Configuration Manual

MSc Internship
Cyber Security

Kaleemuddin Mohammed
Student ID: x18132855

School of Computing
National College of Ireland

Supervisor: Ben Fletcher

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Kaleemuddin Mohammed		
Student ID:	X18132855		
Programme:	Cyber Security	**Year:**	2019
Module:	MSc Internship		
Lecturer:	Ben Fletcher		
Submission Due Date:	12/12/2019		
Project Title:	Prevention and Propagation of Malware by Using Hybrid Adaptive Neuro-Fuzzy Interface System		
Word Count:	1077	**Page Count:**	6

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.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ……………………………………………………………………………………………………

Date: ……………………………………………………………………………………………………

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

Kaleemuddin Mohammed
Student ID: x18132855

1 Introduction

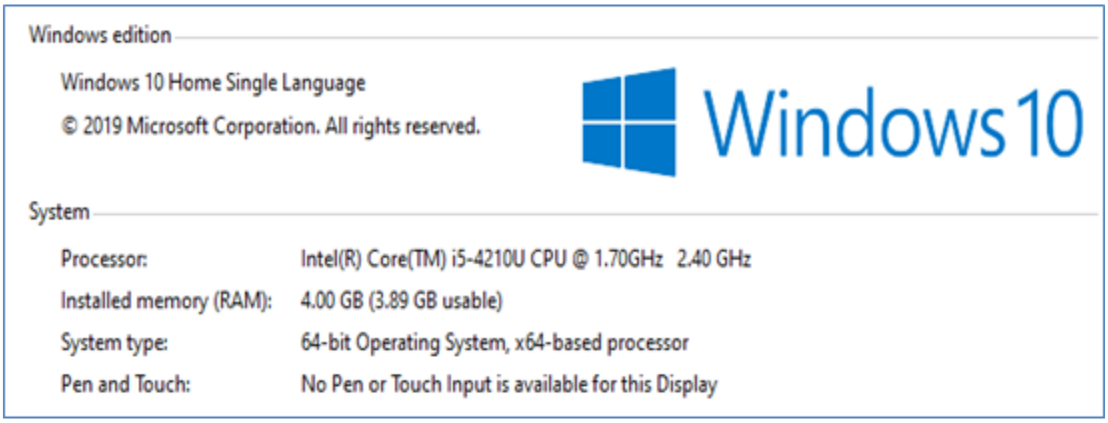
This configuration manual provides the details of the proposed work and model used for detecting malware. It makes use of the proposed algorithm named Hybrid Adaptive Neuro-Fuzzy Inference System. It takes malware dataset as input and produces a fuzzy inference system that can detect malware. This document provides details of configurations required by the project. Some literature on malware detection methods can be found in [1], [2], [3], [4] and [5]. However, in this project, the implementation of the system is based on the algorithm proposed, and the configuration details are provided in this document to help in understanding, setting environment and execution of the project.

2 System Configuration

This section provides an overview of the system used for the implementation of this model.

2.1 Hardware Specification

This project is developed using a laptop running Windows 10 operating system. The system specifications are as shown in Figure 1.



Windows edition

Windows 10 Home Single Language

© 2019 Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz
Installed memory (RAM):	4.00 GB (3.89 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Figure 1: Hardware specification of a system with Windows 10 OS

3 Software Specification

This section describes the details of the tools and technologies used while developing the project.

Tool	Version	Description
Java Development Kit	1.8	It is the Java language support that is essential to develop applications using the Java platform.
NetBeans	8.2	It is the integrated development environment (IDE) used for application development. It provides GUI with drag and drops features to build Java-based GUI applications.

Table 1: Tools used in this model

4 Working

This section illustrates step by step procedure used for setting up the proposed model and demonstrates its operation.

4.1 Software Installation

JDK 1.8 and NetBeans 8.2 are installed. They are downloaded and installed. The links are as follows. Both are open source and freely available.

