Framework for the Detection of Android Malware Using Artificial Immune System

Ndatsu, Z. & Adebayo, O.S.
Department of Cyber Security Science
Federal University of Technology
Minna, Nigeria
E-mails: zainab.pg6717@st.futminna.edu.ng, waleadebayo@futminna.edu.ng
Phone: +2348036108345

ABSTRACT

Artificial immune systems (AIS) are just computational systems that are inspired by theoretical immunology, observed immune functions, principles and mechanisms to solve problems including the detection of malware. AIS was used as optimizer for the selection of best features of android application. The aim of this paper is to propose an android malware classification technique for the detection of android malicious applications. The proposed framework consists of the basic approach and techniques to achieve good model for the detection of android malicious applications. The research methodology of Data Analysis, which involves validation through experimentation, is employed to achieve this. The results show that the models of selected permission-based features are more accurate than those models without the selection of features. The true positive rate and false alarm rate of selected features are also in better forms than those of classifying features without selection.

Keywords: malware, feature selection, classification models and Artificial immune syste

23rd iSTEAMS Conference Proceedings Reference Format

Ndatsu, Z. & Adebayo, O.S. (2020): Framework for the Detection of Android Malware Using Artificial Immune System. Proceedings of the 23rd iSTEAMS Conference, American University of Nigeria, Yola. April, 2020. Pp 117-126 www.isteams.net/yola2020

1. INTRODUCTION

The level of development of mobile devices have brought to people's ways of living cannot be veremphasised. The early phone can only be used to make calls and sending text messages. The arrival of smartphone has revolutionised the mode of communication and data processing activities. For example, with smartphone people no longer visit the cyber café, a popular internet office where people browse internet for various data communication activities. Everybody can remains in his or her convenient place and surf internet with related transaction. Again, with mobile phone, related banking transaction can be performed; foreign transaction can be carried out on the smart phone such as video conferencing, online application and many others. Although the emergence of smartphone has also brought little disadvantages like loss of job, loss of valuables at times over fraud transaction and others. However, the advantages of smartphone outweigh the associated disadvantages. According to Milosevic et al. (2017), it was predicted that there will be about 6.1 billion mobile device users by the year 2020. The smartphone arrives with different operating system (OS) which includes windows operating system, iOS, and android operating system. Among these, android OS is the most popular and friendly due to its openness and application availability in different open sources. This android OS is owned by Google corporation which has made the android application free in an open market.

This openness of application has made android a soft target for malicious softwares. Malware is a malicious software application intentionally built in a computing facility for nefarious purposes. It can also be termed as all kind of intrusions that is disastrous to the computer software and hardware system. Malware writer creates malware for purposes of economic gain, destruction, challenges or retaliation. Computer malwares include computer viruses, worms, Trojan, malicious mobile codes (Botnets, Nitda worm), Tracking Cokies (spywares, adwares, crimewares), Attacker Tools (Backdoors, Keylogger, Rootkits, E-mail generator) and other harmful software. A malware detector is a system introduced to analyze and identify malware. Malware detector can be a commercial virus scanner which uses binaries signature and other heuristic rules and algorithm to identify malware or firewall which monitors the gateways of electronic devices. Computer codes that come with malicious intents are referred to as malware (Adebayo et al., 2013). These malware target at android application include Trojan, Spyware, Ransomeware, Virus, Worm, among others. The attack techniques of malware on a smartphone are by creating a new process to launch its attack and by redirecting the program flow of a legitimate application (e.g. messaging activities) to execute its malicious code within a legitimate security context.

Android applications consist of two basic features: permission based and application programming interface (API). These two features are normally considered in the classification and detection of malicious applications. Recently there has been a drive to provide theoretical underpinnings of some of these algorithms. Artificial Immune Systems (AIS) are being used in many applications such as anomaly detection, pattern recognition, data mining, computer security, adaptive control and fault detection. Effort to detect, analyse and remove malware is in great demand across computing and mobile device platforms. Various methods and algorithms have been worked out by many researchers such as by Mirjalili and Lewis (2016), Siddiqui (2008), Eder et al. (2013), Christodorescu et al. (2005) and Shabtai et al. (2011) among others. Some closely related work that apply almost similar approach as this present research are Agrawal and Srikant (1994), Siddiqui (2008), and (Shabtai et al. (2011). A very common technique adopted by malware writer is code obfuscation (Abhijit et al., 2013); which prevents its detection by the detectors. Code obfuscation technique can be polymorphic or metamorphic. A metamorphic virus obfuscates by hiding itself completely to evade detection while a polymorphic virus obfuscate its decryption loops using code insertion and transposition (Christodorescu et al., 2005). Moreover, a metamorphic malware adopt methods like register renaming, dead code insertion, block reordering and command substitute in order to perform its malicious acts. Another technique adopted by malware writer is the modification and inclusion of new behaviour in their malware so as to increase its strength and viability.

