Configuration Manual

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		
Programme:	Masters in Cyber Security	**Year:**	2022
Module:	Research Project/Internship		
Lecturer:	Vikas Sahni		
Submission Due Date:	15th August 2022		
Project Title:	Can Semi Supervised feature selection improve ransomware detection		
Word Count: 1479	**Page Count:** 10		

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):	

Configuration Manual

Gavin Smyth
Student ID: X16354406

1 Introduction

This document's goal is to describe the implementation process that was used for this research project. It also covers the software requirements that were needed for implementation. Additionally, this configuration manual also contains snippets of the code that was used during the development of the research project.

2 Project implementation

2.1 Data Selection

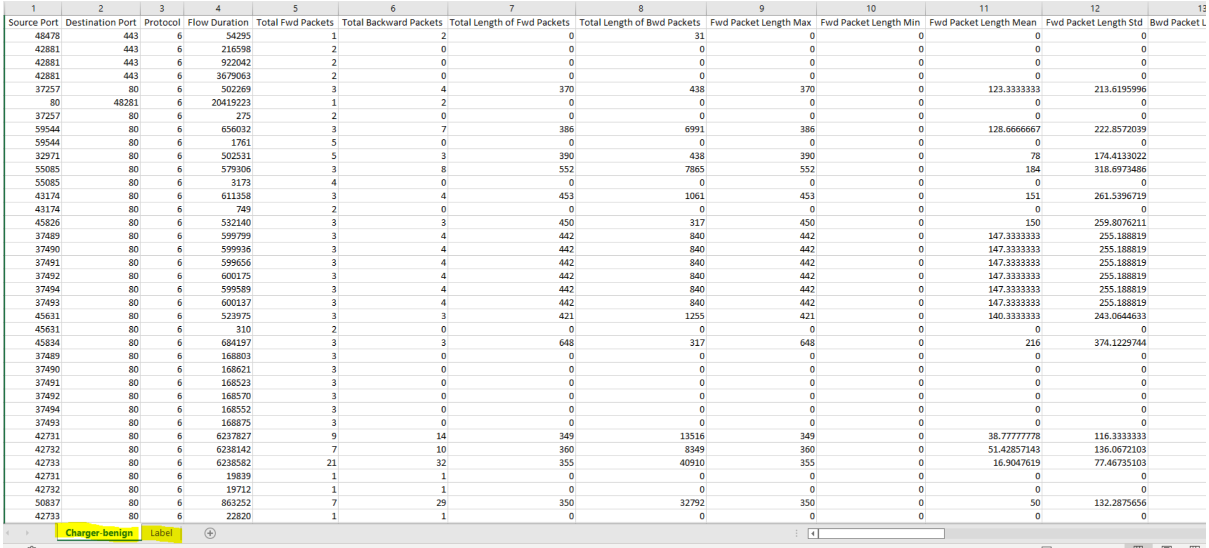
This stage determined the target data and variables used. The dataset chosen for this research project was the CIC-AndMal2017 dataset. CIC-AndMal2017 is an android malware dataset which contains both malware and benign applications which can be used for security testing and malware prevention (Lashkari *et al.*, 2018) . Ransomware was the malware used for this research project.

2.2 Data preperation

Data Preparation was the first stage of the implementation that was undertaken for this research project. The CIC-AndMal2017 dataset was split into subsets based on each ransomware family. Each ransomware family was combined with benign data to create 10 different ransomware/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. This included:

- 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.
- All/Benign11

Also within this step irrelevant data was removed (flow ID, Source IP, Destination IP, Time Stamp and fwd header length).
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).