4.2 Implementation

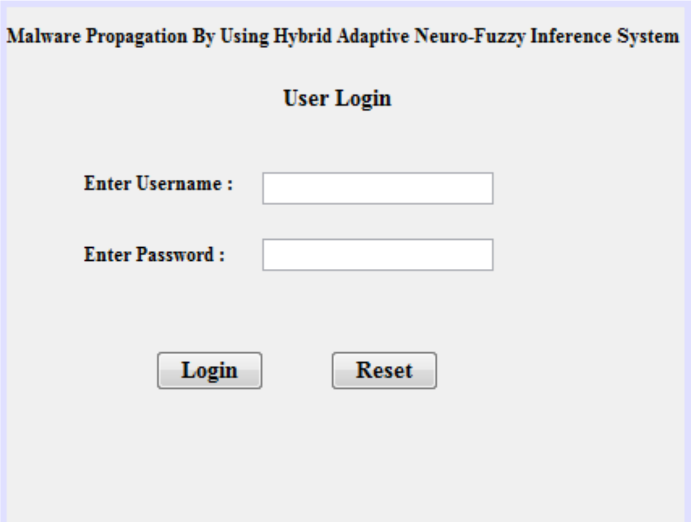
After installing JDK 1.8 and NetBeans 8.2, the NetBeans IDE is used to build the application. For this, open NetBeans, File → New Project → Java Project → Provide project name → Finish.

Then the project has been built with GUI that has features to load the dataset and then apply the proposed algorithm to detect malware.

To run the project:

1. Open NetBeans IDE
2. File → Open → Project Name (Malware Detection)

It shows the authentication screen, with username : admin and password : admin.as shown in Figure 2.

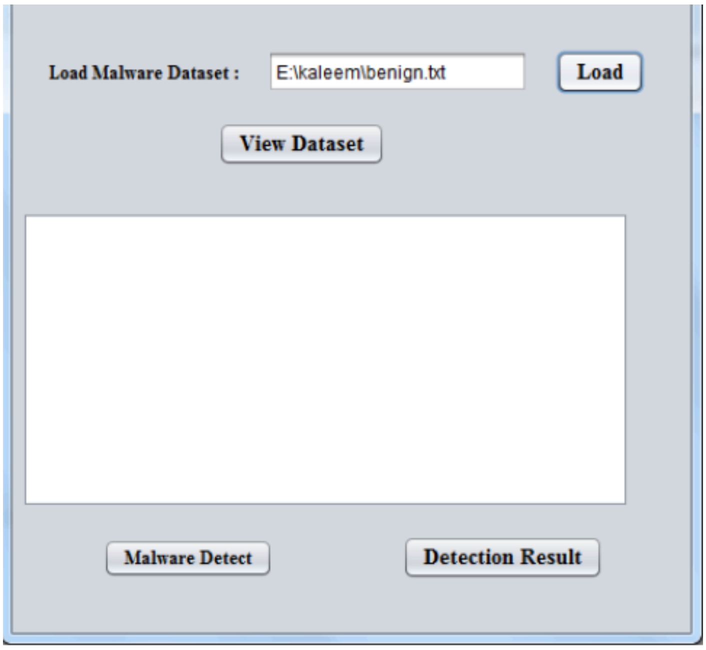


Malware Propagation By Using Hybrid Adaptive Neuro-Fuzzy Inference System

User Login

Figure 2: Authentication screen.

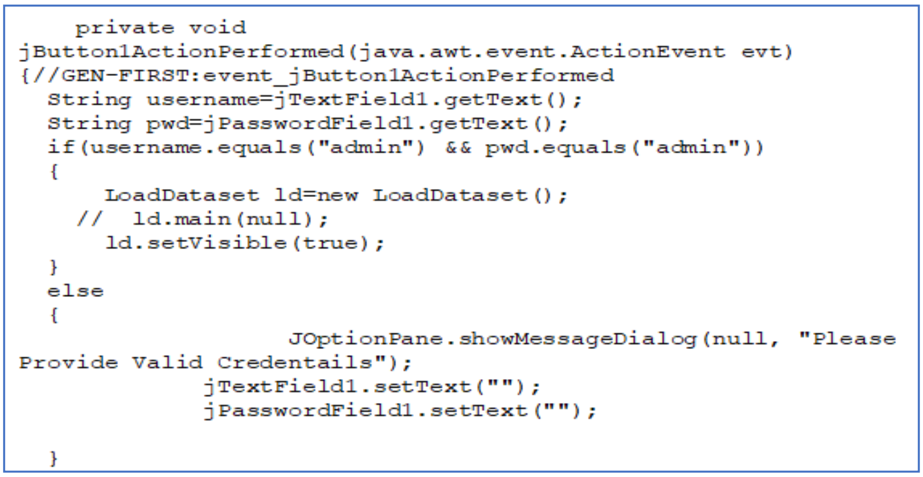
After due authentication, the user is forwarded to the screen in Figure 3.



