National College of Ireland

Prevention and Propagation of Malware by Using Hybrid Adaptive Neuro-Fuzzy Interface System

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
Supervisor:	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:	4754 **Page Count:** 19

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):	

Prevention and Propagation of Malware by Using Hybrid Adaptive Neuro-Fuzzy Interface System

Kaleemuddin Mohammed
x18132855

Abstract

In the wake of cybersecurity issues and the ever-growing incidents of cyber-attacks, it is essential to have a mechanism in a pace that can protect information system from malicious attacks. Malware detection is thus given importance. Adaptive Neuro-Fuzzy Interface System (ANFIS) is the algorithm with the machine learning approach which employs both fuzzy logic and also neural network. By making a hybrid of them, it provides a fuzzy inference system that can detect malware. In this project, the ANFIS algorithm is implemented using the Java programming language. It takes malware dataset and provides malware detection mechanism. It makes use of membership functions and the combination of fuzzy logic principles and neural network to have potential benefits of both the approaches in a single framework. A prototype application is built to show the effectiveness of the hybrid ANFIS algorithm for malware detection besides comparing its performance with state of the art.

1 Introduction

The world has witnessed technological innovations and rapid growth of connected devices due to distributed computing and the Internet of Things (IoT). Unfortunately, it also carries a full spectrum of security challenges due to its distributed nature, the involvement of multiple OEMs, still inadequate global standards and a host of other aspects such as social engineering and the man-made invasions on distributed systems. Cybersecurity has become indispensable for all nations in ever-growing civilization and digitalization.

Cybersecurity is to be given highest importance to protecting systems from malicious attacks. It is essential to secure critical digital infrastructure. There has been increased in the malware attacks. Every day, there are incidents of malware somewhere in the world. Recent attacks with WannaCry showed its impact in multiple countries. Despite increasing efforts from various government organizations and private organizations, the existing solutions are still inadequate to protect the contemporary Internet applications that are distributed in nature. Such applications are exposed to inherent vulnerabilities of distributed networks due to the aforementioned reasons. The traditional security measures like firewall, authentication and cryptography used as the first line of defense are not sufficiently provided the full

landscape of security challenges in the heterogeneous and distributed networks of the real world. Therefore, there is a need for Malware Detection Systems that are to be integrated with anti-virus and other security mechanisms to ensure end-to-end security. The combination of the two lines of defense with provision for end-to-end security mechanism can prove to be a solid defense against various threats. This project is aimed at the implementation of Adaptive Neuro-Fuzzy Inference System (ANFIS) that exploits fuzzy rules and also a neural network for prediction of malware in a better way.

The proposed inference system has a model that maps inputs to corresponding membership functions, rules associated with membership functions, and the rules that are to set output features. Then output features are mapped to output membership functions. The proposed malware detection model is implemented using the Java programming language. It is evaluated with malware dataset collected from the UCI machine learning repository. It is evaluated with different performance metrics like accuracy, false alarm rate and detection rate. It is also compared with state-of-the-art detection models and found to have better performance over them.

2 Related Work

This section reviews literature related to the anomaly detection methods available in the literature. Especially, it focuses on malware detection issues and countermeasures. Ganesh Kumar and Pandeeswari [1] used the ANFIS algorithm for building a malware detection system for cloud computing. They used the virtual environment with the hypervisor in order to have a better application of ANFIS. They developed a hypervisor detector as part of the cloud-based Intrusion Detection System (IDS). Their architecture includes hardware, hypervisor detector, set of Virtual Machines (VMs) and associated operating systems. Hypervisor detector is designed to have ANFIS built into it for making an inference system. Roshna and Ewards [2], on the other hand, employed ANFIS algorithm for botnet detection. A botnet is a set of bots that are malicious in nature. They used ANFIS as a detection mechanism for identifying botnet. In other words, ANFIS characterizes botnets and detects them as part of the underlying inference system. The pattern recognition is made with the help of ANFIS algorithm. It has different steps like input traffic, traffic reduction, feature extraction and ANFIS pattern recognition. Then the infected details are furnished to reflect either legitimate traffic or malicious traffic.

Altaher and Barukab [3] proposed an intelligent and hybrid mechanism for malware detection in Android Apps. The detection process is based on API calls and permissions. They employed both ANFIS and Particle Swarm Optimization (PSO) to achieve this. Membership functions are tuned in order to optimize the parameters of ANFIS. They developed a hybrid classifier known as PSO-ANFIS classifier. A confusion matrix is used to evaluate the efficiency of the system. Accuracy is measured in terms of RMSE. Their method showed better performance over other variants of ANFIS. Hans et al. [4] proposed ANFIS based method to detect redirection spam. They also used a clustering approach to have to pre-process in order to reduce training time. They employed fuzzy inference system along with rules required to detect redirection spam. They could find malicious redirections and distinguish genuine redirections from them. Their algorithm is useful for different applications where URLs are used as part of the application that includes possible redirections for different reasons like shortening address.