1	2	3	4	5	6	7	8	9	10	11	12	13
Source Port	Destination Port	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet L
48478	443	6	54295	1	2	0	31	0	0	0	0	
42881	443	6	216598	2	0	0	0	0	0	0	0	
42881	443	6	922042	2	0	0	0	0	0	0	0	
42881	443	6	3679063	2	0	0	0	0	0	0	0	
37257	80	6	502269	3	4	370	438	370	0	123.3333333	213.6195996	
80	48281	6	20419223	1	2	0	0	0	0	0	0	
37257	80	6	275	2	0	0	0	0	0	0	0	
59544	80	6	656032	3	7	386	6991	386	0	128.6666667	222.8572039	
59544	80	6	1761	5	0	0	0	0	0	0	0	
32971	80	6	502531	5	3	390	438	390	0	78	174.4133022	
55085	80	6	579306	3	8	552	7865	552	0	184	318.6973486	
55085	80	6	3173	4	0	0	0	0	0	0	0	
43174	80	6	611358	3	4	453	1061	453	0	151	261.5396719	
43174	80	6	749	2	0	0	0	0	0	0	0	
45826	80	6	532140	3	3	450	317	450	0	150	259.8076211	
37489	80	6	599799	3	4	442	840	442	0	147.3333333	255.188819	
37490	80	6	599936	3	4	442	840	442	0	147.3333333	255.188819	
37491	80	6	599656	3	4	442	840	442	0	147.3333333	255.188819	
37492	80	6	600175	3	4	442	840	442	0	147.3333333	255.188819	
37494	80	6	599589	3	4	442	840	442	0	147.3333333	255.188819	
37493	80	6	600137	3	4	442	840	442	0	147.3333333	255.188819	
45631	80	6	523975	3	3	421	1255	421	0	140.3333333	243.0644633	
45631	80	6	310	2	0	0	0	0	0	0	0	
45834	80	6	684197	3	3	648	317	648	0	216	374.1229744	
37489	80	6	168803	3	0	0	0	0	0	0	0	
37490	80	6	168621	3	0	0	0	0	0	0	0	
37491	80	6	168523	3	0	0	0	0	0	0	0	
37492	80	6	168570	3	0	0	0	0	0	0	0	
37494	80	6	168552	3	0	0	0	0	0	0	0	
37493	80	6	168875	3	0	0	0	0	0	0	0	
42731	80	6	6237827	9	14	349	13516	349	0	38.77777778	116.3333333	
42732	80	6	6238142	7	10	360	8349	360	0	51.42857143	136.0672103	
42733	80	6	6238582	21	32	355	40910	355	0	16.9047619	77.46735103	
42731	80	6	19839	1	1	0	0	0	0	0	0	
42732	80	6	19712	1	1	0	0	0	0	0	0	
50837	80	6	863252	7	29	350	32792	350	0	50	132.2875656	
42733	80	6	22820	1	1	0	0	0	0	0	0	

Figure 1. Excel spreadheet before its converted to .mat file

Figure 1 shows an example of the Charger/Benign1 dataset before its converted to a .mat file so that feature selection can be performed. The Charger-benign tab contains all the data within the dataset and the label tab states if it is ransomware or benign data.

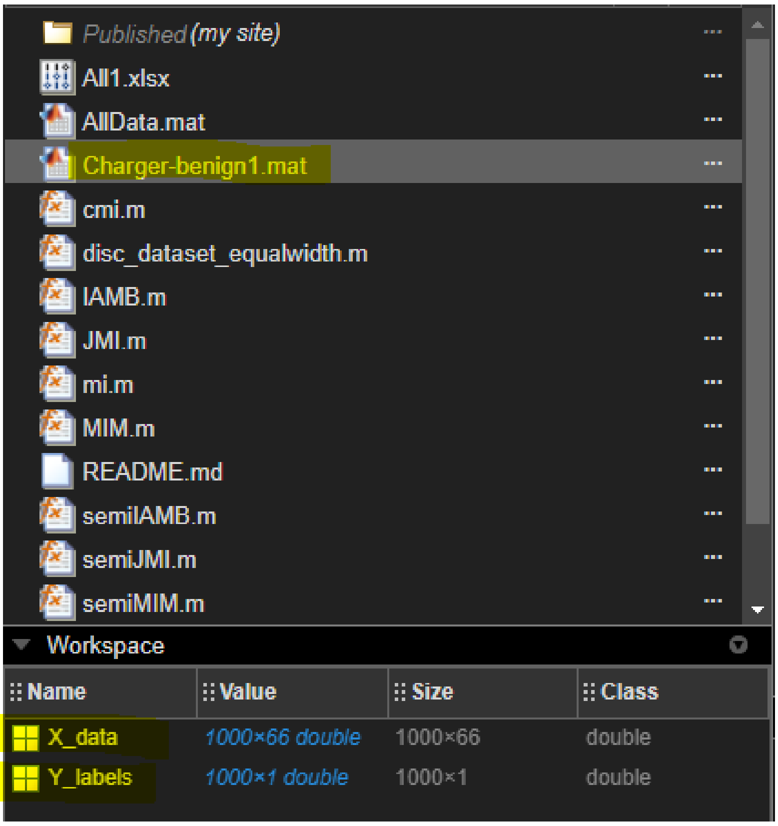


Figure 2. Charger/Benign1 dataset coverted to .mat file

Figure 2 shows the charger benign dataset after its has been converted to a .mat file. Shown are two variable within the file(X_data and Y_Labels).