Several solutions that have been adopted in the past in the analysis and detection of malware can be classified into static analysis, dynamic analysis and a hybrid of both static and dynamic methods. Static analysis is the process of analyzing a program's code statistically without actually executing the code (Drake et al., 2014). Static analysis scans the software for malicious patterns without installing it. In this static analysis on a smartphone, the sandbox decompresses installation files and disassembles corresponding executable. The static analysis approach has the advantage that an entire code can be covered and therefore, possibly a complete program behaviour, independent of any single path executed during run-time, will be easily captured. However, the statics analysis is constrained with its inability to detect new malware or new variants of malware. Dynamic analysis, on the other hand, is necessary to complement the lapses of static analysis due to various obfuscation mechanisms, which rendered static analysis an ineffective method. Dynamic analysis executes the application in a fully isolated environment, i.e. sandbox, which intervenes and logs low-level interactions with the system for further analysis using Android emulator which is normally used for testing and debugging ordinary Android applications.

The recent researches on the detection of malware on mobile platform include Framework for Analysing Android Applications (ANANAS) (Eder et al., 2013) and lightweight malware detection system for Android-based mobile devices (ANDROMALY) (Shabtai et al., 2011). Other android malware detection researches include MADAM, a Multi-level Anomaly Detector for Android Malware (Saracino et al., n.d.), Walenstein, Deshotels and Lakholia (2012), Holla and Katti (2013), Burguera, Zurutuza and Nadjm-Tehrani (2011) and Christodorescu et al. (2005).

The immune system is a system with high complexity and is under active research (from the biological point of view), likewise the current artificial immune system (AIS) works adopted only a few immune mechanisms. Specifically, three immunological principles are primarily used in a piecemeal in AIS methods. These include the immune network theory, the mechanisms of negative selection, and the clonal selection principles. Artificial Immune Systems (AIS) emerged in the 1990s as a new computational paradigm in AI. They are being used in many applications such as anomaly detection (Dasgupta, 1999c), pattern recognition (Cao, 2003), data mining (Knight, 2002), computer security (Hofmeyr, 2000, Dasgupta1999d, Kim, 2002b), adaptive control (Kumar; 1999, 2003) and fault detection (Bradley, 2000). Preliminary comparisons have also been made between AISs and other soft computing paradigms: ANN (artificial neural network), EA (evolutionary algorithms), FS (fuzzy system), and PR (probabilistic reasoning).

During the last five years, AIS has earned its position on the map of soft computing paradigm (Dasgupta et al., 2003). AIS are models of the immune system that can be used by immunologists for explanation, experimentation and prediction activities that would be difficult or impossible in 'wet-lab' experiments. This is also known as 'computational immunology.' Also AIS is an abstraction of one or more immunological processes. Since these processes protect us on a daily basis, from the ever-changing onslaught of biological and biochemical entities that seek to prosper at our expense, it is reasoned that they may be computationally useful (Garrett, 2005). Four major AIS algorithms have been constantly developed and gained popularity. These are: (1) negative selection algorithms (NSA); (2) artificial immune networks (AINE); (3) clonal selection algorithms (CLONALG); (4) the Danger Theory and dendritic cell algorithms (DCA). There have been many successful applications of AISs, such as computer security, optimization, data mining, and anomaly detection and so on (Dipankar Dasgupta et al., 2011).

2. RESEARCH METHODOLOGY

Next we discuss the various stages involved in the the process of achieving the aim of this research. These stages are also summarized in Figure 1.

2.1 Problem Identification

Arising from a thorough review of literature the following problem was identified: the false positive rate still relatively high, and the accuracy of the detection is not yet satisfactory. The successful application of artificial immune system in the detection of malware motivates the adoption of this algorithm. In order to remedy the lacuna of the existing researches, this research will use McNemar's test to measure the performance of the new model in relation to existing ones.