Pundkar and Upadhye [5] focused on a self-evolving anti-virus with the help of a neuro-fuzzy inference system. This system has learning mechanisms that catch up to the actual speed of virus programs. As the method is self-adaptive, it learns from time to time with new training data. Thus, its knowledge will be incremental, and it will have improved heuristics as time goes on. It knows the files and signatures are viruses as it learns from an available training set. ANFIS is part of the system that helps in leering and provide inference capabilities. Once a model is made, it will get improved as new training is made. Thus, it is an adaptive system to enhance itself to cope with new kinds of traffic models.

Altaher [6] proposed an Android malware detection system with a hybrid neuro-fuzzy classifier. It could detect obfuscated malware as it throws challenges to detection mechanism. Permission-based features are employed as the underlying process in the proposed methodology. It uses fuzzy rules in order to detect obfuscated malware and also learn its structure and learn to get adapted to new malware structures from time to time. Thus, it is dynamically evolving. The model developed initially gets updated and thus improves classification accuracy. It has feature selection mechanisms in order to improve the performance of classification. Each permission-based feature is given a score as part of the detection mechanism. Both confusion matrix and RMSE based accuracy are used for evaluation.

Raut and Singh [7] focused on adaptive neuro-fuzzy inference system based on entropy for feature reduction. The feature redirection mechanism employed uses entropy for finding the attributes that can contribute to the class label. It employs both closed loop and

open loop feature selection methods. They employed both ANFIS and rough set theory to achieve a prediction model. KDD Cup'99 datasets are used for empirical study. It has feature selection, ANFIS training and ANFIS testing phases in order to classify unlabeled data. Three kinds of feature selection methods are used before ANFIS training. They include the entropy-based approach, open-loop approach and closed-loop approach. Each reduction technique could reduce features to get rid of the curse of dimensionality problem.

Shalaginov and Franke [8] proposed a novel method based on fuzzy patches construction for detecting malware. They employed the Gaussian method as the underlying membership function. Their algorithm has phases like rule discovery procedure, clustering for feature selection, finding elliptic regions, finding parameters for fuzzy patches, construction of membership functions and tuning of the proposed classification model. They intended to use the non-parametric distribution approach in future instead of the Gaussian model. Shalaginov et al. [9] focused on multi-nominal malware detection approaches. Multinomial classification, according to them, is little explored, and they followed it for better performance. Their method has the capability to generate rules that are generalized. They used training data from PEframe and VirusTotal for improving quality of classification algorithms. Different malware families are used in training, and the system evaluated with diversified malware in the testing set.

Komar et al. [10] proposed an adaptive system for high performance to detect cyber-attacks. They used fuzzy address analyzer for detecting malware and make well-informed decisions. Fuzzy address analyzer has trained with historical data and applies that knowledge to current data. It has efficient search mechanisms such as minimum code search and gravity centre search. Their simulation results showed that it could detect malware, threat and vulnerabilities. Huda et al. [11] employed both filter and wrapper-based feature selection methods in order to have better quality in training. This could eliminate redundant and irrelevant features in the dataset. Thus, it could improve the training quality. They employed Support Vector Machine (SVM) for classification. They gained heuristics from system calls, and API calls related statistics. The filtered phase was used to reduce features and then the wrapper phase was used to optimize the features. Both obfuscated and benign samples are used for effective training. Due to the heuristics from the wrapper method, their malware detection system could improve performance over state of the art.

Shalaginov [12] improved the process of making fuzzy sets along with neuro-fuzzy inference mechanism for malware detection. They employed dynamic feature-based expansion. The notion of model retraining is used. They proposed an algorithm named

Dynamically Expanded Neuro-Fuzzy (DENF) inference system. It could provide security proactively when used for intrusion detection. Jabez and Muthukumar [13] proposed an outlier detection method as part of an IDS that detect malware. Their method could detect attacks with considerable efficiency. Their system could find anomalies with ease. Zhao et al. [14] proposed a deep learning-based approach for malware detection. They employed Long Short-Term Memory (LSTM) algorithm for leering patterns in the training phase to form a detection system that provides better prediction performance.

Dovom et al. [15] proposed an edge malware detection mechanism. They used a data structure known as a fuzzy pattern tree as the underlying detection mechanism. They employed approaches for both categorization and malware and its detection in Internet of Things (IoT) environment. They analyzed OpCodes of applications by converting them into a vector space. Barraclough et al. [16] proposed a neuro-fuzzy based framework for intelligent detection of phishing attacks. Their system was able to generate fuzzy rules require for decision making on phishing attacks. They tuned parameters and experimented on them to obtain improved results. In future, they intended to improve their method with different feature sets. Firdausi et al. [17] proposed a novel paradigm for malware detection. It is known as bio-inspired malware detection paradigm. They used Artificial Neural Network (ANN), Multi-Layer Perceptron (MLP) and Radial Basis Function Network (RBFN). They found better performance in prediction with the MLP approach. In future, they intended to focus on the division of malware into different classes.

