Ensemble Method for Mobile Malware Detection using N-Gram Sequences of System Calls

Nor Azman Mat Ariff[1], Mohd Zaki Mas'ud[1], Nazrulazhar Bahaman[1], Erman Hamid[1] and Noor Azleen Anuar[1]

[1]Information Security, Digital Forensic and Computer Networking (INSFORNET), Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.

***Abstract*:** Mobile device has become an essential tool among the community across the globe and has turned into a necessity in daily life. An extensive usage of mobile devices for everyday life tasks such as online banking, online shopping and exchanging e-mails has enable mobile devices to become data storage for users. The data stored in these mobile devices can contain sensitive and critical information to the users. Hence, making mobile devices as the prime target for cybercriminal. To date, Android based mobile devices is one of the mobile devices that are dominating the phone market. Moreover, the ease of use and open-source feature has made Android based mobile devices popular. However, the widely used Android mobile devices has encouraged malware author to write malicious application. In a short duration of time mobile malware has rapidly evolve and have the capability to bypass signature detection approach which requires a constant signature update to detect mobile malware. To overcome this drawback an anomaly detection approach can be used to mitigate this issue. Yet, using a single classifier in an anomaly detection approach will not improve the classification detection performance. Based on this reason, this research formulates an ensemble classification method of different n-gram system call sequence features to improve the accuracy of mobile malware detection. This research proposes n-number of classifier models for each different n-gram sequence call feature. The class probability output of each classifier is then combined to produce a better classification performance which is better compared to a single classifier.

***Keywords*:** Mobile Malware Detection, Ensemble method, N-gram**.**

1. Introduction

Nowadays, mobile devices are no longer confined to be used as a traditional communication tool. Mobile device has evolved to have a feature for doing online banking, acting as personal assistant, involving in social networking, having entertainment and much more. Mobile devices are becoming more similar to a personal desktop computer. Based on the International Telecommunication Union (ITU) [1] report in the end of 2018, there are 8.1 billion mobile users and 5.2 billion mobile-broadband subscribers worldwide. In the meantime, the Gartner Inc. [2] reported, 86% of the smartphone market has been monopolized by Android based mobile devices in the end of 2018. Due the popularity and the technological advancement of the Android-based mobile possessed in the market, nowadays it has become the main target of malware author. The number of mobile malwares targeting Android based platform has increased tremendously over the year. In 2012 alone, Kaspersky Security Bulletin [3] has reported that 98.96% of newly found mobile malware is targeting Android. According to [4], the current research of malware should focus on platform-specific malware such as mobile malware and IoT malware because the rise of cybercriminals in mobile phones and smart devices.

One of mobile malware detection approach is signature-based detection approach which detects mobile malware by matching up byte code pattern of the application with the database of signatures of known malicious mobile application. Signature based detection approach have an ability to detect mobile malware accurately, however signature-based approach can only detect known malware which the signature has been identified. If a new mobile malware infected a mobile device, signature-based detection approach is not able to detect the mobile malware as the signature of the new mobile malware is not yet exist. In this day, mobile malware author uses obfuscation technique to change signature of mobile malware for evading detection. The obfuscation of mobile malware in android platform is much easier to be done as android application is easily modified, repackaged and release into a new mobile application with a new set of byte pattern or signature [5]. The drawbacks of signature-based detection approach can be overcome using an anomaly approach detection where previously unknown mobile malware is able to be detected [6].

Anomaly based detection approach monitors normal activities in mobile devices and look for any irregular behavior that different from the normal pattern. Even though anomaly base detection approach is able to detect known malware, this approach still has drawbacks where the approach requires more computational resources and tend to generate some number of false alerts. Classification technique is one of the anomaly-based detection techniques and formulating an optimum classification technique can improve the mobile malware detection accuracy. According to [7] have shown that single classifier has some drawbacks, therefore is not an optimal approach to solve all problems. This limitation leads to the ensemble methods which exploit the strengths of individual classifier models by performing information fusion of classification decisions. Ensemble methods train multiple classifiers to resolve the same problem. Ensemble methods build a set of classifiers and merge the result in order to produce one optimal predictive model.

Another aspect to improve mobile malware detection accuracy is the audit data source. Audit data source contains useful traces of mobile application activity and can be processed to find mobile application behavior. The behavior of mobile application is then used in making critical decisions whether a mobile application activity is benign or malicious. Audit data source are able to reveal features of malicious activity that can be used for detection. One of the proposed solutions to enhance mobile malware detection is

to use n-gram system call sequence as a feature in classifying benign and malicious mobile applications [8]. Tracing each element of mobile malware behavior on the captured system call log has discovered that each element has its own set of system call processes and a sequence of system call invoke that can represent a malicious mobile application process. The sequential system call is known as n-gram analysis, where n is the number of the sequence length or the number of co-occurring sets of system call invoked in the process. Crowdroid [9], and Dini et. al [10] are among the previous research using system call as features in classifying benign and malicious mobile application, yet the approach only takes each single occurrence of the system call or 1-gram as the feature. Based on this, the research is considering several n values to be the classification features to improve the accuracy.