2.2 Problem Formulation

The problem formulation is to classify android applications into either malicious or benign application. In order to do this, android applications executable are represented using AP while the extracted android features were represented using AF. The malicious application (MA) and benign application as (BA).

The dataset and feature set were represented as:

$$AP = \begin{pmatrix} ap_{11} & ap_{12} & \cdots & ap_{1n} \\ ap_{21} & ap_{22} & & \vdots \\ \vdots & \vdots & & \\ ap_{m1} & ap_{m2} & \cdots & ap_{mn} \end{pmatrix} \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$AF = \begin{pmatrix} af_{11} & af_{12} & \cdots & af_{1n} \\ af_{21} & af_{22} & & \vdots \\ \vdots & \vdots & & \\ af_{m1} & af_{m2} & \cdots & af_{mn} \end{pmatrix} \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The feature is represented in $n \times n$ dimensional binary vector where 1 represents the presence of a feature and 0 denotes absence of a feature in an application.

$$AF = \begin{cases} 1 & if\ af\ is\ present\ in\ the\ application \\ 0 & if\ af\ is\ absent\ in\ the\ application \end{cases}$$

A function F such that AF → {MA, BA} is to be found using the defined selected features to train machine learning algorithms. This classification yields pairs $(ma_1, ba_1, ma_2, ba_2 \ldots ma_n, ba_n) \in (MA, BA)$

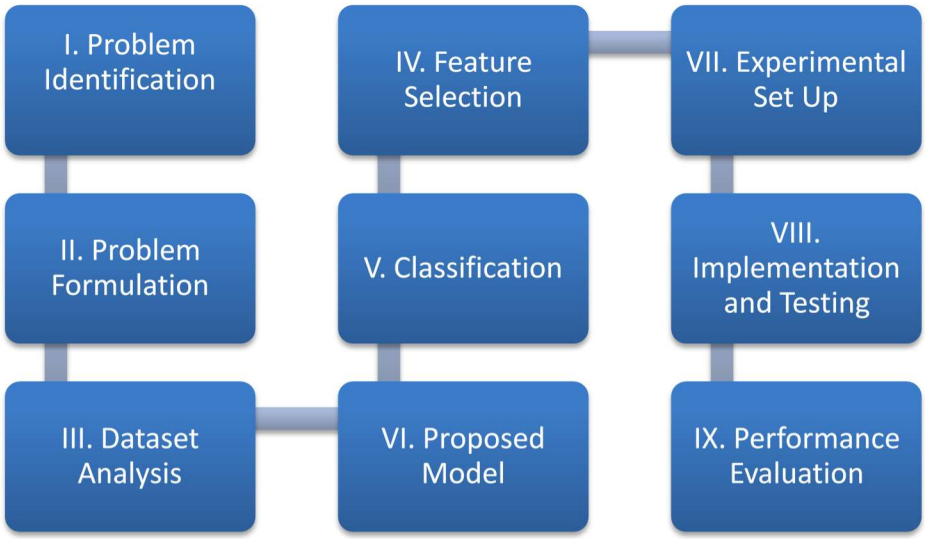

Figure 1: Research Processes

.

2.3 Dataset Analysis

The dataset consists 1000 good android applications gathered from the official website of Google Play store (https://play.google.com/store/apps?hl=en) and 1500 malicious applications downloaded from contagiomobile website (www.contagiomobile.com) and contagio minidump. Virus total (VirusTotal, 2015) online application was used to scan the benign applications to ensure they are truly good android files. Android features were extracted from the .apk executable. The features were normalized and transformed into numerical $n \times n$ dimensional vector. In the feature vectors, the binary number 1 is used to represent the presence of a feature while 0 is used to represent absence of a feature as represented above.

2.4 Feature Selection

Artificial immune systems (AIS) can be defined as computational systems that are inspired by theoretical immunology, observed immune functions, principles and mechanisms to solve problems. The immune system is a system with high complexity and is under active research (from the biological point of view), likewise the current artificial immune system (AIS) works adopted only a few immune mechanisms. Specifically, three immunological principles are primarily used in a piecemeal in AIS methods. These include the immune network theory, the mechanisms of negative selection, and the clonal selection principles.

2.5 Classification Algorithms

The classification algorithms are data mining tools used for the classification of android application features to generate new model. The classification algorithms used with artificial immune system algorithm in this research are Naïve Bayes (NB), Decision tree (J48), Neutral network (NN) and Random forest (RF). These algorithms are discussed in the following subsections.