Al-Duwairi et al. [18] proposed a system known as BotDigger, which is used to detect botnets. They used a fuzzy inference system to achieve this. The architecture of BotDigger contains an input layer, set of rules and output layer. The fuzzy inference system is based on the well-known Sugeno model.

3 ANFIS STRUCTURE

ANFIS is used to solve many real-world problems such as parameter identification and malware detection, to mention few. It is based on fuzzy rules and a neural network that has effective mechanisms to have prediction models. It supports learning rules, combining fuzzy rules and neural networks for parameter identification. Graphical network representation of ANFIS is shown in Figure 1. It is based on Sugeno-type fuzzy systems that have built-in capabilities for neural learning. The network contains nodes and layers with specific

functions. It supports usage of fuzzy rules with IF-THN model. When the Sugeno type of fuzzy system is considered, the rule base is shown as follows.

1. If x is A1 and y is B1, then $f_1 = c_{11}x+c_{12}y+c_{10}$
2. If x is A2 and y is B2, then $f_2 = c_{21}x+c_{22}y+c_{20}$ [3]

If the membership functions of fuzzy sets are provided as Ai, Bi, i=1,2, be, μ_{Ai} μ_{Bi}. Rules can be evaluated.

1. Results of rule premises can be evaluated as:

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y),\ i=1,2.$$

2. The evaluation of rule consequences and implications can be done as:

$$f(x,y) = \frac{w_1(x,y)f_1(x,y) + w_2(x,y)f_2(x,y)}{w_1(x,y) + w_2(x,y)}.$$

When arguments are left out:

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

This is possible to separate by defining:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}$$

Afterwards, the function can be defined as:

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2$$ [3]

The operations are shown graphically in Figure 1. ANFIS generally has neurons in five layers. The neurons in the same layer can have similar functionality.

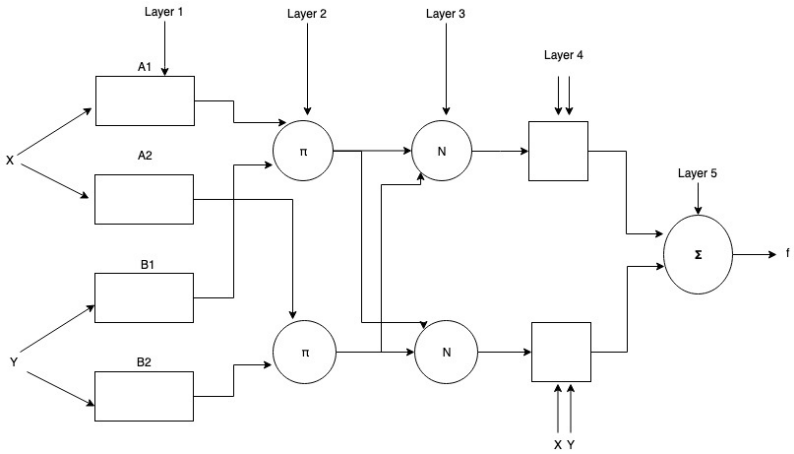

Figure 1: Overview of ANFIS structure

As presented in Figure 1, the ANFIS structure has many layers. It is suitable for inferring rules based on the underlying Sugeno fuzzy model. It has rules such as:

Rule1:

If x is A1 and y is B1, then f1 = p1x + q1y + r1

Rule2:

If x is A1 and y is B1, then f1 = p1x + q1y + r1

This way, fuzzy rules are used, and they are combined with the neural network model. Thus, the ANFIS functionality is built for malware detection in this project. ANFIS has five layers, as shown in Figure 2.

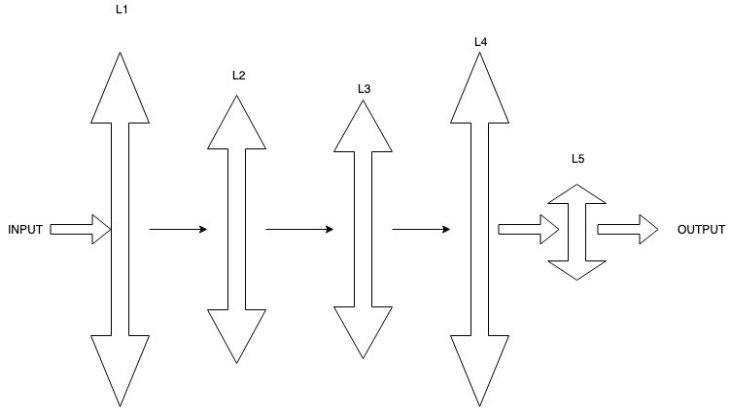

Figure 2: Layers of ANFIS model

It has different layers. It follows a feed-forward neural network with the capability of supervised leering. Nodes and links are connected as part of the neural network. The nodes are adaptive in nature with leering capabilities and thus infer the malware when it is implemented for malware detection. More on the layers is shown below.

