha, F.L., Sobhan, S., Khan, M.A., Whitman, M., Cuzzocrea, A., Lo, D., Rahman, A., Wu, F., 2021. Malware Detection and Prevention using Artificial Intelligence Techniques, in: 2021 IEEE International Conference on Big Data (Big Data). Presented at the 2021 IEEE International Conference on Big Data (Big Data), IEEE, Orlando, FL, USA, pp. 5369–5377. https://doi.org/10.1109/BigData52589.2021.9671434

IDC's 2021 Ransomware Study: Where You Are Matters! [WWW Document], n.d. . IDC: The premier global market intelligence company. URL https://www.idc.com/getdoc.jsp?containerId=US48093721 (accessed 12.4.22).

Idika, N., Mathur, A.P., 2007. A Survey of Malware Detection Techniques 48.

Kandel, S., Paepcke, A., Hellerstein, J., Heer, J., 2011. Wrangler: interactive visual specification of data transformation scripts, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Presented at the CHI '11: CHI Conference on Human Factors in Computing Systems, ACM, Vancouver BC Canada, pp. 3363– 3372. https://doi.org/10.1145/1978942.1979444

Kapoor, A., Gupta, A., Gupta, R., Tanwar, S., Sharma, G., Davidson, I.E., 2021. Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions. Sustainability 14, 8. https://doi.org/10.3390/su14010008

Khammas, B.M., 2020. Ransomware detection using random forest technique. ICT Express 6, 325–331.

Kirda, E., 2017. UNVEIL: A large-scale, automated approach to detecting ransomware (keynote), in: 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER). Presented at the 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, Klagenfurt, Austria, pp. 1–1. https://doi.org/10.1109/SANER.2017.7884603

Kolodenker, E., Koch, W., Stringhini, G., Egele, M., 2017. Paybreak: Defense against cryptographic ransomware. Presented at the Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 599–611.

Kolter, J.Z., Maloof, M.A., 2004. Learning to detect malicious executables in the wild, in: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '04. Presented at the the 2004 ACM SIGKDD

international conference, ACM Press, Seattle, WA, USA, p. 470. https://doi.org/10.1145/1014052.1014105

Kumar, A., Lim, T.J., 2019. EDIMA: Early detection of IoT malware network activity using machine learning techniques. Presented at the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), IEEE, pp. 289–294.

Li, B., Wang, Q., Hu, J., 2011. Feature subset selection: a correlation-based SVM filter approach. IEEJ Trans Elec Electron Eng 6, 173–179. https://doi.org/10.1002/tee.20641

Matin, I.M.M., Rahardjo, B., 2019. Malware Detection Using Honeypot and Machine Learning, in: 2019 7th International Conference on Cyber and IT Service Management (CITSM). Presented at the 2019 7th International Conference on Cyber and IT Service Management (CITSM), IEEE, Jakarta, Indonesia, pp. 1–4. https://doi.org/10.1109/CITSM47753.2019.8965419

Mohaisen, A., Alrawi, O., Mohaisen, M., 2015. AMAL: high-fidelity, behavior-based automated malware analysis and classification. computers & security 52, 251–266.

Moore, C., 2016. Detecting ransomware with honeypot techniques.

Mujumdar, A., Masiwal, G., Meshram, B.B., 2013. Analysis of signature-based and behavior-based anti-malware approaches. International Journal of Advanced Research in Computer Engineering and Technology 2, 2037–2039.

Number of ransomware attacks per year 2022 [WWW Document], n.d. . Statista. URL https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/ (accessed 12.3.22).

Olawale Surajudeen, A., 2012. Malware Detection, Supportive Software Agents and Its Classification Schemes. IJNSA 4, 33–49. https://doi.org/10.5121/ijnsa.2012.4603

Potdar, K., S., T., D., C., 2017. A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. IJCA 175, 7–9. https://doi.org/10.5120/ijca2017915495

Raschka, S., 2015. Python machine learning. Packt publishing ltd.

Ren, Z., Wu, H., Ning, Q., Hussain, I., Chen, B., 2020. End-to-end malware detection for android IoT devices using deep learning. Ad Hoc Networks 101, 102098.

Security, H.N., 2022. Ransomware attacks, and ransom payments, are rampant among critical infrastructure organizations. Help Net Security. URL https://www.helpnetsecurity.com/2022/02/10/critical-infrastructure-ransomware/ (accessed 12.4.22).

Singh, A., Ikuesan, R.A., Venter, H., 2022. Ransomware Detection using Process Memory. arXiv preprint arXiv:2203.16871.

Takeuchi, Y., Sakai, K., Fukumoto, S., 2018. Detecting ransomware using support vector machines. Presented at the Proceedings of the 47th International Conference on Parallel Processing Companion, pp. 1–6.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. Presented at the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), IEEE, Ottawa, ON, Canada, pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528

Team, C., n.d. Report: Ransomware Attacks and the True Cost to Business 2022 [WWW Document]. URL https://www.cybereason.com/blog/report-ransomware-attacks-and-the-true-cost-to-business-2022 (accessed 12.3.22).

Tseng, A., Chen, Y., Kao, Y., Lin, T., 2016. Deep learning for ransomware detection. IEICE Technical Report; IEICE Tech. Rep. 116, 87–92.

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Venkatraman, S., 2019. Robust Intelligent Malware Detection Using Deep Learning. IEEE Access 7, 46717–46738. https://doi.org/10.1109/ACCESS.2019.2906934

Wecksten, M., Frick, J., Sjostrom, A., Jarpe, E., 2016. A novel method for recovery from Crypto Ransomware infections, in: 2016 2nd IEEE International Conference on Computer and Communications (ICCC). Presented at the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), IEEE, Chengdu, China, pp. 1354–1358. https://doi.org/10.1109/CompComm.2016.7924925

Wu, Y., Wei, D., Feng, J., 2020. Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey. Security and Communication Networks 2020, 1–17. https://doi.org/10.1155/2020/8872923

Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., Sangaiah, A.K., 2019. Classification of ransomware families with machine learning based onN-gram of opcodes. Future Generation Computer Systems 90, 211–221.

Zhang, J., Zulkernine, M., 2006. A hybrid network intrusion detection technique using random forests. Presented at the First International Conference on Availability, Reliability and Security (ARES'06), IEEE, pp. 8-pp.