2.3 Feature Selection

The code used to perform the feature selection was taken from (Sechidis and Brown, 2018) and the code can be found in the footnote below[1]. The Feature selection stage was conducted using the MATLAB platform. Each dataset was added to MATLAB as previously stated 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

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.

```
% Load one of the provided dataset, e.g.
load('Charger-benign1.mat')
% Transform multi-class problems to binary in 1-vs-all strategy
Y_labels(Y_labels~=1) =0;
% Discretise continuous features via an equal-width-strategy, for example using 5 bins
X_data = disc_dataset_equalwidth(X_data,5);

%%% Generate semi-supervised datasets %%%
pS1 = 0.25; % probability p(s=1)
c = 1.25; % For c=1. we are in the MCAR scenario, while for c>1 in MAR-C
pY1 = mean(Y_labels); % probability p(y=1)
pS1givenY1 = c*pS1; % probability p(s=1|y=1)
pS0givenY0 = (1-c*pY1)*pS1/ (1-pY1); % probability p(s=1|y=0)

% Label some positive examples
ypos_indices = find(Y_labels==1);%Find the positive indeces
positiveSet = binornd(1,pS1givenY1,1,length(ypos_indices))';% Find which examples will be labelled
while sum(positiveSet~=0) == 0 && pS1givenY1>0 % Check if you have empty labelled set, if yes re-sample
    positiveSet = binornd(1,pS1givenY1,1,length(ypos_indices))';
end;
S1Y1_indices = ypos_indices(find(positiveSet==1));  %update

% Label some negative examples
yneg_indices = find(Y_labels==0);%Find the positive indeces
negativeSet = binornd(1,pS0givenY0,1,length(yneg_indices))';% Find which examples will be labelled
while sum(negativeSet~=0) == 0 && pS0givenY0>0 % Check if you have empty labelled set, if yes re-sample
    negativeSet = binornd(1,pS0givenY0,1,length(yneg_indices))';
end;
S1Y0_indices = yneg_indices(find(negativeSet==1));  %update

% So the proxy variable that we observe in the semi-supervised scenario is:
Y_proxy = NaN(size(X_data,1),1); % Initialize with all values missing, NaN
Y_proxy(S1Y1_indices) = 1; % The positvely-labelled examples
Y_proxy(S1Y0_indices) = 0; % The negatively-labelled examples

%%% Select the features using our suggested algorithms %%%
% Now we will select the top-10 features with Semi-JMI and we will compare
% these subsets with the idea, which is the one that we have if we use JMI
% in the unobserved class labels Y
topK=10;
Selected_with_semiJMI = semiJMI(X_data, Y_proxy, topK, pY1);
disp('Returned subset using our Semi-JMI:')
disp(Selected_with_semiJMI)
Selected_with_ideal_JMI = JMI(X_data, Y_labels, topK);
disp('Returned subset using JMI with unobserved class labels Y:')
disp(Selected_with_ideal_JMI)
% We repeat the same for MIM
Selected_with_semiMIM = semiMIM(X_data, Y_proxy, topK, pY1);
disp('Returned subset using our Semi-MIM:')
disp(Selected_with_semiMIM)
Selected_with_ideal_MIM = MIM(X_data, Y_labels, topK);
disp('Returned subset using MIM with unobserved class labels Y:')
disp(Selected_with_ideal_MIM)
% And the same with IAMB
alpha = 0.05;
Selected_with_semiIAMB = semiIAMB(X_data, Y_proxy, alpha, pY1);
disp('Returned subset using our Semi-IAMB:')
disp(Selected_with_semiIAMB)
Selected_with_ideal_IAMB = IAMB(X_data, Y_labels, alpha);
disp('Returned subset using IAMB with unobserved class labels Y:')
disp(Selected_with_semiIAMB)
```

Figure 3. Feture selection performed on Charger/Benign1 dataset

[1] https://github.com/sechidis/2018-MLJ-Semi-supervised-feature-selection (Sechidis and Brown, 2018)

