National College of Ireland

Can Semi Supervised Feature Selection improve Ransomware Detection

MSc Research Project
Masters in Cyber Security

Gavin Smyth
Student ID: x16354406

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Gavin Smyth

Student ID: X16354406

Program: Masters in Cyber Security **Year:** 2022

Module: Research Project/Internship

Supervisor: Vikas Sahni

Submission Due Date: 15 August 2022

Project Title: Using Semi-Supervised Feature Selection for ransomware Detection

Word Count: 7121 **Page Count:** 20

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: Gavin Smyth

Date: 03/08/2022

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):	

Can Semi Supervised Feature Selection improve Ransomware Detection

Gavin Smyth
X16354406

Abstract

Today, ransomware is one of the most harmful cybersecurity threats that organizations and people face especially with the expanse of organisations attack surface with employees working remotely. Machine learning has proven to be extremely helpful in ransomware detection, although, this requires a huge amount of labelled data for training and categorizing data takes time and money. However, there is a huge amount of unlabelled data. Semi-supervised learning, which uses a small number of labelled data and a large number of unlabelled data for learning, can be used to address this issue. This also encourages academics to create semi-supervised feature selection techniques that assesses feature relevance using both labelled and unlabelled data. Although researchers have proposed a variety of feature selection methods combined with Semi Supervised learning, this paper attempts to analyse different Semi Supervised feature selection and semi-supervised classification methods applied to the CICAndMal 2017 dataset. Semi JMI, Semi MIM and Semi IAMB were applied to different Semi Supervised classification models and the accuracy measured. Analysis on the subsets determine that Semi JMI outperformed Semi MIM and Semi IAMB with an average accuracy of 85% when datasets are balanced and again Semi JMI performed on the Overall dataset with an average accuracy of 73%. Therefore, Semi Supervised feature selection combined with Semi Supervised classification methods can be considered for future research in detecting ransomware.

1 Introduction

1.1 Background

Ransomware is a malware which usually encrypts the important or confidential data of a system in demand of a ransom. These types of attacks are usually conducted using various social engineering tricks used to allure/force a user to click on a malicious link in an email or through other methods. Ransomware attacks have experienced a resurgence, with recent attacks focused on international healthcare, local government, and education sectors, in particular.

68% of ransomware attacks go unnoticed according to a report by US cybersecurity provider FireEye (Adamov and Carlsson, 2020). This draws the cybersecurity experts' attention to this problem. The rate of cyberattacks has only increased as a result of the COVID-19 epidemic. Attackers used phishing emails with COVID-19-themed ransomware to trick people into paying the ransom as the paradigm of the workplace transitioned to home-based scenarios, leading to weaker security safeguards.

For example, many phishing campaigns prompted users to click on specific links to get sensitive information related to a COVID-19 vaccine, shortage of surgical masks, etc. False COVID-19-related information was effectively used as a hook by attackers to start more effective phishing attempts. Another issue that drives people to engage in cybercrime, such as initiating ransomware attacks and interrupting vital IT services, in order to support themselves is higher unemployment rates (Beaman *et al.*, 2021).

Traditional ransomware defence methods like back-ups are proving to be less effective. Ransomware frequently avoids detection because they operate without a trusted machine identity. New attack methods include exploiting customers directly with stolen data, selling stolen data on the dark web and letting customers know their data is stolen. A quick google search on ransomware will show just how common and just how much ransomware can cost organisations each year.
The increase in frequency and sophistication of ransomware has led people to look for new ways to detect ransomware. As victims are willing to pay, more threat actors have joined this growing field, bringing innovation, creativity, and more sophisticated attack methods, from ransomware-as-a-service to triple extortion (Alhawi, Baldwin and Dehghantanha, 2018).
Try as they might, companies can't avoid ransomware forever. Eventually, attackers will get into an enterprise system. The goal then becomes detecting ransomware before it encrypts and exfiltrates business-critical data.
Today, machine learning-based ransomware detection has received a great amount of attention from researchers. Most methodologies focus on supervised learning that requires vast amounts of labelled data which can be hard to obtain.. To combat this, semi-supervised learning techniques can be used to learn a model from a large amount of unlabelled data and a small number of labelled training data. The same applies for feature selection as stated previously, collecting labelled data can be challenging in many real-world applications, yet unlabelled data is widely available and simple to access. This encourages researchers to create semi-supervised feature selection techniques that assess feature relevance using both labelled and unlabelled data (Sheikhpour *et al.*, 2017).

1.2 Research Question and Motivation

This paper focuses on addressing the following research question: ***Can Semi Supervised Feature Selection improve Ransomware Detection***

Motivation to conduct this paper occurred daily as ransomware attacks happen all the time and can hit close to home. Take for example National college of Ireland who's IT systems fell victim to a ransomware attack which resulted in the suspension of access to all IT systems while service providers worked to restore services or the HSE who suffered a major ransomware attack during the Covid 19 pandemic that caused all of its IT systems nationwide to shut down, both of which only occurred last year in 2021 (Board, 2021) . The evolution of cyber criminals shows just how cruel and just how ruthless they can be. This motivates not just this paper but other researchers to find new ways to help mitigate this type of crime. In saying this, there seems to be a research gap between proposed machine learning solutions to ransomware detection and this papers solution.
The primary motive for this project was to implement Semi Supervised feature selection into Semi Supervised learning methodologies to see if this cheaper and less time consuming method of machine learning could be beneficial for ransomware detection. As ransomware is constantly evolving, it makes sense to look for new ways to combat this issue especially with new types of ransomware who's presence may only be known through key attributes and not signatures/labels. This paper aims to highlight the need to constantly conduct new methods of detection for the ever changing implantation of ransomware.

2 Related Work

The subsections that follow provide an academic literature review related to previous approaches similar to detecting ransomware. The covered areas are ransomware detection approaches, supervised, semi-supervised and unsupervised learning. Also covered is semi-supervised feature selection to create a subset of relevant features for use in model construction.

2.1 Ransomware Detection Approaches

Ransomware detection techniques can be categorized as behaviour-based, I/O request packet monitoring, network traffic monitoring, API call monitoring, at the storage level, in android devices, and other techniques. Most ransomware detection methods rely on the machine learning approach as it wields the power of prediction (Bijitha, Sukumaran and Nath, 2020).

Several papers have classified ransomware groups and described their general behavior. Antivirus and firewall software detection algorithms have been developed due to these discoveries(Berrueta *et al.*, 2019). As ransomeware is constantly evolving and becoming more sophisticated, new ways in which these more advanced forms of ransomeware can be detected are being explored. As the malware evolves so should detection. Most ransomware detection solutions are installed locally on a user's computer and work similarly to antivirus software in detecting and blocking ransomware activities. However, there are proposals based on detecting ransomware network activity or limiting network traffic to the malware's required servers. These detection systems are based on a variety of ad-hoc heuristics and AI techniques (Berrueta *et al.*, 2019).