This research paper introduces an anomaly based mobile malware detection approach that applied an ensemble classification technique that uses different n-gram sequence call features. Through these n-gram sequence call features, the research built n number of classifier models where the class probability output of each classifier is then merged to produce a better prediction output that differentiate between benign or malicious mobile application. This research examines ensemble of multiple classifiers for better predictive performance compared to a single classifier. The remaining of this paper is structured as follows: Section 2 discusses the related works on mobile malware detection, Android n-gram system call sequence and ensemble classification technique; Section 3 discusses the methodology of this research, and the proposed anomaly-based detection approach using ensemble classifier with n-gram system call sequence; Section 4 describes the model evaluation, experimental settings, and results. Section 5 concludes the paper.

2. Related Works

This section describes the related work associated with mobile malware detection approach, n-gram system calls sequence and ensemble classification technique techniques.

Android malware detection techniques can be classified into signature-based detection (SB) and anomaly-based detection (AB) ([11], [12], [13]). SB detection detects malware by assessing a mobile application signature or pattern captured with known attack or threat in the signature database. Meanwhile, AB detection observed regular activities in mobile devices and look for any irregular behavior that is different from the normal pattern.

Type of Audit data source used in the detection have the ability to improve or downgrade the performance of Android malware detection. [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] and [24] are the researcher that used a static audit source features such as Permission, opcode, Intent, Network address and Strings. These audit sources are collected from Application packages (.apk) and it is easier to analyze, however, it can be hard to analyze if the packages are obfuscated and encrypted [24]. An alternative audit source data to overcome this drawback is through dynamic analysis. [25], [26], [27], [28] and [29] are the researchers that used dynamic audit data source features such as Network traffic traces, systems call, user interaction behaviors and opcode. This type of audit data source can evade the code obfuscation and encryption as the mobile application activity traces are captured during the runtime of the application. Nonetheless, monitoring dynamic mobile malware behavior can be complex as it requires a large amount of data to be processed.

Each of the mobile malware detection approaches has advantages and disadvantages; there is no one complete solution to improve mobile malware detection. SB detection has an advantage of high accuracy detection and low false alert, yet SB are not able to detect an unknown mobile malware or mobile malware that uses obfuscation technique to randomize or encrypt its application package. Whereas, AB analysis provides the detection of known and unknown mobile malware. However, AB does produce a false alert during detection. In term of audit data source, a static audit data source such as permission and API are easy to be collected but once the application package is obfuscated or encrypted the malware behavior is difficult to trace. Whereas dynamic audit data source such as system call and network traffic can reveal the true behavior of malware disregard whether a malware is obfuscated or not. Yet, the downside of dynamic audit data source is that the log captured is very large and complex. The analysis shows that to improve the accuracy of detection, a mobile malware detection requires a detection technique and audit data source that is:

1. Able to detect known and unknown malware.
2. Able to reveal the true behavior of mobile malware activity.
3. Able to reduce false alerts.

Dynamic mobile malware analysis and anomaly detection with system call as audit source data can be a good combination in detecting mobile malware analysis. This combination of approaches is able to provide a detection of known and unknown malware while revealing the true behavior of malware. A captured system call invoked by a mobile application has been able to show a mobile application behavior. Previous researcher like [9], [10] and [30] has used the frequency of system call occurrence and the system call graph as the feature in detecting mobile malware application. However, a single system call occurrence is not enough to generalize a whole set of mobile application instructions. Hence, a sequence of n system call is more relevance to present the mobile malware application instruction [31].

This research uses system call sequence as a feature in classifying benign and malicious mobile applications. The approach of n-gram system call sequence is similar to text classification where sequence of characters or words has been used to classify document, language modelling and speech recognition [32], [33] and [34]. The sequential characters or words introduced in those works is known as word n-gram analysis, where n is the length number of a sequence or the number of co-occurring sets of characters in a string. Whereas in this research, n value is representing number of a system call sequence. For example, a system call sequence for a sample process to open a file might be recorded as:

writev, access, fstat, chmod, write, fstat, close.

Thus, the frequencies of the n-gram sequence will be

analyzed as follows:

n=1: (writev)-2, (access)-1, (fstat)-2, (chmod)-1, (close)-1
n=2:(writev,acesss)-1, (access,fstat)-1, (fstat,chmod)-1, (chmod,write)-1, (write,fstat)-1, (fstat,close)-1,
n=3:(writev,acesss,fstat)-1,(access,fstat,chmod)-1,(fstat,chmod,write)-1,(chmod,write,fstat)-1,(write,fstat,close)-1

Among the advantages of n-grams is ability of the technique being used efficiently to find an appropriate matching of a sequential item. Converting any sequence of items to a set of n-grams allows the sequence of items to be efficiently compared to other sequences [35]. Applying the n-gram in mobile malware detection system, facilitate this research to find the sequence of system call that is used by mobile application during execution which might matched a sequence of malicious processes.

In order to improve the classification performance in classifying benign and malicious mobile application, this research proposed a classification technique that takes the differences of n-gram system call sequence as features and uses ensemble method approach to train multiple classifiers to solve the same problem.