Layer 1 (L1): In this layer, membership grades of the linguistic label are made by each node.

Layer 2 (L2): In this layer, each node computes the strength of rules by employing operators like prod or min besides using any fuzzy AND operations.

Layer 3 (L3): In this layer, the nodes compute ratios related to firing strength of rules, and the total strength is found besides normalization of the firing strength.

Layer 4 (L4): In this layer, the nodes calculate parameter functions associated with layer three outcomes. The consequent name parameters are used to denote the

parameters in this layer.

Layer 5 (L5): In this layer, aggregation of outputs is made by a single node.

The flow of the ANFIS model is, as shown in Figure 3. The flow includes the loading of training data, setting input parameter membership, training the actual ANFIS model, obtain the best inference system, input matching parameters for prediction and arrive at the prediction results.

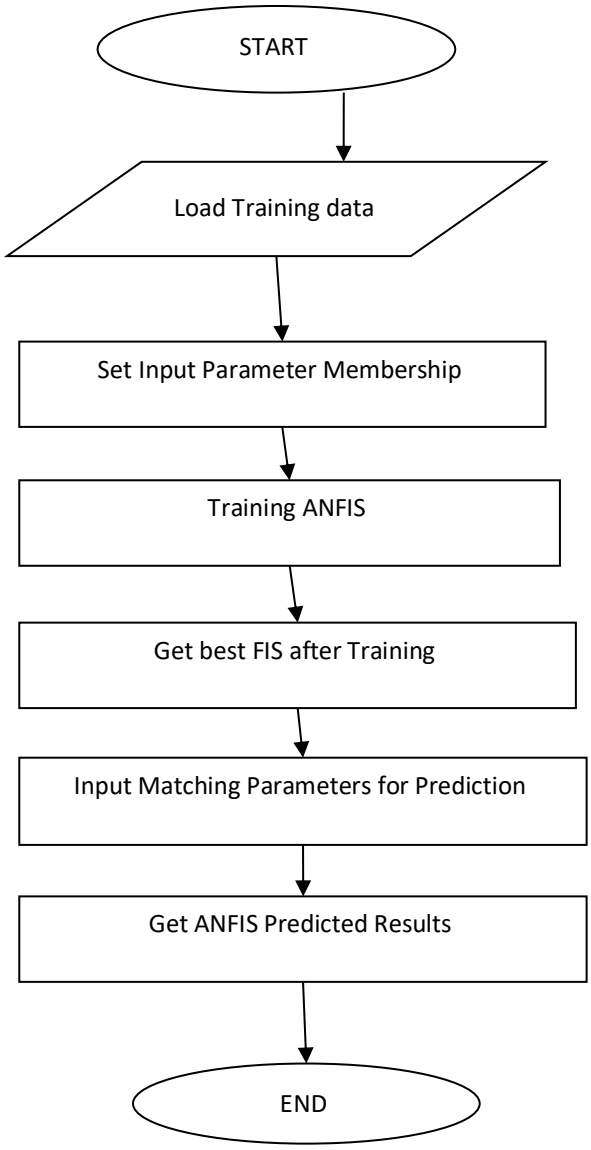

Figure 3: Flow chart of ANFIS based prediction model

As shown in Figure 3, the flow chart provides modus operandi of the ANFIS based prediction model for malware detection. The fuzzy inference system and neural network associated with

the model work for supervised leering in order to have a knowledge model that can be used to detect malware. Figure 4 shows the generalized fuzzy inference model.

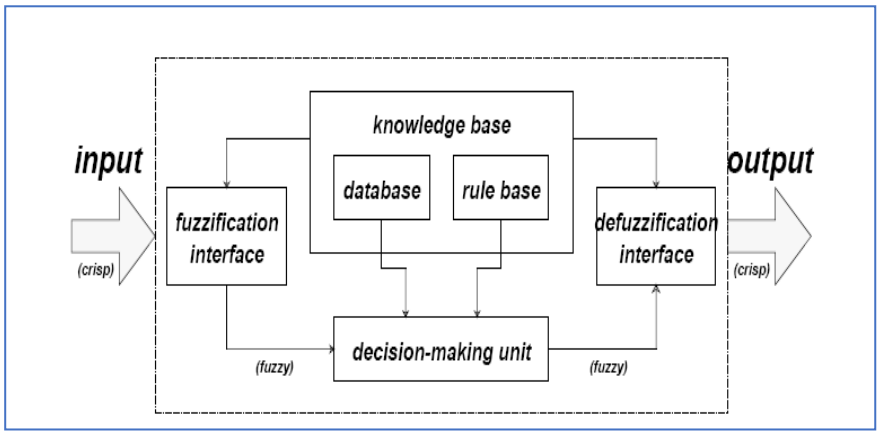

Figure 4: Adaptive Neuro-fuzzy system for malware detection

As presented in Figure 4, the model is used for the adaptive system for malware detection. It has inputs, outputs, russification interface, defuzzification interface besides knowledge base and decision-making unit. The knowledgebase contains database as well as a rule base. There is a verification process for the attack. The threshold is used for detection. The output <0.1 is considered to be normal. For possible attack, the verification is made as 0.1< output < 0.5 and for actual attack it is >0.5.