Misuse and anomaly detection methods are two types of ransomware detection approaches. Anomaly detection approaches model the typical behavior of the system and alert when any violation happens, whereas misuse detection methods use known ransomware signatures (Noorbehbahani, Rasouli and Saberi, 2019).

2.1.1 Static Based Approach

Static-based Approach is another approach to ransomware detection and focuses more on misuse detection. Malware detection utilizing Static-based analysis entails examining an application's code before it is executed to see if it is capable of malicious behavior. The executable will be prevented from launching if the static analysis detects any dangerous code.

Signature analysis is the most prevalent sort of static analysis and is commonly employed in commercial virus scanners. Signature analysis involves extracting code string patterns (signatures) from the target application's code and comparing them to a database of known harmful code patterns.

Signature-based detection is based on a massive database of harmful code signatures. To stay current, this repository must be updated on a regular basis, which is not an easy undertaking (Nieuwenhuizen, 2017).

2.1.2 Behavioural based approach

Behavioral Based Approach is a dynamic approach to detecting rasomware and focuses more on Anomaly detection. Dynamic-based analysis detection requires the continuous monitoring of processes to see if any of them are acting maliciously. Any process that is acting maliciously will be reported as hazardous and terminated (Nieuwenhuizen, 2017).

2.1.3 Machine Learning

Machine Learning is by far the most common technique when it comes to ransomware detection and researching techniques for ransomware detection. Machine learning can be used for both above approaches. There are three types of techniques when it comes to machine learning. These are:

- **Unsupervised Learning, Supervised Learning, Semi-supervied Learning.**

(Sgandurra *et al.*, 2016) presents EldeRan, a machine learning method for classifying and detecting ransomware. To identify ransomware, this approach dynamically monitors the actions made by software as they are installed. They tested their strategy on a dataset that included 582 ransomware and 942 benign occurrences. Machine learning proved to be successful in detecting ransomware and its new variations, according to the findings. Mutual information was used to choose features, and

regularized logistic regression was used to train and update the model. Despite the fact that their proposed method is effective in detecting Ransomware, the assessment dataset appears to be relatively limited, and the proposed method needs be evaluated on fresh ransomware datasets.

There are also papers that combine the two approaches. (Shijo and Salim, 2015) presents a method for analyzing and classifying an unknown executable file that combines static and dynamic analysis. The technology employs machine learning, with training data consisting of known malware and benign programs. The feature vector is chosen after examining both the binary code and the dynamic behavior. The suggested method takes advantage of both static and dynamic analysis, resulting in increased efficiency and classification accuracy. The testing results reveal that the static approach is 95.8% accurate, the dynamic method is 97.1 percent accurate, and the integrated method is 98.7% accurate.

2.2 Supervised, Unsupervised and Semi Supervised Learning

As mentioned above Unsupervised, Supervised and Semi-supervised learning are the three techniques of machine learning used in malware/ransomware detection.

2.2.1 Supervised Learning

Supervised Learning techniques require sufficient labelled training data that is expensive and time-consuming to obtain them. A comparable technique to supervised classification is signature-based detection, which uses examples of known malware to develop a classification model that identifies the known risks from other applications. In the same way that signature-based detection fails to discover new and developing malware, supervised classification systems do as well. Furthermore, due to the diversity of malware classes, their unequal distribution, and data imperfection issues (noise, missing values, and correlated characteristics) that continue to hinder the adoption of increasingly sophisticated learning algorithms, which means creating an effective classification model is difficult (Comar *et al.*, 2013).

2.2.2 Semi Supervised Learning

Semi supervised learning is a field of machine learning that focuses on performing particular learning tasks using both labelled and unlabelled data. It allows leveraging the massive amounts of unlabelled data available in many use cases in combination with typically smaller sets of labelled data. It is conceptually located between supervised and unsupervised learning. In recent years, research in this area has largely followed the broader trends in machine learning, with a focus on neural network-based models and generative learning (van Engelen and Hoos, 2020). Also, there have been studies in how this type of machine learning can be used to detect malware/ransoware (Santos, Nieves and Bringas, 2011) proposes a new method for detecting unknown malware that uses a semi-supervised learning approach. The Learning with Local and Global Consistency (LLGC) technique was used by the authors, which is a semi-supervised classification algorithm. Their approach is to train a classifier utilizing a set of labeled (malware and legitimate software) and unlabeled data. Using 50% labeled data, they were able to attain an accuracy of 0.86. (Noorbehbahani and Saberi, 2020) proves that semi supervised learmning in bennificial to ransomware detection using wrapper classification and random farrest feature selection leading to 69.50% accuracy rate. Within the conclusion states it is necessary to investigate and propose a semi-supervised feature selection method for ransomware detection in future works and this is the basis of this research projerct.

2.2.3 Unsupervised Learning

Unsupervised learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data. In recent years, statistical and unsupervised learning techniques for anomaly detection have received a lot of attention. The ability to detect zero-day attacks is a key advantage of anomaly-based detection. However, it has a high false

alarm rate, which means that a huge number of good programs will be mistakenly detected as harmful (Comar *et al.*, 2013).
Unsupervised learning takes unlabelled data and analyses and clusters it using machine learning techniques (HSU, Levine and Finn, 2019). Clustering, association, and dimensionality are the three primary issues with unsupervised learning. Unlabelled data is grouped using clustering algorithms based on similarities and contrasts. In order to detect links between variables, association algorithms use rules. When a dataset has a large number of features, dimensionality methods are used to reduce the number of inputs while keeping data integrity; this process is frequently employed for data pre-processing. Both supervised and unsupervised learning methods are used in the previous research covered in this literature review.

2.3 Semi-Supervised Feature Selection

In data mining and machine learning applications, feature selection is a significant task that eliminates unnecessary and redundant features while also improving learning performance (Sheikhpour *et al.*, 2017). Feature selection can be used to create the static feature set.
A semi-supervised feature selection integrates a small amount of labelled data into unlabelled data as additional information to improve the performance of an unsupervised feature selection. Recently, increasing attention has been directed to the study of semi-supervised feature selection, and hence, many semi-supervised feature selection methods (Sheikhpour *et al.*, 2017). According to the perspective of the first taxonomy, semi-supervised feature selection methods can be categorized into filter, Wrapper and Embedded. Then, each category is divided into smaller categories based on the procedures used for semi-supervised feature selection. (Sheikhpour *et al.*, 2017). In paper (Sechidis and Brown, 2018) a simple strategy is used to perform semi supervised feature selection. This paper shows that the approaches taken provide powerful results for feature selection, via hypothesis testing and feature ranking. They derive two unique algorithms (Semi-JMI, Semi-IAMB) from their methodology and some "soft" prior knowledge of the domain, which beat much more complex competing approaches, demonstrating notably high performance when the labels are missing-not-at-random (Sheikhpour *et al.*, 2017).
There is a lack of literature related to the testing of semi supervised feature selection combined with semi supervised classification for detecting ransomware.
This paper tests the use of semi-supervised feature selection in a semi-supervised learning environment for ransomware detection. Accuracy of detection is compared with previous works of similar approaches.