The code provided in figure 3 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 a portion of 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).

```
function [selectedFeatures] = semiJMI(X_data, Y_proxy, topK, prior_y)
% Summary
%    Semi-JMI algorithm for feature selection (Sechidis Brown, 2017)
% Inputs
%    X_data: n x d matrix X, with categorical values for n examples and d features
%    Y_proxy: n x 1 vector \tilde{Y}, with values 1 and 0 for the positevely
%             labelled and negatively labelled examples respectively (labelled data),
%             and NaN for the examples with missing label (unlabelled data)
%    topK: Number of features to be selected
%    prior_y: Our bilief over the marginal probabilty p(y=1), default value
%             is the probability in the labelled set p(Y_proxy=1)


if nargin<3
    error('Not enough input arguments');
else if nargin<4
        prior_y = nanmean(Y_proxy==1);
    end
end

% Step 1: Initialise
n = sum(Y_proxy==0); % number of negatives supplied with labels
p = sum(Y_proxy==1); % number of positives supplied with labels
m = sum(isnan(Y_proxy)); % number of missing labels

% Step 2: Create surrogate variables
Y_proxy_0 = Y_proxy; Y_proxy_0(isnan(Y_proxy)) = 0;
Y_proxy_1 = Y_proxy; Y_proxy_1(isnan(Y_proxy)) = 1;

% Step 3: Calculate switching threshold
a = sqrt(p*(p+m));
b = sqrt(n*(n+m));
phi = a/(a+b);

% Step 4: Decide optimal surrogate (Theorem 8) and use it in IAMB to derive MB
if prior_y < phi
    Y_labels = Y_proxy_0;
else
    Y_labels = Y_proxy_1;
end
selectedFeatures = JMI(X_data,Y_labels, topK);
```

Figure 4. Semi JMI Feature Selection Code Snippet

Figure 4 shows an example of one of the Semi supervised feature selection methods (Semi JMI)

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 results obtained for the feature selection:

```
>> 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
```

```
>> 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
```

Charger/Benign1

```
>> 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

>>
```

Jiust/Benign2

```
>> clear
>> 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
```

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
```

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
```

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
```

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

>>
```

Ransombo Benign 7

```
>> clear
>> 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
```

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
```

svpeng – Benign 9

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

2.4 Normalization

After the feature selection stage, datasets were put back together within excel containing the new variables for each dataset. Once the features where selected and the new datasets were created with Y labels included, analysis was done to identify all the continuous data. The continuous data was then separated from the dataset into their own dataset and added to one drive. The continues data was normalized between zero and one using the Min -Max method. Google collab was used to perform the Normalization stage of this project.

```
import pandas as pd
import numpy as np
ransomeware = pd.read_csv("/content/drive/MyDrive/DataSets/All-semi/wanalocker - benign 10 - semi mim normalized.csv")

from sklearn import preprocessing

x = ransomeware.values

min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x)

df = pd.DataFrame(x_scaled)

ransomeware.head()
```

	Flow Duration	Flow IAT Mean	Flow IAT Max	Fwd IAT Total	Fwd IAT Mean	Fwd IAT Std	Fwd IAT Max
0	38	38.000	38.0	38.0	38.0	0.000000e+00	38.0
1	678	339.000	415.0	678.0	339.0	1.074802e+02	415.0
2	226491	18874.250	143367.0	60582.0	12116.4	1.658777e+04	35140.0
3	26421468	3774495.429	26000000.0	26400000.0	8807156.0	1.510000e+07	26200000.0
4	37272	37272.000	37272.0	0.0	0.0	0.000000e+00	0.0

```
df.to_csv('/content/drive/MyDrive/DataSets/All-semi/wanalocker - benign 10 - semi mim normalized1.csv')
```

Figure 5. Normalization Code

Once the continuous data was normalized, it was then added back into the correct dataset.

2.5 Classification Stage

The Classification stage was performed using the WEKA software tool. To get the semi supervised algorithms required for this project, the collection- classification package was installed within the weka platform. [2] 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. All the datasets were added to Weka and used within each algorithm to calculate the accuracy and all the overall dataset, time was analysed.