ANFIS algorithm can have parameters that are used with a linear combination of consequent parameters. The function f is written as follows:

$$f = (\bar{w}_1 x)c_{11} + (\bar{w}_1 y)c_{12} + \bar{w}_1 c_{10} + (\bar{w}_2 x)c_{21} + (\bar{w}_2 y)c_{22} + \bar{w}_2 c_{20} \quad [3]$$

There are linear consequent parameters denoted as c_{ij} (i = 1,2, j = 0,1,2). In the case of a hybrid algorithm, the consequent parameters are adjusted to has forward passed and also backward pass. This will help the inputs propagated to different layers and finally get the desired outputs. There is an update of rules such as consequent and premise and decoupled in the rule associated with hybrid learning. Thus, there is a possibility of speed up in computations by employing variants of optimization methods and gradient methods that work on the premise parameters.

4 Design Specification

This section to implement a proposed model for malware detection, the following software is used.

1. JDK 1.8
2. NetBeans

JDK 1.8 is the Java Development Toolkit which is essential for developing applications using the Java platform. It is the basis for Java language and its built-in API available in different packages. NetBeans is the Integrated Development Environment (IDE) which has a complete development environment required for building Java applications. It is used to develop a GUI application to demonstrate proof of the concept.

4.1 Proposed Algorithm

An algorithm is proposed to have an adaptive approach for neuro-fuzzy inference system. The algorithm follows a hybrid mechanism for effective detection of malware.

Algorithm: Hybrid Adaptive Neuro-Fuzzy Inference System

Inputs: Training Data D

Output: Fuzzy inference system to detect malware

1. Start
2. Load D
3. Set Initial Parameters P
4. Determine membership functions M
5. Complete fuzzification
6. Determine training epochs
7. Training the ANFIS classifier
8. Get the best fit after training□
9. Use matching parameters for prediction
10. Perform prediction on test data
11. Generate prediction results
12. Evaluate performance
13. Stop

Algorithm 1: A Proposed algorithm based on ANFIS

As presented in Algorithm 1, the proposed algorithm takes loads training set and set up initial parameters as needed. Then it determines membership functions and completes the fuzzification process. Training epochs are considered, and ANFIS classifier is trained in the training data. With the training, ANFIS becomes the best fit with knowledge for classification. It uses matching parameters for prediction and carries out the prediction process. One prediction is made, it eventually provides prediction results and performance details.

5 Implementation

This section describes the implementation of the proposed model. We have developed an application in JAVA programming language by using NetBeans IDE. The application needs user authentication. After authentication, it allows users to perform malware detection.

As presented in Figure 5, the proposed system has a user authentication mechanism. This is to prevent unauthorized access to the system. The authentication process takes user credentials as input and then perform authentication to know whether the user is a valid user. On the successful authentication, the user is redirected to the actual malware detection phase.

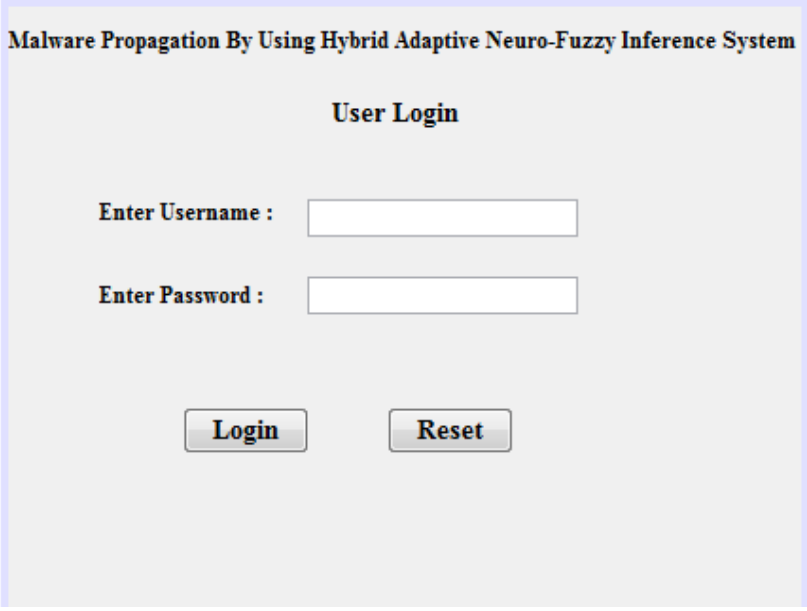

Figure 5: User authentication.