3 Research Methodology

For this research project the methodology which was chosen was Knowledge Discovery in Databases methodology. This roadmap-style methodology emphasizes the importance of the early stages of the KDD process and demonstrates how careful preparation may lead to a successful and well-managed project (Debuse *et al.*, 2001).

3.1 Step 1 – Data Selection

This section acts upon a database of compiled data the targeted data is determined, and variables that will be used to evaluate for knowledge discovery are determined (Azevedo and Santos, 2008). The dataset chosen for this research project was the CIC-AndMal2017 dataset. CIC-AndMal2017 is an android malware dataset which contains both malware and benign applications which can be used for security testing and malware prevention (Lashkari *et al.*, 2018).
CIC-AndMal2017 dataset has collected advanced malware samples that are able to detect the emulator environment. They gathered about 10,854 samples from various sources (4,354 malware and 6,500 benign). The benign data was collected from Googleplay market published in 2015, 2016, 2017.

The samples contain 42 unique malware families The malware collected can be split into 4 different categories: **Adware, Ransomware, Scareware, SMS Malware**.
For the purpose of this research paper I chose to use the ransomware category as my dataset and a benign dataset chosen at random. The ransomware category contains 10 different families:
Charger family, Jisut family, Koler family, LockerPin family, Simplocker family, Pletor family, PornDroid family, RansomBO family, Svpeng family, WannaLocker family.
All traffic which is not considered Normal or does not fall under the categories previously described, is considered Benign data.

3.2 Step 2 – Data Pre Processing

This stage consists on the target data cleaning and pre-processing in order to obtain consistent data. Predictive models for unreliable data are created in order to forecast similarly faulty, missing, and attributional mismatched data in the future, and then to work it out of future processes (Lashkari *et al.*, 2018).
This is one of the most important steps when it comes to data as the preparation of your data can have a huge impact on the performance of a machine learning algorithm (Kotsiantis, Kanellopoulos and Pintelas, 2006).
For the purpose of this research project the first step of the data pre-processing was to split the dataset into subsets based on their families as previously described. Next the same number of benign instances were added to each ransomware dataset. Each benign dataset was chosen at random. This means that there was now 10 datasets:
Charger family/Benign1, Jisut family/Benign2, Koler family/Benign3, LockerPin family/Benign4, Simplocker family/Benign5, Pletor family/Benign6, PornDroid family/Benign7, RansomBO8 family/Benign, Svpeng family/Benign9, WannaLocker family/Benign10.
Finally I created a dataset with all 10 that were previously created, combined. Therefore there were 11 datasets in total. Each data entry was either labelled 0 meaning ransomware or 1 meaning Benign. Also within this step irrelevant data was removed (flow ID, Source IP, Destination IP, Time Stamp and fwd header length).

3.3 Step 3 – Transformation Stage

This stage consists on the transformation of the data using dimensionality reduction or transformation methods. As the data within the dataset had different attribute values and in very large intervals, all continues features were normalized [0,1].
Once all this was done I attempted to dimensionality reduce the data using feature selection techniques, specifically semi supervised feature selection techniques specified in (Sechidis and Brown, 2018).

The first approach I used for semi-supervised feature selection is a hypothesis testing approach with Markov Blanket discovery, using the IAMB algorithm but with semi-supervised nodes in the Bayesian network. This is referred to as:

- Semi-IAMB

The second attempt is a feature ranking approach to semi supervised feature selection. (Bennasar, Hicks and Setchi, 2015) propose a feature selection method called Joint Mutual Information (JMI). In this method, the candidate feature that maximises the cumulative summation of Joint Mutual Information with features of the selected subset is chosen and added to the subset. This method is reported to perform well in terms of classification accuracy and stability. The Semi supervised algorithm using surrogate variables in an informed way, naturally extends this to semi-supervised scenarios.
JMI is the switching procedure applied to the below semi supervised feature selection:

- Semi-JMI

Another Ranking approach to semi supervised feature selection stated in (Sechidis and Brown, 2018) is:

- Semi-MIM

This feature selection follows mutual information scoring criterion. This Semi supervised feature selection approach uses MIM. MIM adopts mutual information to measure each feature's relevancy to the class label, which does not consider redundancy and complementariness among features.

3.4 Step 4 – Data Mining Stage

The 3 Semi supervised learning algorithms I chose for the data mining stage were:
YATSI RF - YATSI (Yet Another Two-Stage Idea) is a collective classifier that uses the basic classifier to learn and identify the unlabelled data (called pre-labelled data). Following that, the test instances are categorised using the kNN with the actual training set and pre-labelled data (Noorbehbahani and Saberi, 2020).
Collective IBK - To determine the best k in the training set, Collective IBK (Bennasar, Hicks and Setchi, 2015) uses the IBK algorithm (a kNN-based approach). Then, for each test instance, it discovers the k-nearest instances from the training and test sets, which are sorted by distance from the test instance. The differences in class occurrences are used to rank the neighbourhoods. The highest-ranking unlabelled test instance is then classified by a majority vote (Noorbehbahani and Saberi, 2020).
Collective Wrapper (RF) - A supervised base classifier is used in the collective wrapper semi-supervised approaches. The classifier is first trained on the labelled data, and then its predictions are used to produce more labelled data in order to retrain the classifier (Noorbehbahani and Saberi, 2020). In this case we are using random forest.
All three learning algorithms were combined with the combined datasets and the results were analysed in the Evaluation stage.

3.5 Step 5 – Evaluation Stage

The evaluation stage is where the results from the classification models are analysed
After the previous stages the results are evaluated and analysed using k-fold cross-validation and the accuracy measured.
The performance of each model was based on how accurate they were. Graphs were used to display the accuracy.