2.6 The Proposed Model

The pseudocode of the proposed model is presented as follows:

Pseudocode

1. *Start*
2. *Load permission features*
3. *If features is normalized go to step 5 else go to step 4*
4. *Normalize features dataset*
5. *Select features with AIS*
6. *Classify dataset with Classification Algorithm*
7. *If application is classified as malicious go to step 8 else go to step 9*
8. *Report as Malicious Application*
9. *Report as Benign Application*
10. *Stop*

The corresponding flowchart is presented in Figure 2

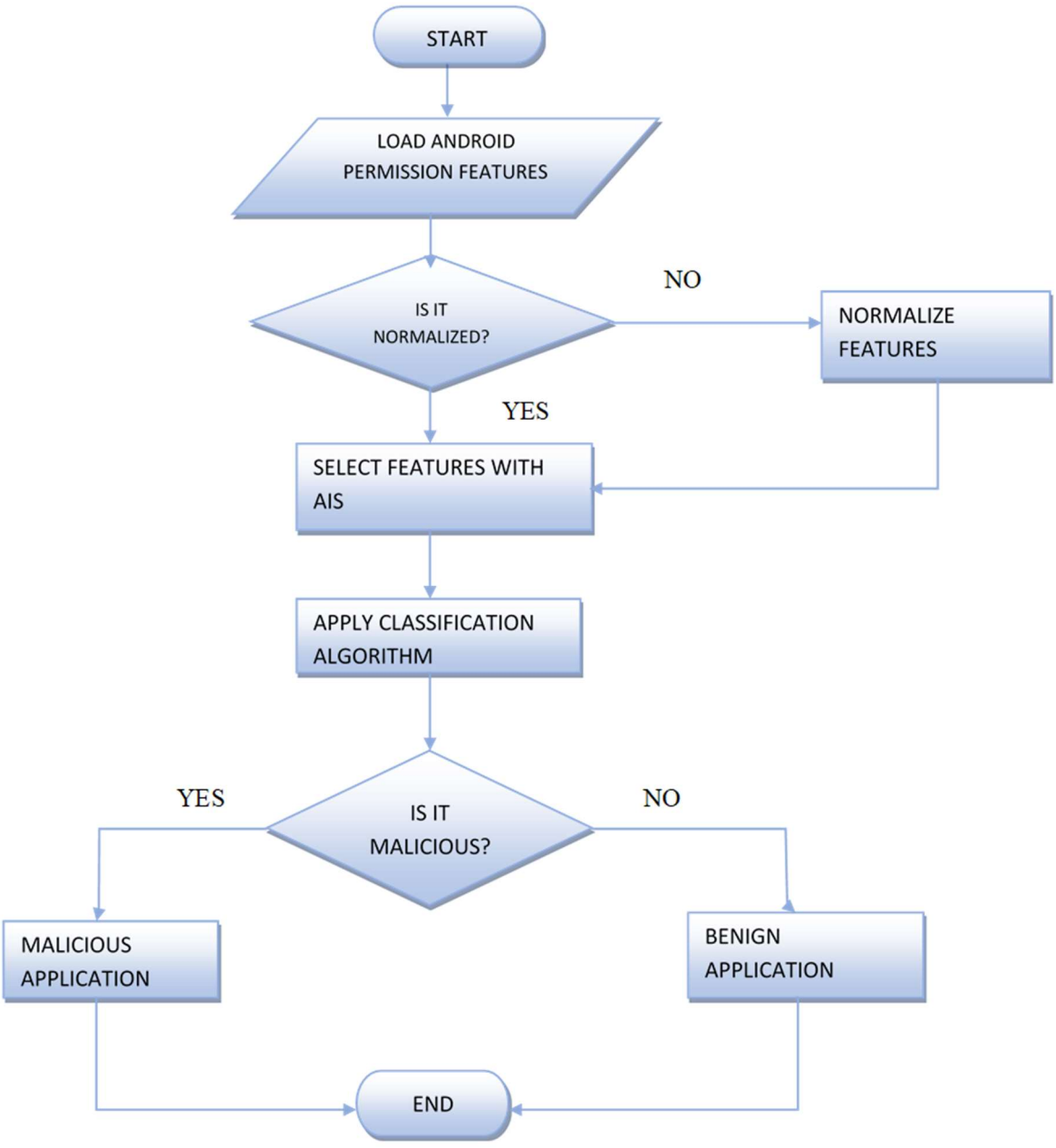

Figure 2 Flowchart of the proposed system

2.7 Experimental Setup

The MatLab R2012a will be used for feature selection while Waikato Environment for Knowledge Analysis (WEKA) toolkit will be used for machine learning applications for the classification phase. The selected features from artificial immune system algorithm will be trained using the aforementioned classification algorithms in this tool.