Load Malware Dataset : E:\kaleem\benign.txt

Figure 3: Helps in loading dataset for malware detection.

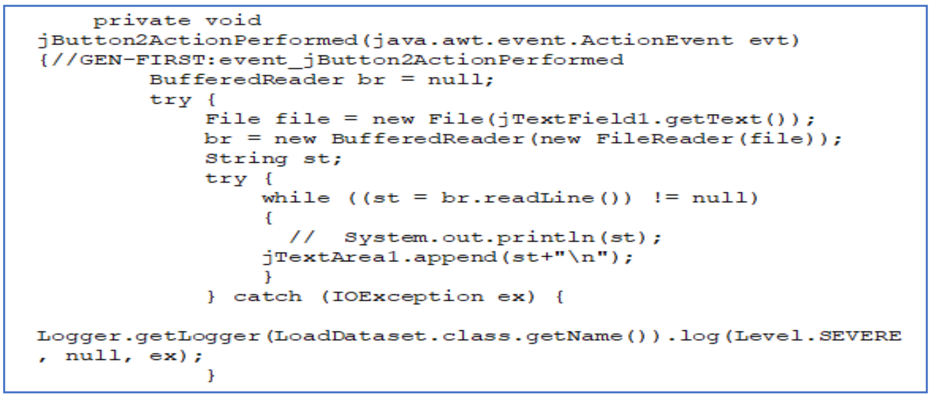
As presented in Figure 3, the application helps in loading the dataset.



```
    private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
  String username=jTextField1.getText();
  String pwd=jPasswordField1.getText();
  if(username.equals("admin") && pwd.equals("admin"))
  {
      LoadDataset ld=new LoadDataset();
    //  ld.main(null);
      ld.setVisible(true);
  }
  else
  {
                   JOptionPane.showMessageDialog(null, "Please
Provide Valid Credentails");
             jTextField1.setText("");
             jPasswordField1.setText("");

  }
```

Listing 1: Authentication.

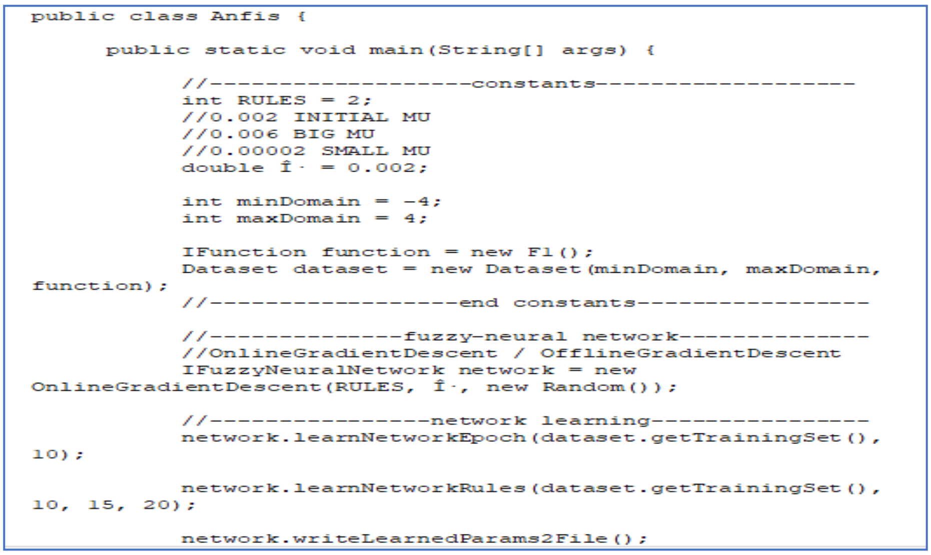
As presented in Listing 1, user credentials are authenticated before proceeding further.



```
    private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
        BufferedReader br = null;
        try {
            File file = new File(jTextField1.getText());
            br = new BufferedReader(new FileReader(file));
            String st;
            try {
                while ((st = br.readLine()) != null)
                {
                  //  System.out.println(st);
                jTextArea1.append(st+"\n");
                }
            } catch (IOException ex) {

Logger.getLogger(LoadDataset.class.getName()).log(Level.SEVERE
, null, ex);
            }
```