4 Design Specification

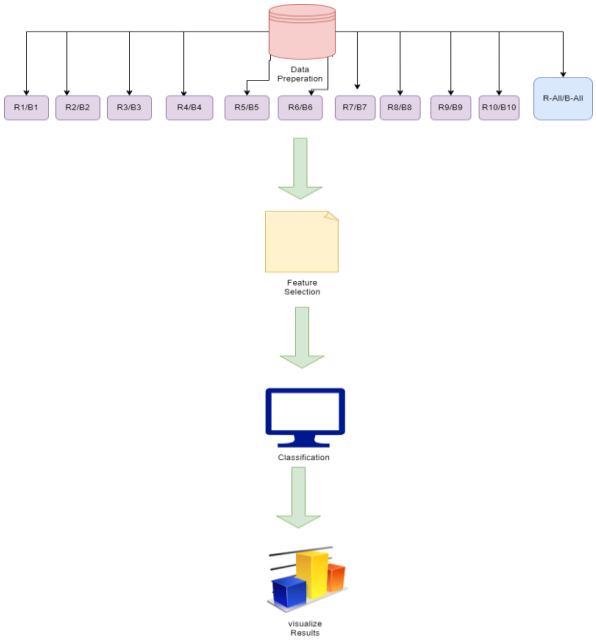

Figure 2. Four key design stages

The design of this research can be separated into four key stages such as can be seen in the figure above.

4.1 Data Preparation

This stage consists on the target data cleaning and pre-processing in order to
obtain consistent data as mentioned above. Predictive models for unreliable data are created in order to forecast similarly faulty, missing, and mismatched data in the future, and then to work it out of future processes (Lashkari *et al.*, 2018). This way we can come up with the target data.

As mentioned before the ransomware dataset was spilt into the 10 different families. Then 10 benign datasets were chosen at random and processed so that the number of rows matched the number of rows contained in each ransomware sub dataset. The benign data was then combined with the ransomware data. Therefore, 10 ransomware-family/benign balanced datasets (R1/B1 to R10/B10) were formed. Finally I created a dataset containing all the ransomware families and benign data combined. All instances were relabelled to 0(ransomware) or 1(benign).
As mentioned previously irrelevant features were eliminated from the datasets (flow ID, Source IP, Destination IP, Time Stamp, fwd header length). I then normalized all continues features within each dataset[0,1].

4.2 Feature Selection

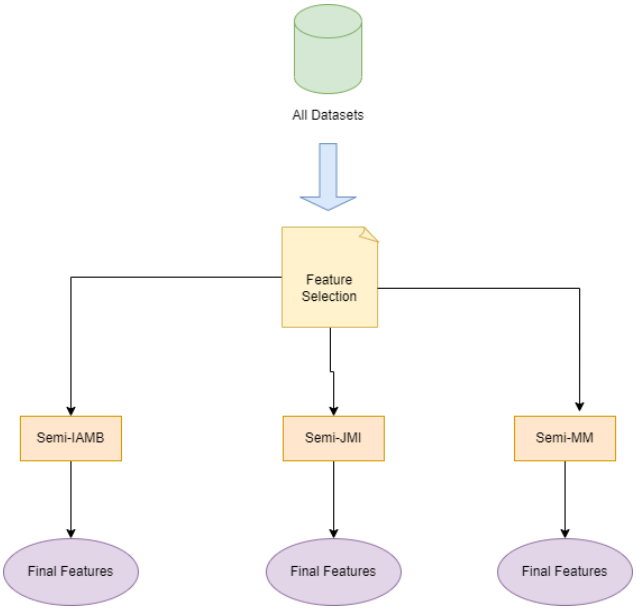

Figure 3. Semi Supervised Feature Selection

Figure 3 above, shows the Semi Supervised feature selection approach for this research. The feature selection process is done using MatLab. Each dataset will be run through each feature selection technique. Steps taken for each semi supervised feature selection technique

- Each dataset is run through all the feature selection techniques
- Each ransomware families key features are combined
- These combined features are then run through all three techniques again to form the final features model.

This is a multilayer feature selection approach. The first step running all datasets through the three algorithms and the second steps running the combined selected features through the three algorithms again making it more of a refined selection. After the feature selection process there should be 11 datasets with the chosen features for each algorithm.

4.3 Classification

After the Selection stage comes the classifications stage. As mentioned previously the learning methods used are semi supervised. The three methods I chose were:
YATSI(RF), Collective IBK, Collective Wrapper(RF)

Random Forrest is used as the base classifier for YATSI and Wrapper as it outperforms the other methods that are implemented with these two algorithms (Noorbehbahani and Saberi, 2020).

4.4 Visualize Results

The evaluation of the classification models is graphically represented in the visualisation step as a graph. The Research project will evaluate the effectiveness of detecting ransomware using semi supervised feature selection methods within semi supervised learning.
As stated previously the results are evaluated and analysed using k-fold cross-validation and the accuracy measured and the performance of each model will be based on how accurate they are.

5 Implementation

Many different tools were used in the implementation of this research project. Each step within the implementation was essential for the project to work.
MATLAB programming language, python and the Weka application was used to complete the implementation of this project. MATLAB platform was used to preform feature selection on the given data files.
Python was used for the normalization of the continuous data. This was done using the Google Collaboration platform. This is a cloud based application which means the programming for the normalization step could be accessed from any machine.
Finally Weka was used for the classification step of this project. Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.

The implementation of this project can be broken into five key steps. These include:
Data Preparation stage, Feature Selection stage, Normalization stage, Classification stage, Evaluation stage

5.1 Data Preparation Implementation Stage

This was the first stage undertaken for the implementation. This stage was essential as the way data is prepared can have a big impact in the results given at the end from a machine learning algorithm.
The first step in the data preparation stage was to remove unwanted columns. The columns previously mentioned were removed as they had no impact on the end result.
Once the unwanted columns were removed, the datasets were created. Each ransomware family was combined with Benign data to create 10 different ranomware/benign subsets. Also, All ransomware families and benign data were combined to form an overall dataset. This meaning there were 11 different datasets used for the implementation of this project.
Each dataset was split within excel, one page being data and the other page being labels(1 for ransomware and 2 for benign). This step was performed as the .csv files needed to be converted to .mat files containing 2 variables (X_data and Y_labels).