As presented in Figure 6, the malware detection system proposed has the interface to load any malware dataset, as here I used a dataset from VirusSign website [19] which is for free. by using this dataset, the proposed system has been developed. The “Load” button helps in browsing the dataset from a local system or remote system. Once the dataset is specified, it

can be viewed. The "View Dataset" button can help in viewing the actual dataset prior to employing the proposed ANFIS algorithm. The "Malware Detect" button, is used to perform the functionality of the ANFIS, which has the intended functionality to learn and detect malware.

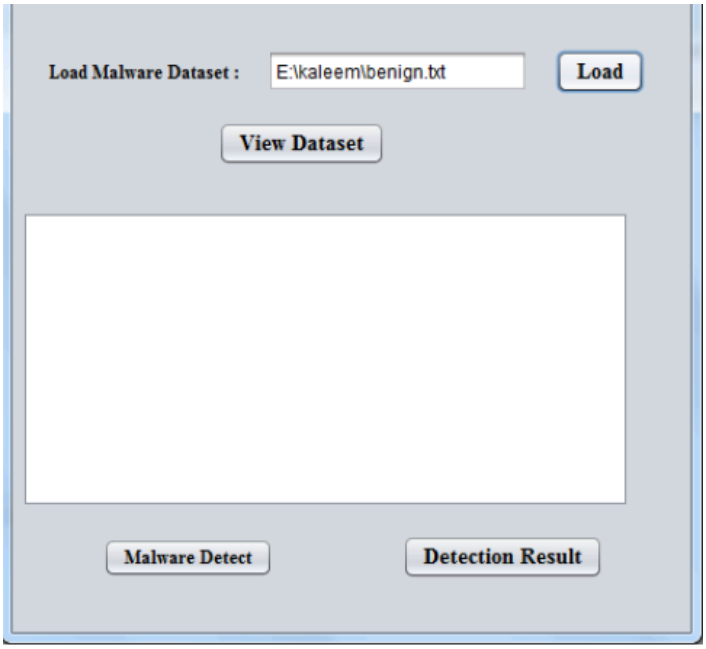

Figure 6: Malware detection phase.

As presented in Figure 7, 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 choosing 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 4.

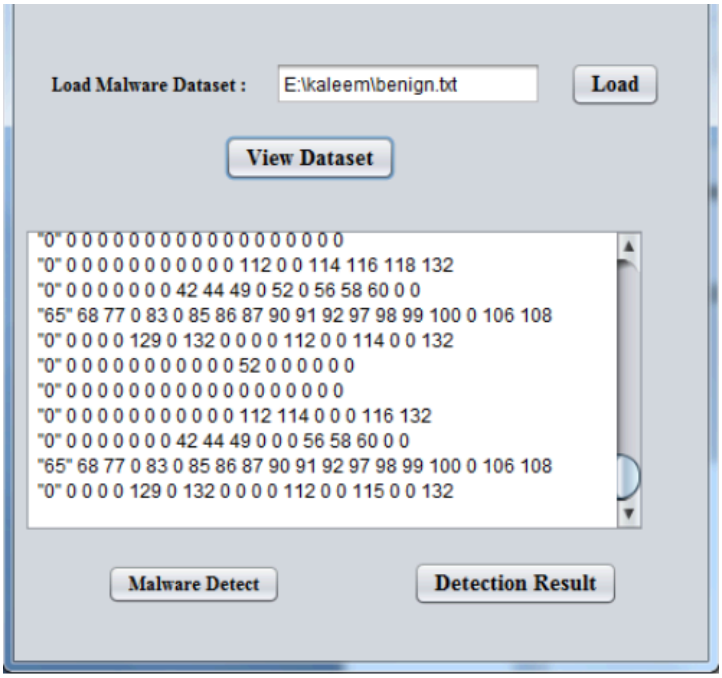

Figure 7: Shows loaded the dataset.

As presented in Figure 8, 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.

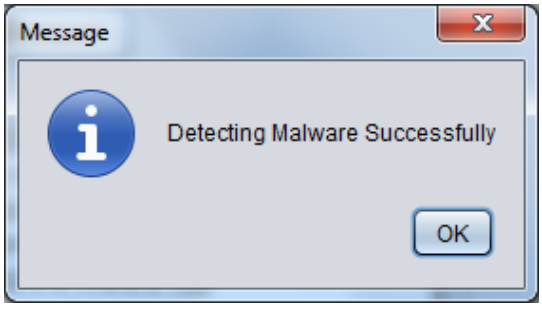

Figure 8: Result message of malware detection.

As presented in Figure 9, 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.