Listing 2: Source code for loading dataset.

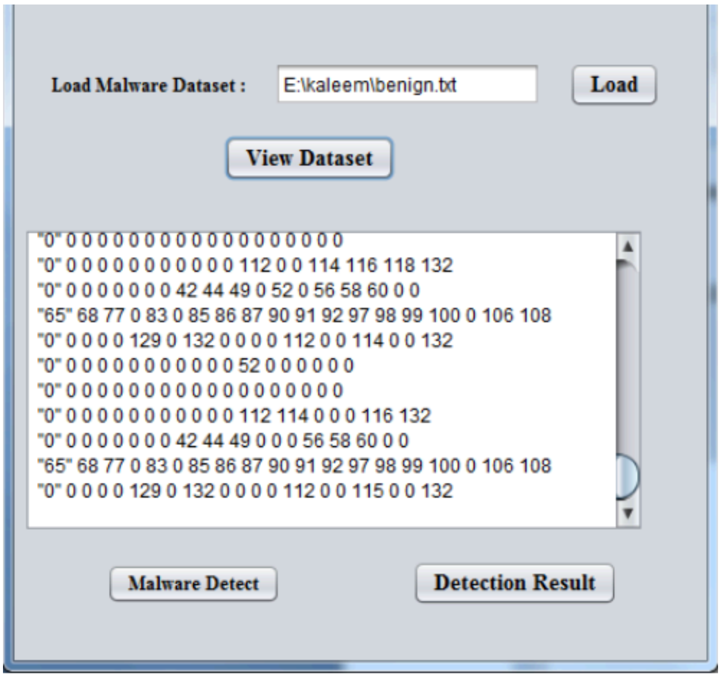
As presented in Listing 2, the code is meant for loading dataset into the GUI form.



```
public class Anfis {

     public static void main(String[] args) {

          //--------------------constants--------------------
          int RULES = 2;
          //0.002 INITIAL MU
          //0.006 BIG MU
          //0.00002 SMALL MU
          double Î· = 0.002;

          int minDomain = -4;
          int maxDomain = 4;

          IFunction function = new F1();
          Dataset dataset = new Dataset(minDomain, maxDomain,
function);
          //------------------end constants------------------

          //-------------fuzzy-neural network-------------
          //OnlineGradientDescent / OfflineGradientDescent
          IFuzzyNeuralNetwork network = new
OnlineGradientDescent(RULES, Î·, new Random());

          //---------------network learning---------------
          network.learnNetworkEpoch(dataset.getTrainingSet(),
10);

          network.learnNetworkRules(dataset.getTrainingSet(),
10, 15, 20);

          network.writeLearnedParams2File();
```

Listing 3: Main class of the proposed system.

As presented in Listing 3, the main class of the proposed system is provided. It is the starting point of the whole application.