5.2 Feature Selection Implementation Stage

The code used for feature selection was taken from (Sechidis and Brown, 2018). The Feature selection stage was conducted using the MATLAB platform. Each dataset was added to MATLAB and converted to variables which were added to .mat files.
There were three different Semi-Supervised feature selection algorithms implanted. These include:

- Semi MIM
- Semi JMI
- Semi IMAB

The Semi-JMI and SEMI-MIM algorithms use a switching procedure that is applied to feature ranking (Sechidis and Brown, 2018). Both algorithms are Semi-supervised filter feature selection methods. These algorithms use "soft" prior knowledge in the analysis to choose the best surrogate to employ when ranking the attributes. When the surrogate is chosen, they both use they can use MIM or JMI feature selection criterion with this variable instead of the unobservable Y labels. Semi-IAMB uses Markov Blanket discovery which is a supervised learning algorithm used to find a Bayesian Network (A statistical model that may represent the independencies that exist in a domain graphically) that characterizes the Target node (Yaramakala and Margaritis, 2005).
Feature selection was performed on each ransomware/benign dataset as well as the overall dataset to find the best features to use in the classification stage. Sample code used to perform all three feature selection algorithms for the Charger/Benign1 dataset is shown in figure 3 of the configuration manual. The code provided in the configuration manual was used for each subset including the overall dataset to discover the selected features for each dataset. The code calls the three algorithms and the data passed through is X_data (all the data that makes up the dataset) and Y_labels(1 and 2). For example, semiJMI algorithm is called and it passes through the data, the labels and the probability p(y=1).

The results returned for each dataset included the returned subset for each semi supervised algorithm and the features returned if all Y labels are present. This project only focused on the returned features for the semi supervised algorithms. Below are the features that were selected for each dataset.

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    58    16    45    23    40    33    42    46     9    41

Returned subset using JMI with unobserved class labels Y:
    58    45    23    26    33     4    40    55    38    43

Returned subset using our Semi-MIM:
    58    45    16    33    42    23    55    26    43    46

Returned subset using MIM with unobserved class labels Y:
    58    45    23    26    55    33    42    43    25     4

Returned subset using our Semi-IAMB:
    45    58    33

Returned subset using IAMB with unobserved class labels Y:
    45    58    33
```

Charger/Benign1

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    59     1    23     3     4    41    26     9    25    21

Returned subset using JMI with unobserved class labels Y:
    59    42    33    46    43     9     1    44    38    40

Returned subset using our Semi-MIM:
    59     3     1    26    24    41    28    21    63    11

Returned subset using MIM with unobserved class labels Y:
    59    46    33    43    56    12     9    44    37     1

Returned subset using our Semi-IAMB:
    59

Returned subset using IAMB with unobserved class labels Y:
    59
```

Jiust/Benign2

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    59    40    12     3    63    46    23    64    15    60

Returned subset using JMI with unobserved class labels Y:
    59    12     3    46    57    37     9    56    26    23

Returned subset using our Semi-MIM:
    59     3    46    37    63    23    60    62    26    64

Returned subset using MIM with unobserved class labels Y:
    59     3    46    12    56    37     9    57    44    38

Returned subset using our Semi-IAMB:
    59

Returned subset using IAMB with unobserved class labels Y:
    59

>>
```

Koler – benign 3

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    58     3     2     1     4    24    21    23    46    19

Returned subset using JMI with unobserved class labels Y:
    58     3     1     2     4    21    46    23    24    26

Returned subset using our Semi-MIM:
    58     3     1     2     4    24    21    19    23    22

Returned subset using MIM with unobserved class labels Y:
    58     3     1     2     4    21    23    24    19    55

Returned subset using our Semi-IAMB:
    58     3

Returned subset using IAMB with unobserved class labels Y:
    58     3
```

Pletor - Benign 5

```
>> clear
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
     3    58    37    26    55    46     9    40     1    43

Returned subset using JMI with unobserved class labels Y:
     3    58    37    26     1    55     4    46     9    43

Returned subset using our Semi-MIM:
     3    37    58    55    43     9     1    26    40    23

Returned subset using MIM with unobserved class labels Y:
     3    37    58    55    43     1     9    26    12    23

Returned subset using our Semi-IAMB:
     3    58

Returned subset using IAMB with unobserved class labels Y:
     3    58
```

Ransombo benign 7

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    19    66    23    24    39    21    46     4     9    16

Returned subset using JMI with unobserved class labels Y:
    24     4    45    21    19    28    26     9    12    22

Returned subset using our Semi-MIM:
    19    22    27    24    26     4    21    23     6     8

Returned subset using MIM with unobserved class labels Y:
    24     4    21    19    26    22    27    23    12    45

Returned subset using our Semi-IAMB:
    45     4

Returned subset using IAMB with unobserved class labels Y:
    45     4
```

svpeng – benign 9

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    58     9    16    12    41     4     1    23    38    45

Returned subset using JMI with unobserved class labels Y:
    58    45     1    46    55    43     4     9    12    41

Returned subset using our Semi-MIM:
    58    45     9     1    12    40    38    61    55    60

Returned subset using MIM with unobserved class labels Y:
    58    45    55    43    12     9    40     1    47    39

Returned subset using our Semi-IAMB:
    58    43

Returned subset using IAMB with unobserved class labels Y:
    58    43
```

Lockerpin benign 4

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
     3    12     9    40    37     4    44    15    14    56

Returned subset using JMI with unobserved class labels Y:
     3     9    37     4    14    56    12    44    45    40

Returned subset using our Semi-MIM:
     3    37     9    12    14    44    56    38    40    15

Returned subset using MIM with unobserved class labels Y:
     3    37    14    56     9    44    12     1    40    38

Returned subset using our Semi-IAMB:
     3    44     4

Returned subset using IAMB with unobserved class labels Y:
     3    44     4
```

Porndroid benign 6

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    58     1    47    21    39     4    28    23    13    20

Returned subset using JMI with unobserved class labels Y:
    58     1    21    26    33    31    42    13    23    28

Returned subset using our Semi-MIM:
    58     1     9    38    40    26    23    11    48     2

Returned subset using MIM with unobserved class labels Y:
    58     1    33    42    26    23    31    17    28    21

Returned subset using our Semi-IAMB:
    58     1    44

Returned subset using IAMB with unobserved class labels Y:
    58     1    44

>>
```

simplocker - benign8

```
>> clear
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
     3     1    21     4    16    39    56    26    19    25

Returned subset using JMI with unobserved class labels Y:
     3    22    59     4    19    18     1    21    23    48

Returned subset using our Semi-MIM:
     3     1    21     4    19    26    59    24    25    23

Returned subset using MIM with unobserved class labels Y:
     3     4    59    19    21    23    26    22    24    27

Returned subset using our Semi-IAMB:
     3

Returned subset using IAMB with unobserved class labels Y:
     3
```

wanalocker - benign 10

```
>> Tutorial_SemiSupervised_FS
Returned subset using our Semi-JMI:
    46    25    57    20    26     2    21    39    23    16

Returned subset using JMI with unobserved class labels Y:
    46    26    23    21    66    56     4    45    25    20

Returned subset using our Semi-MIM:
    46    25    45    57     6     8    54    55    20    39

Returned subset using MIM with unobserved class labels Y:
    46    26    21    45    23     4    56    25    20    24

Returned subset using our Semi-IAMB:
    46    25

Returned subset using IAMB with unobserved class labels Y:
    46    25
```

Ransomware Benign All

Each Feature that was chosen was created the datasets used with the classification stage. All labels were added back to the data and the datasets were converted to .csv files.