2.8 Design and Implementation

The proposed model involved the application of a bio inspired, optimization algorithm, artificial immune system for feature selection and the classification algorithms for classification of the android application features.

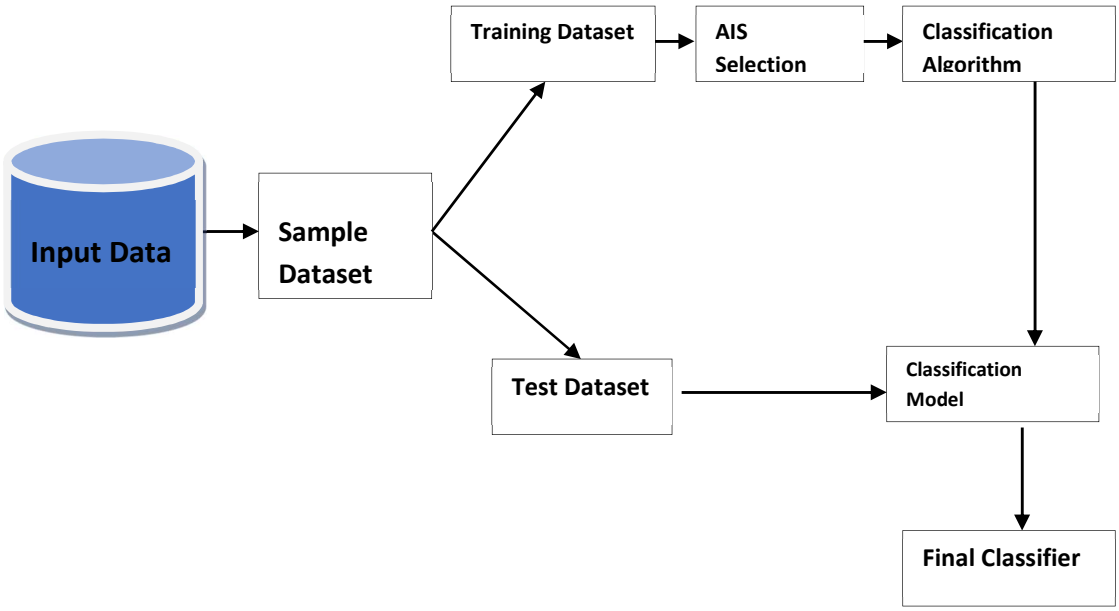

Figure 3 Research Model Data Flow

2.9 Performance Evaluation

The following performance metrics are used: Accuracy, False Positive Rate and true positive rate, while the performance of the proposed system was evaluated using McNemar' test.

Accuracy

The accuracy of an algorithm is calculated as the percentage of the dataset correctly classified by the algorithm. It looks at positives or negatives dependently and therefore other measures for performance evaluation apart from the accuracy were used.

$$A = \frac{TP+TN}{TP+TN+FP+F} * 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

where,
TP= True Positive
FP = False Positive
TN = True Negative
FN = False Negative

Positive and negative represents the classifier's prediction, true and false signify the classifier's expectation.

3. RESULTS AND DISCUSSION

3.1 Results

The results of the model obtained from classification are presented below:

Table 1: Classification without applying AIS as selection technique

Models	TPR	FPR	ACC
J48	0.714286	0.204545	0.82381
NB	0.847059	0.125	0.911905
NN	0.920354	0.052941	0.955952
RF	0.953623	0.030405	0.97483