```
=== Summary ===

Correctly Classified Instances          91               91      %
Incorrectly Classified Instances         9                9      %
Kappa statistic                          0.8085
Mean absolute error                      0.141
Root mean squared error                  0.2578
Relative absolute error                 28.2044 %
Root relative squared error             51.5617 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.825    0.033    0.943      0.825    0.880      0.813    0.985     0.976     1
                 0.967    0.175    0.892      0.967    0.928      0.813    0.985     0.991     2
Weighted Avg.    0.910    0.118    0.913      0.910    0.909      0.813    0.985     0.985

=== Confusion Matrix ===

  a  b   <-- classified as
 33  7 |  a = 1
  2 58 |  b = 2
```

Figure 6. Results of Semi MIM feature selection - Charger/Benign1 dataset was used with the YATSI RF Classification

Figure 6 shows an example of the results obtained when Semi MIM feature selection - Charger/Benign1 dataset was used with the YATSI RF Classification. The Correctly classified instance percentage allowed me to get the accuracy for each algorithm.

2.6 Evaluation

In this section the accuracy of the classification models were evaluated for each Feature selection method as well as the time for the overall dataset. For the subsets as they were balanced, the results were calculated using k-fold cross-validation. As the overall dataset is unbalanced a percentage split was used that can be generated using the Weka platform from the dataset provided.
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.

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

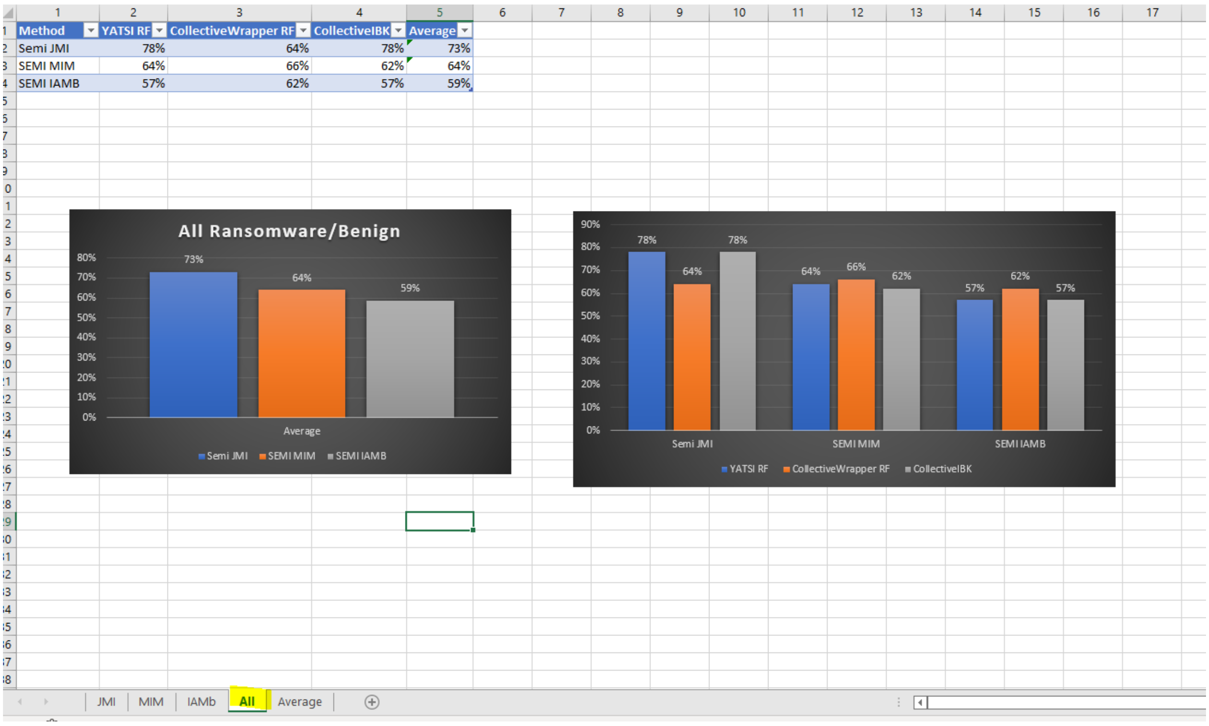


Figure 7. Accuracy of Overall dataset evaluated using Excel

Figure 7 shows an example of how accuracy was evaluated for the overall dataset. Each Semi supervised feature selection method was compared to how well they performed with the three different classification models

3 References

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.

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.