5.3 Normalization stage

Once the features where selected and datasets created with Y labels included, analysis was done to identify all the continuous data. The continues data was normalized between zero and one using the Min -Max method. This was done because the data provided in the CICANDMAL 2017 dataset contained different attribute values and in very large intervals. As stated previously Python code and Google collab was used to perform the Normalization stage of this project. Sample code for this stage can be seen in figure 4 of the configuration Manual
Once the continuous features were normalized they were then added back into the dataset to be used in the classification stage.

5.4 Classification Stage

As stated previously, this stage was performed using the WEKA software tool. For automatic classification, regression, clustering, and feature selection—common data mining issues in bioinformatics research—the Weka machine learning workbench offers a multipurpose platform. It includes a wide range of machine learning algorithms and data pre-processing techniques, as well as graphical user interfaces for exploring data and comparing several machine learning approaches on the same problem experimentally (Frank *et al.*, 2004).

To get the semi supervised algorithms required for this project, the collection- classification package was installed within the weka platform. [1] is a package for algorithms around semi-supervised learning and collective classification. When this package is run a collective folder is added containing all the Semi Supervised learning algorithms needed. As stated in the classification section of the design specification the three algorithms used for classification were YATSI(RF), Collective IBK and Collective Wrapper(RF). All three methods are semi supervised approaches to machine learning. Random forest was used within YATSI and WRAPPER methods as the base classifier as it out performs other forms of classification (Noorbehbahani and Saberi, 2020).
All the datasets were added to Weka and used within each algorithm to calculate the accuracy. An example of the results obtained when Semi MIM feature selection - Charger/Benign1 dataset was used on YATSI RF Classification can be seen in figure 6 of the configuration manual .
All the results were then added to Excel and graphed to perform the evaluation stage of the project.

[1] https://github.com/fracpete/collective-classification-weka-package

6 Evaluation

In this section the accuracy of the classification models were evaluated for each Feature selection method. The Accuracy of results were added to excel and graphed for further analysis and comparison. Accuracy can be defined as the number of times the model correctly classified all Benign traffic and all Ransomware traffic. Also, speed was evaluated for the overall dataset. Again this was added to excel and graphed.
Utilizing k-fold cross-validation, the accuracy of the models is determined. A well-known evaluation statistic in machine learning is the accuracy measure. Although, it is not a proper measure when the dataset is imbalanced (Noorbehbahani and Saberi, 2020). The accuracy serves as a good measure for evaluation because all subsets in our experiment are balanced.
As the overall dataset is unbalanced a percentage split was used that can be generated using the Weka platform from the dataset provided.

6.1 Experiment for Semi JMI with each dataset and each classification model

Figure 7. Semi JMI – Classification Model

6.2 Experiment for Semi MIM with each dataset and each classification model

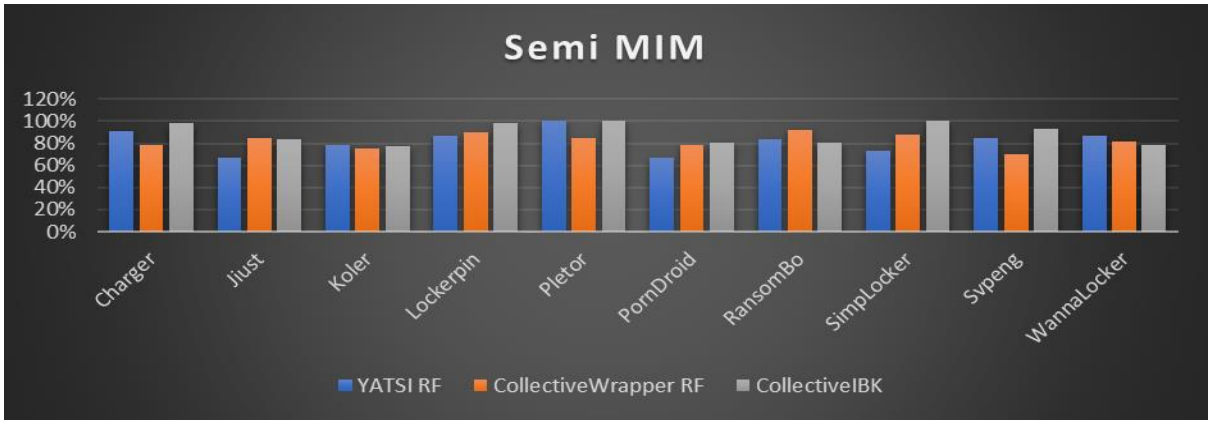

Figure 8. Semi MIM – Classification Model

6.3 Experiment for Semi IAMB with each dataset and each classification model

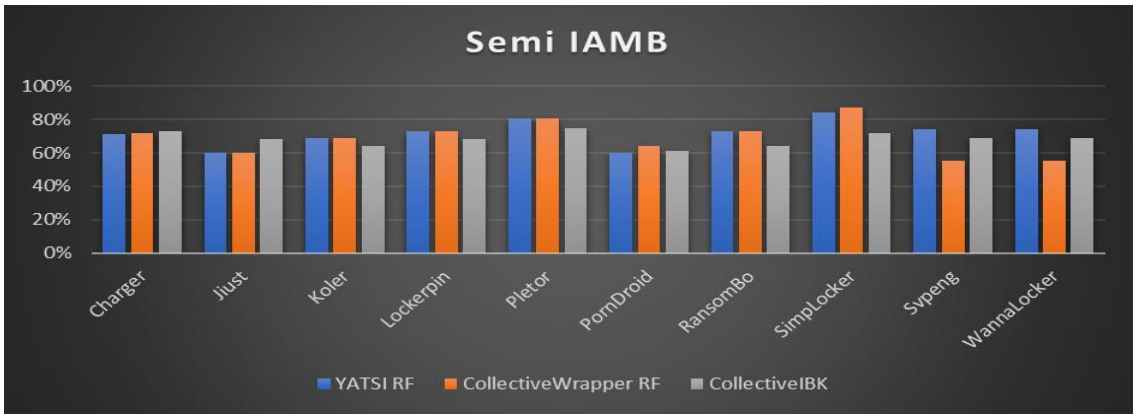

Figure 9. Semi IAMB – Classification Model

6.4 Experiment for the full dataset with each Semi Supervised feature selection and classification included

The k-fold cross-validation method is efficient and beneficial for assessing the effectiveness of the classifiers. This is because it lowers evaluation bias by validating the data numerous times (Noorbehbahani and Saberi, 2020).
We can see in Figure 7 and 8 that the classification models performed really well when used with Semi JMI and Semi MIM feature election methods. Although Semi JMI had more accurate results for some of the subsets. The classification models combined with Semi IAMB showed weaker results when compared to the other two.