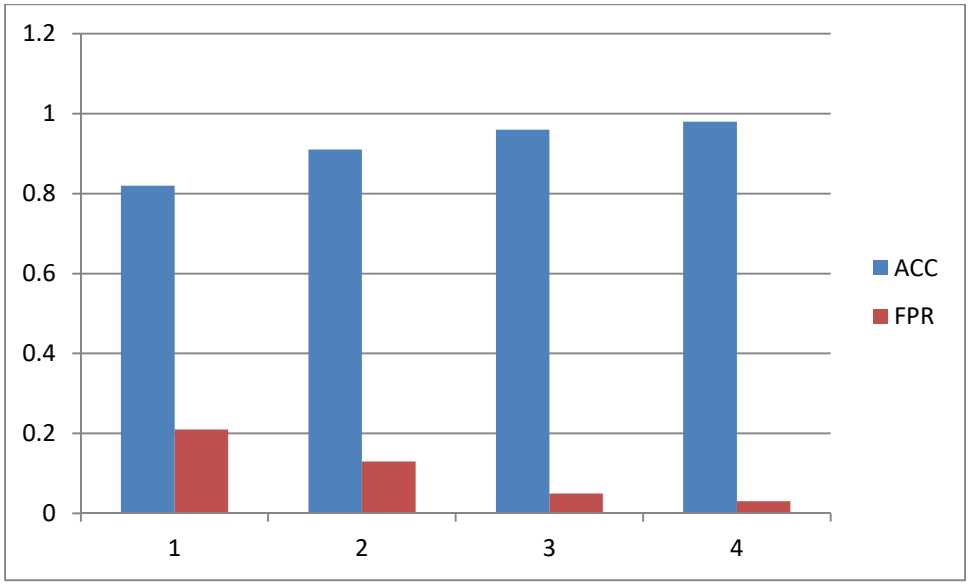

Figure 4 Features Classification without AIS

Table 2: Classification results after applying AIS as selection technique

Models	TPR	FPR	ACC (%)
J48-AIS	0.771429	0.165138	91.2
NB-AIS	0.877358	0.068807	95.6
NN-AIS	0.88	0.035047	97.8
RF-AIS	0.936759	0.022267	98.7

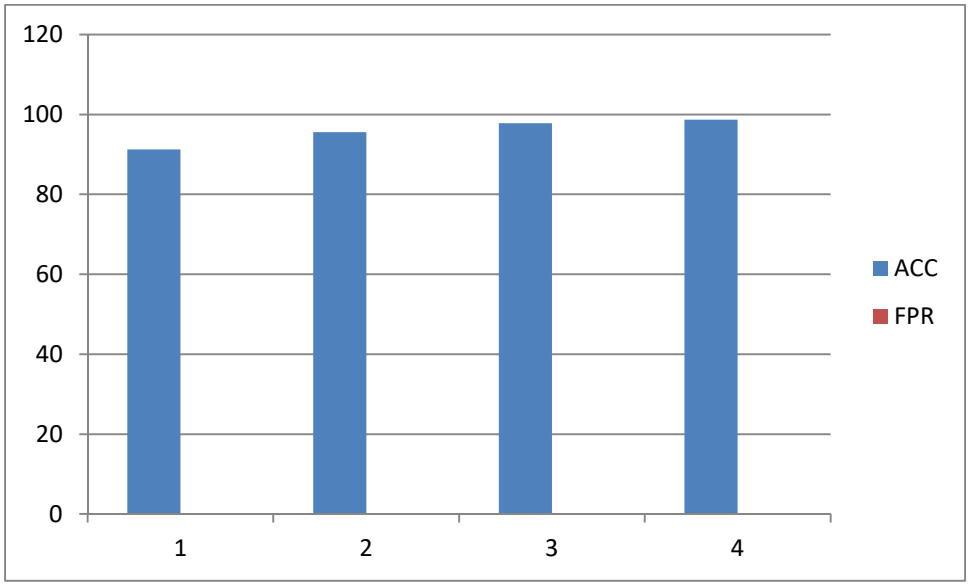

Figure 5 Features Classification using AIS

Table I shows the result of the classification model when the classification algorithms are applied on the data without any selection of the data using artificial immune system (AIS) while Table II shows the results of the models after the selection of data using artificial immune system (AIS).

3.2 Discussion

Accuracy (ACC), false positive rate (FPR) and true positive rate were measured and used to determine the effectiveness of the models. The result shows that random forest with AIS has the best results in term accuracy of 98.7%with least false positive rate of 0.22267, figure 5 clearly shows Random forest with Artificial immune system to have best accuracy and lowest false alarm rate. The result also shows that other algorithm with AIS has selection technique perform better in terms of accuracy and with least false positive rate compare to the ones without AIS. In addition, Figure 4 shows that the false positive rate of classification models without AIS is negligible, that is the false alarm rate is relatively low compared to the results in Table I. Figure 5 shows that the accuracy of classification models with AIS is relatively high compared to the results in Table 1.