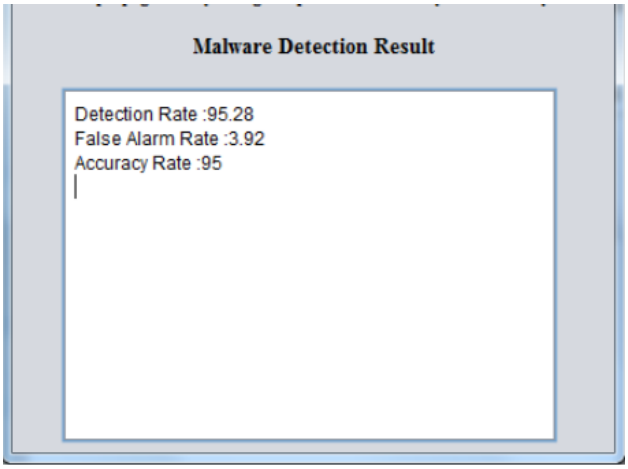

Figure 9: Shows the performance metrics such as prediction rate, false alarm rate and accuracy rate.

6 Evaluation

Experiments are done with malware dataset on which the proposed hybrid ANFIS algorithm is employed in order to detect malware. Observations are made in terms of detection rate, false alarm rate and accuracy.

Algorithm Name	Performance		
	Detection Rate	False Alarm Rate	Accuracy
ANFIS	95.28	3.92	95
Naïve Bayes	93.877	11.538	91.08
Random Forest	84.783	11.9047	86.364
SMO	91.837	6.122	92.857

Table 1: Shows the performance of different detection models

As can be seen in Table 1, it is evident that the experimental results are provided in terms of detection rate, false alarm rate and accuracy for different detection methods, including the ANFIS.

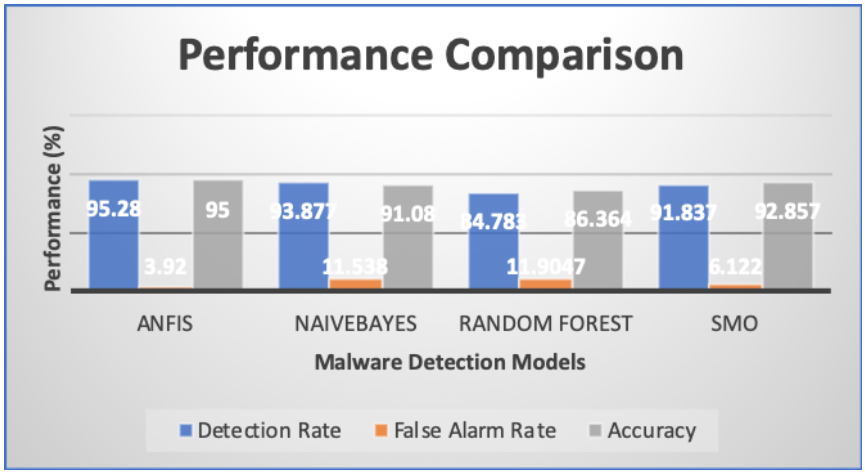

Figure 10: Performance comparison

As presented in Figure 10, the performance of different prediction modes is compared, including the ANFIS algorithm. Different malware detection models are provided in a horizontal axis while the vertical axis shows different performance metrics like accuracy, false alarm rate and detection rate. The detection rate of ANFIS is 95.28% which is higher than other detection methods like Naïve Bayes, RF and SMO. Similarly, the false alarm rate of ANFIS is 3.92, which is lower than other detection methods. The accuracy of the ANFIS method is 95% which is higher than other prediction models.

6.1 Case Study

Malware detection is the case study used for the project. The dataset is related to malware issues in mobile applications. The dataset contains details of many mobile applications. Each instance is a mobile application related data.

As shown in Listing 1, a part of data is provided here. The data contains information related to different permissions in the mobile applications where the presence of permission provides significant evidence towards determining class label.

6.2 Discussion

As presented in Table 2, the execution time is computed and provided against different prediction models.

Prediction Model	Execution Time (sec)
ANFIS	0.47
NaiveBayis	0.87
Random Forest	1.23
SMO	1.03

Table 2: Performance comparison in terms of execution time

As presented in Figure 11, different prediction models are compared in terms of execution time. The prediction models used for malware detection are shown in a horizontal axis, while the vertical axis shows the time taken by each prediction model. The results revealed that the prediction models showed varied performance. ANFIS takes 0.47 seconds. Naïve Bayes took 0.87 seconds. Random Forest needed 1.23 seconds time while SMO needed 1.03 seconds. Based on this analysis, it is understood that the proposed method outperformed existing approaches.