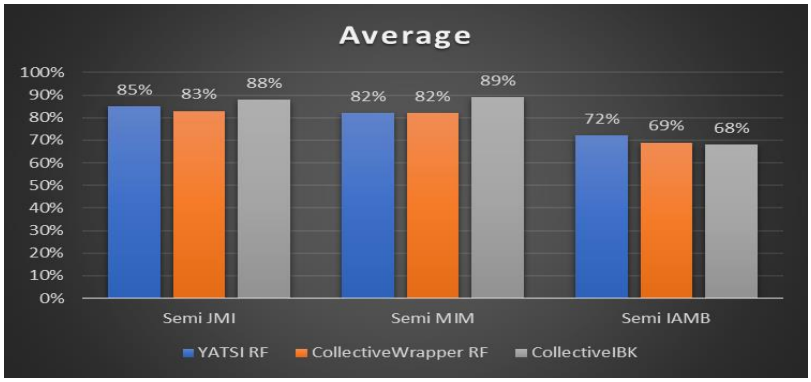

Figure 11. Average of all results on subsets

As seen in Figure two the best accuracy average is when collective IBK combined with Semi MIM at 89%. Although it can been seen that Semi JMI has the better overall average with the highest average accuracy being 88% when combined with Collective IBK. The classification models perform poorly when used with Semi IAMB as the highest average being 72% when used with YATSI RF.
As stated previously K-fold was not used on the combined dataset is unbalanced. This was chosen as the dataset was meant to mirror real world traffic which will never be balanced. A percentage split was chosen to calculate the accuracy instead.

Method	YATSI RF	CollectiveWrapper RF	CollectiveIBK
Semi JMI	78%	64%	78%
SEMI MIM	64%	66%	62%
SEMI IAMB	57%	62%	57%

Figure 12. Results of combined Overall dataset

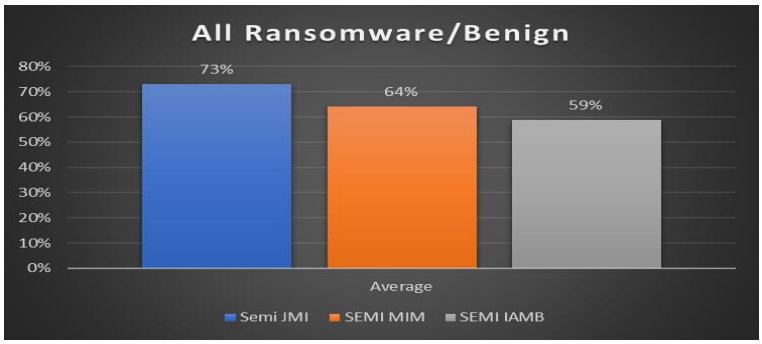

Figure 13. Average of all results for combined Overall dataset

As shown in figure 13 Semi JMI showed the best average when combined with the classification models at 73%. Semi IAMB showed the weakest resusts having an average of 59%. The approach within this project shows impressive results for accuarcy and would definitely have benefits in the real world when it comes to ransomware detection.
Although this project lacked big datasets. The Weka platform could not perfrom the classification stage on the datasets that this project aimed to use. Datasets were shortened so they could be implemented. Also, the use of a percentrage split within Weka means you will test your knowledge on the same data you learned giving the results an bit of an advantage.

(Noorbehbahani and Saberi, 2020) proposes a semi supervised learning approach with supervised feature selection
methodologies for ransomware detection. The results are calculated using crossvalidation of 5 folds although, this project used 10 folds for its calculation. (Noorbehbahani and Saberi, 2020) shows impresive results using Wrapper RF classification model with OneR or Chi-squared feature selection methodoligies. While the results are impresive, the results are lower scores than the scores achieved by the model conducted in this research project.

Selection	YATSI RF	CollectiveWrapper RF	CollectiveIBK
Semi JMI	52.42	26.71	102.3
Semi MIM	15.85	12.46	51.07
Semi IAMB	8.37	4.44	30.26

Figure 14. Time in seconds on Overall Dataset

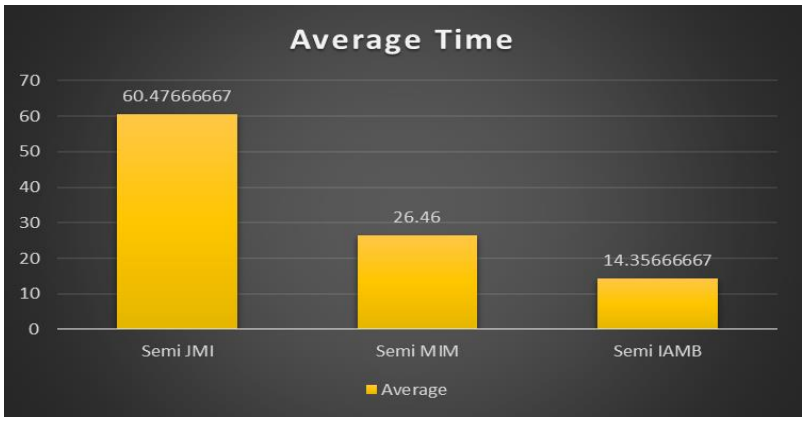

Figure 15. Average Time in seconds on Overall Dataset

Figure 14 shows the time it took for the classification to be implemented when paired with the feature selection algorithms. Although Semi JMI performed the best with accuracy, it was the slowest to perform classification. Semi IAMB was the quickest but accuracy wise the algorithm was weak. Semi MIM was a lot quicker than Semi JMI and was impressive with accuracy. Although Semi JMI was slower its average accuracy was still 12% higher then Semi JMI confirming that it is the better algorithm to use.

7 Conclusion and Future Work

7.1 Conclusion

In this paper the impact of Semi Supervised feature selection in combination with classification models for Ransomware detection was assessed and the accuracy measured. Three Semi supervised feature selection methods (Semi JMI, Semi MIM, Semi IAMB) were implemented to get the subset section features for model construction with Semi Supervised classification methods. The feature selection approaches were performed on 10 ransomware/benign datasets and one dataset of all the data combined. All data was gathered using the CICANDMAL 2017 dataset.