4. CONCLUSION

This research proposed an android malware classification using artificial immune system for feature selection. The AIS selects the best features for the proposed classification model. This is to reduce the features redundancy and duplication. The results show that the model with AIS as selector is better than the one without selector in terms of accuracy and false alarm rate. This research is limited to only the permission features of the android applications. It is therefore, sufficient to say that the application of AIS to the selection of features for data classification usually produce better and more accurate model than the otherwise.

REFERENCES

1. Abhijit, B. H. Xin, Kang, G. S., & P. Taejoon, "Behavioral detection of Malware on Mobile Handsets," June 17–20, 2008, Breckenridge, Colorado, USA. ACM 978-1- 60558-139-2/08/06, 2008.
2. Adebayo, O. S., & Abdul Aziz, N. "Android Malware Classification Using Static Code Analysis and Apriori Algorithm Improved with Particle Swarm Optimization" 4th World Congress on Information and Communication Technologies Malacca, Malaysia December 08-10, 2014.
3. Agrawal, R., & Srikant, R., "Fast algorithms for mining association rules," In Proc. 20th int. conf. very large data bases, VLDB, Vol. 1215, pp. 487-499, September 1994.
4. Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, Yael Weiss "Andromaly": a behavioral malware detection framework for android devices**".** *Journal of Intelligent Information Systems* 38(1) (January 2011) 161{190}, 2011.
5. Adebayo, O. S., & AbdulAziz, N. (2014). Techniques For Analysing Android Malware. *International Conference on Information and Communication Technology For The Muslims World (ICT4M) 2014*, 1–6. Kuching, Sarawak, Malaysia.
6. Castro, L. N. de, & Timmis, J. I. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, *7*(8), 526–544. https://doi.org/10.1007/s00500-002-0237-z
7. Christodorescu, M., Jha, S., Seshia, S. a., Song, D., & Bryant, R. E. (2005). Semantics-Aware Malware Detection. *2005 IEEE Symposium on Security and Privacy (S&P'05)*, 32–46. https://doi.org/10.1109/SP.2005.20
8. Dasgupta, D., Ji, Z., & Gonzalez, F. (2003). Artificial immune system (AIS) research in the last five years. *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, 123–130. https://doi.org/10.1109/CEC.2003.1299565
9. Dasgupta, Dipankar, Yu, S., & Nino, F. (2011). Recent Advances in Artificial Immune Systems: Models and Applications. *Applied Soft Computing*, *11*(2), 1574–1587. https://doi.org/10.1016/j.asoc.2010.08.024
10. Garrett, S. M. (2005). How Do We Evaluate Artificial Immune Systems? *Evolutionary Computation*, *13*(2), 145–177. https://doi.org/10.1162/1063656054088512
11. Iker Burguera, Urko Zurutuza, Simin Nadjm-Tehrani Crowdroid: Behavior-Based Malware Detection System for Android. In Proceedings of the 1st ACM workshop on Security and Privacy in Smartphones and mobile devices (October 2011), Pp.15-26, 2011.
12. Milosevic, N., Dehghantanha, A., & Choo, K. R. (2017). Machine learning aided Android malware classification R. *Computers and Electrical Engineering*, *0*, 1–9. https://doi.org/10.1016/j.compeleceng.2017.02.013
13. Saracino, A., Sgandurra, D., Dini, G., & Martinelli, F. (n.d.). *MADAM : Effective and Efficient Behavior-based Android Malware Detection and Prevention*. 1–15.
14. Siddiqui, M. A. "Data Mining Methods for Malware Detection," A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Modeling and Simulation in the College of Sciences at the University of Central Florida, Orlando, Florida, 2008.
15. Suhas Holla, and Mahima M Katti "Android based Mobile Application and Its Security". International Journal of Computer Trends and Technology, 3 (3) Pp. 486-490, 2013. ISSN 2231-2801.
16. Thomas Eder, Michael Rodler, Dieter Vymazal, Markus Zeilinger "A Framework For Analyzing Android Applications". Workshop on Emerging Cyberthreats and Countermeasures ECTCM 2013.
17. VirusTotal. (2015). Free Online Virus, Malware and URL Scanner.