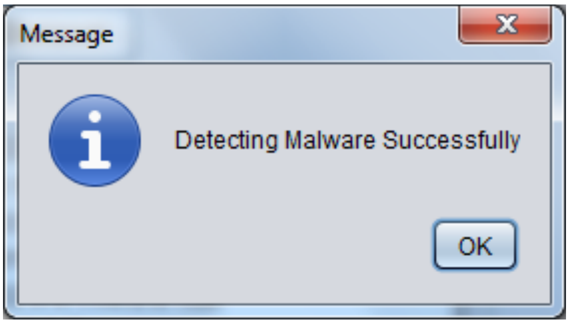


Load Malware Dataset : E:\kaleem\benign.txt

```
"0" 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
"0" 0 0 0 0 0 0 0 0 0 0 0 112 0 0 114 116 118 132
"0" 0 0 0 0 0 0 0 42 44 49 0 52 0 56 58 60 0 0
"65" 68 77 0 83 0 85 86 87 90 91 92 97 98 99 100 0 106 108
"0" 0 0 0 0 129 0 132 0 0 0 0 112 0 0 114 0 0 132
"0" 0 0 0 0 0 0 0 0 0 0 0 52 0 0 0 0 0 0
"0" 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
"0" 0 0 0 0 0 0 0 0 0 0 0 112 114 0 0 0 116 132
"0" 0 0 0 0 0 0 0 42 44 49 0 0 0 56 58 60 0 0
"65" 68 77 0 83 0 85 86 87 90 91 92 97 98 99 100 0 106 108
"0" 0 0 0 0 129 0 132 0 0 0 0 112 0 0 115 0 0 132
```

Figure 4**:** Shows loaded the dataset.

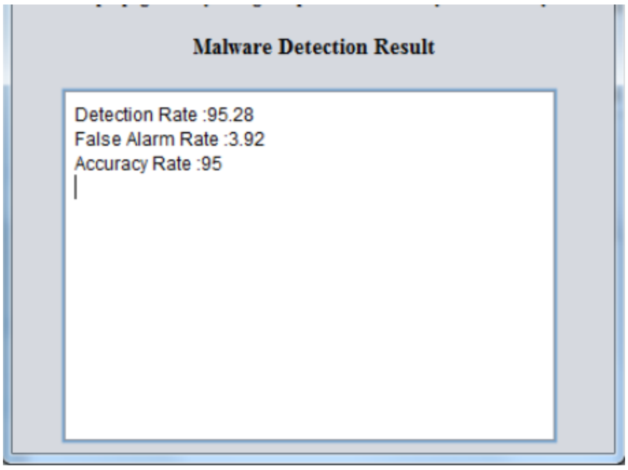
As presented in Figure 4, it is understood that the dataset is loaded for visualization. Once the dataset is loaded, it is possible to choose the "Malware Detect" button. On selecting this button, the algorithm performs its functions and provides an appropriate message. On completion of the algorithm, it shows the message, as shown in Figure 5.



Detecting Malware Successfully

Figure 5**:** Result message of malware detection.

As presented in Figure 5, malware detection is made successfully. It is then possible to have the performance metrics for evaluation. The performance metrics are then compared with state of the art later.



Malware Detection Result

Detection Rate :95.28
False Alarm Rate :3.92
Accuracy Rate :95

Figure 6: Shows the performance metrics such as prediction rate, false alarm rate and accuracy rate.

As presented in Figure 6, it is understood that the detection rate of the proposed system is 95.28, its false alarm rate is 3.92, and the accuracy rate is 95. These results are discussed and evaluated in the ensuing section.

5 References

[1] Ganeshkumar, P., & Pandeeswari, N. (2015). *Adaptive Neuro-Fuzzy-Based Anomaly Detection System in Cloud. International Journal of Fuzzy Systems, 18(3), 367–378.*

[2] Roshna R.S, Vinodh Ewards. (2013). Botnet Detection Using Adaptive Neuro Fuzzy Inference System. *International Journal of Engineering Research and Applications,*3 (2), p1-6.

[3] Altyeb Altaher And Omar Mohammed Barukab. (2017). Intelligent Hybrid Approach for Android Malware Detection based on Permissions and API Calls. *International Journal of Advanced Computer Science and Applications*,8 (6), p1-8.

[4] Hans, K., Ahuja, L., & Muttoo, S. K. (2017). *An adaptive neuro-fuzzy inference system for detecting redirection spam. 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO),* P1-6.

[5] Sumedh Pundkar and Pratik R. Upadhye. (2015). Self Evolving Antivirus Based on Neuro-Fuzzy Inference System. *International Journal of Research in Engineering and Science,*3 (6),p06-09.