The Semi Supervised feature selection models were then combined with the Semi Supervised classification models and the accuracy was evaluated. These classification models were evaluated using K-fold validation and test sets. This was performed using the Weka application and Excel. The results determined that Semi-MIM combined with collective IBK resulted in the best overall average when tested on the subsets although Semi-JMI displayed a better overall average for

all three classification methods at 85%. Semi IAMB showed the weakest average compared to the other two.
For the combined dataset that was unbalanced mirroring the real world, Semi-MIM proved to be the most accurate when combined with the classification models having an average of 91% accuracy for determining the difference between ransomware and benign data.

7.2 Limitations

Due to the limitations on the amount of data Weka can process as even increasing the heap only improved the amount the was able to be classified so much, it was not possible to run the classification on the dataset size this project was aimed at to mirror the amount of traffic a network might see in the real world.

7.3 Future work

For the future work of this research, live traffic passing through a security gateway or antivirus could be used and evaluated as apposed to using a dataset like CICANDMAL 2017. The model would need to be further developed in order to be integrated into a live Security Gateway after being successfully applied to real-world traffic and should the results maintain a high level of accuracy.d

8 References

Adamov, A. and Carlsson, A. (2020) 'Reinforcement Learning for Anti-Ransomware Testing', in *2020 IEEE East-West Design & Test Symposium (EWDTS)*. Varna, Bulgaria: IEEE, pp. 1–5. Available at: https://doi.org/10.1109/EWDTS50664.2020.9225141.

Alhawi, O.M.K., Baldwin, J. and Dehghantanha, A. (2018) 'Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection', in A. Dehghantanha, M. Conti, and T. Dargahi (eds) *Cyber Threat Intelligence*. Cham: Springer International Publishing (Advances in Information Security), pp. 93–106. Available at: https://doi.org/10.1007/978-3-319-73951-9_5.

Azevedo, A. and Santos, M.F. (2008) 'KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW'. P.Porto. Available at: https://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf.

Beaman, C. *et al.* (2021) 'Ransomware: Recent advances, analysis, challenges and future research directions', *Computers & Security*, 111, p. 102490. Available at: https://doi.org/10.1016/j.cose.2021.102490.

Bennasar, M., Hicks, Y. and Setchi, R. (2015) 'Feature selection using Joint Mutual Information Maximisation', *Expert Systems with Applications*, 42(22), pp. 8520–8532. Available at: https://doi.org/10.1016/j.eswa.2015.07.007.

Berrueta, E. *et al.* (2019) 'A Survey on Detection Techniques for Cryptographic Ransomware', *IEEE Access*, 7, pp. 144925–144944. Available at: https://doi.org/10.1109/ACCESS.2019.2945839.

Bijitha, C.V., Sukumaran, R. and Nath, H.V. (2020) 'A Survey on Ransomware Detection Techniques', in S.K. Sahay et al. (eds) *Secure Knowledge Management In Artificial Intelligence Era*. Singapore: Springer Singapore (Communications in Computer and Information Science), pp. 55–68. Available at: https://doi.org/10.1007/978-981-15-3817-9_4.

Board, H. (2021) 'HSE publishes independent report on Conti cyber attack', *HSE*. Available at: https://www.hse.ie/eng/services/publications/conti-cyber-attack-on-the-hse-full-report.pdf.

Comar, P.M. *et al.* (2013) 'Combining supervised and unsupervised learning for zero-day malware detection', in *2013 Proceedings IEEE INFOCOM*. Turin, Italy: IEEE, pp. 2022–2030. Available at: https://doi.org/10.1109/INFCOM.2013.6567003.

Debuse, J.C.W. *et al.* (2001) 'Building the KDD Roadmap', in R. Roy (ed.) *Industrial Knowledge Management*. London: Springer London, pp. 179–196. Available at: https://doi.org/10.1007/978-1-4471-0351-6_12.

van Engelen, J.E. and Hoos, H.H. (2020) 'A survey on semi-supervised learning', *Machine Learning*, 109(2), pp. 373–440. Available at: https://doi.org/10.1007/s10994-019-05855-6.

Frank, E. *et al.* (2004) 'Data mining in bioinformatics using Weka', *Bioinformatics*, 20(15), pp. 2479–2481. Available at: https://doi.org/10.1093/bioinformatics/bth261.

HSU, K., Levine, S. and Finn, C. (2019) 'UNSUPERVISED LEARNING VIA META-LEARNING'. Available at: https://arxiv.org/pdf/1810.02334.pdf.

Kotsiantis, S.B., Kanellopoulos, D. and Pintelas, P.E. (2006) 'Data Preprocessing for Supervised Leaning'.

Lashkari, A.H. *et al.* (2018) 'Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification', in *2018 International Carnahan Conference on Security Technology (ICCST)*. Montreal, QC: IEEE, pp. 1–7. Available at: https://doi.org/10.1109/CCST.2018.8585560.

Nieuwenhuizen, D. (2017) 'A behavioural-based approach to ransomware detection'. MWR Labs Whitepaper. Available at: https://labs.withsecure.com/assets/resourceFiles/mwri-behavioural-ransomware-detection-2017-04-5.pdf.

Noorbehbahani, F., Rasouli, F. and Saberi, M. (2019) 'Analysis of Machine Learning Techniques for Ransomware Detection', in *2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*. Mashhad, Iran: IEEE, pp. 128–133. Available at: https://doi.org/10.1109/ISCISC48546.2019.8985139.

Noorbehbahani, F. and Saberi, M. (2020) 'Ransomware Detection with Semi-Supervised Learning', in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*. Mashhad, Iran: IEEE, pp. 024–029. Available at: https://doi.org/10.1109/ICCKE50421.2020.9303689.

Sechidis, K. and Brown, G. (2018) 'Simple strategies for semi-supervised feature selection', *Machine Learning*, 107(2), pp. 357–395. Available at: https://doi.org/10.1007/s10994-017-5648-2.

Sgandurra, D. *et al.* (2016) 'Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection'. Available at: https://arxiv.org/pdf/1609.03020.pdf.

Sheikhpour, R. *et al.* (2017) 'A Survey on semi-supervised feature selection methods', *Pattern Recognition*, 64, pp. 141–158. Available at: https://doi.org/10.1016/j.patcog.2016.11.003.

Shijo, P.V. and Salim, A. (2015) 'Integrated Static and Dynamic Analysis for Malware Detection', *Procedia Computer Science*, 46, pp. 804–811. Available at: https://doi.org/10.1016/j.procs.2015.02.149.

Yaramakala, S. and Margaritis, D. (2005) 'Speculative Markov Blanket Discovery for Optimal Feature Selection', in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. Houston, TX, USA: IEEE, pp. 809–812. Available at: https://doi.org/10.1109/ICDM.2005.134.