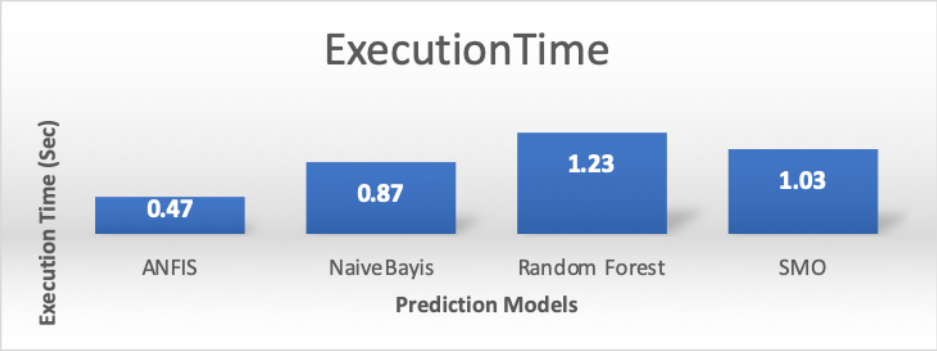

Figure 11: Execution time comparison

7 Conclusion and Future Work

The main focus of this research is to implement a hybrid approach with ANFIS algorithm that exploits fuzzy rules and neural network. The algorithm uses membership functions in order to achieve better results. It is based on an underlying inference system known as Takagi–Sugeno fuzzy. The ANFIS is implemented using Java programming language. An application is built with an intuitive user interface. A dataset collected from the UCI machine learning repository is used for empirical study. The application is implemented and executed with the given dataset. The empirical results revealed that the utility of the proposed system. The system is compared with many states of the art prediction models like Naïve Bayes, SMO and RF. The results revealed that the ANFIS outperforms other models in terms of accuracy, false alarm rate and detection rate. In future, the proposed malware detection system will be improved with many changes. First, it is implemented for big data processing that exploits the power of parallel computing. Second, it will be used in IoT based applications where large volumes of data are generated, and malware detection is essential. Third, the algorithm is enhanced to support different domain-specific datasets.

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.

[6] Altaher, A. (2016). An improved Android malware detection scheme based on an evolving hybrid neuro-fuzzy classifier (EHNFC) and permission-based features. *Neural Computing and Applications*, 28(12), 4147–4157.

[7] A. S. Raut and K. R. Singh. (2014). ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM FOR ANOMALY-BASED INTRUSION DETECTION. *International Journal of Research in Engineering and Applied Sciences*, 2 (2), p1-10.

[8] Andrii Shalaginov and Katrin Franke. (2015). A new method of fuzzy patches construction in Neuro-Fuzzy for malware detection. *16th World Congress of the International Fuzzy Systems Association*, p1-8.

[9] Shalaginov, A., Grini, L. S., & Franke, K. (2016). Understanding Neuro-Fuzzy on a class of multinomial malware detection problems. *2016 International Joint Conference on Neural Networks (IJCNN)*, P1-8.

[10] Myroslav Komar, Volodymyr Kochan, Lesia Dubchak and Anatoliy Sachenko. (2017). High-performance adaptive system for cyber-attacks detection. *IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, p1-4.

[11] Huda, S., Abawajy, J., Alazab, M., Abdollalihian, M., Islam, R., & Yearwood, J. (2016). Hybrids of support vector machine wrapper and filter based framework for malware detection. *Future Generation Computer Systems*, 55, 376–390.

[12] Shalaginov, A. (2017). Dynamic feature-based expansion of fuzzy sets in Neuro-Fuzzy for proactive malware detection. *2017 20th International Conference on Information Fusion (Fusion)*, P1-8.

[13]. Jabez, J., & Muthukumar, B. (2015). Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach. *Procedia Computer Science*, 48, 338–346.

[14] Zhao, B., Han, J., & Meng, X. (2017). A malware detection system based on intermediate language. *2017 4th International Conference on Systems and Informatics (ICSAI)*, P1-7.

[15] Dovom, E. M., Azmoodeh, A., Dehghantanha, A., Newton, D. E., Parizi, R. M., & Karimipour, H. (2019). Fuzzy Pattern Tree for Edge Malware Detection and Categorization in IoT. *Journal of Systems Architecture*, P1-7.

[16] Barraclough, Phoebe, Sexton, Catherine, Hossain, Alamgir and Aslam, Nauman (2014) Intelligent phishing detection parameter framework for E-banking transactions based on Neuro-fuzzy. In: *Proceedings of 2014 Science and Information Conference. IEEE, pp*, 545-555.

[17] Firdaus, A., Anuar, N. B., Razak, M. F. A., & Sangaiah, A. K. (2017). A bio-inspired computational paradigm for feature investigation and malware detection: interactive analytics. *Multimedia Tools and Applications*, 77(14), 17519–17555.

[18] Al-Duwairi, B., & Al-Ebbini, L. (2010). BotDigger: A Fuzzy Inference System for Botnet Detection. *2010 Fifth International Conference on Internet Monitoring and Protection*,p1-6.

[19] VirusSign. (2019). *MalwareList.* [online] Available at: https://www.virussign.com/index.html [Accessed 11 Dec. 2019].