Many studies ([7]; [36]; [37]) have shown that single classifier has their own domain of competence, therefore is not an optimal approach to solve all problems. This limitation leads to the increasing research in ensemble methods among machine learning community. These methods exploit the strengths of individual classifier models, obtaining enhanced performance by performing information fusion of classification decisions. Ensemble methods train multiple classifiers to solve similar problem. Ensemble methods try to construct a set of classifiers and combine them. An ensemble contains a number of classifiers called individual classifiers and constructed from a set of training data trained using classification algorithm such as neural network, support vector machine (SVM), decision tree or etc. Figure 1 shows a common ensemble method architecture.

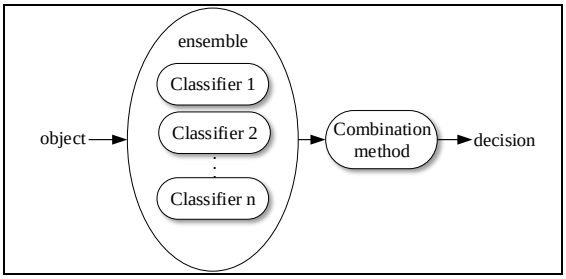

Figure 1. A common ensemble architecture

The main issue in ensemble method is how to interconnect individual classifiers. There are two structures to connect individual classifiers, namely, serial ensemble and parallel ensemble methods. Serial ensemble method combines several individual classifiers a sequential way, whereas parallel ensemble methods combine several individual classifiers in parallel. Boosting and Bagging are the examples of sequential and parallel ensemble methods respectively. Figure 2 shows how serial and parallel ensemble methods interconnect individual classifiers.

Boosting method in serial ensemble is an algorithm that is able to improve weak classifiers by exploiting the dependency factor between the individual classifiers. The main idea of boosting is to adjust the misclassified instances made by subsequent individual classifier. These misclassified instances are assigned a higher weight in the training process of the next classifier. This process is replicated until the whole set of classifiers are trained. Thus, improve the performance of each classifier which is influenced by the performance of the subsequent constructed classifier.

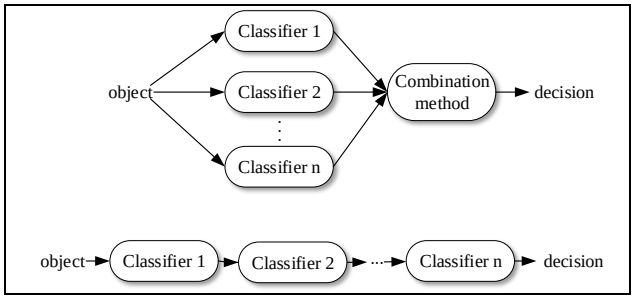

Figure 2. Parallel(top) and serial(bottom) ensemble methods

In contrast, bagging exploits the independence between the individual classifiers, since the error can be reduced dramatically by combining individual classifiers. It is a method used for sampling training dataset into a number of different subsets of the same size. Each subset is trained on a specific classifier and the classification prediction is combined using a majority voting technique. The performance of the classification can be improved significantly if the error of the single classifier is not strongly correlated. Based on this, this research considers bagging as the ensemble method to improve the accuracy.

3. Methodology

This section describes the methodology used in classifying mobile malware applications. The research methodology consists of three phases, namely, N-gram system calls sequence extraction, N-gram single classifier model generation, and ensemble of N-gram using Linear and RBF. The research methodology is illustrated in Figure 3.

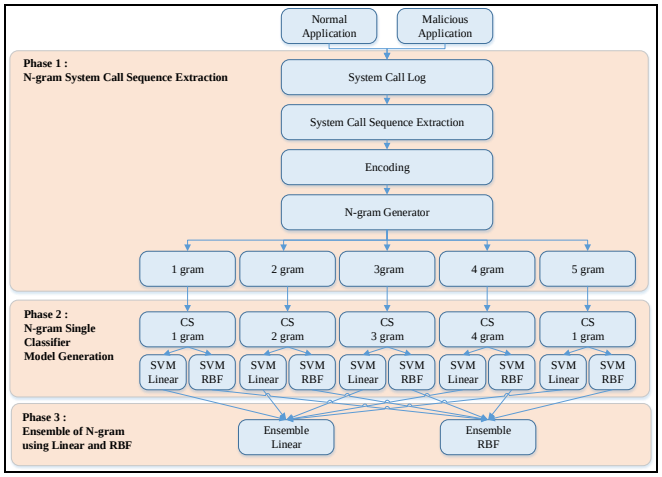

Figure 3. Research Methodology Process Flow

Phase 1 begins with the selection of android applications to generate the dataset. The studied dataset contains 480 android applications, namely 240 malware applications and 240 benign applications. The category for each application is determined by scanning the application using anti-virus software such as Bitdefender, ESET NOD32 and VirusTotal. Each application is run on the tablet for 10 minutes with interactions such as web browsing and SMS. System calls

generated by the application is extracted using a tool called strace. Every system call is uniquely represented using Unicode UTF-8 encoding. These Unicode UTF-8 characters will go through an n-gram generator which functions to extract the frequency of contiguous sequence of n Unicode UTF-8 characters, from 1-gram to 5-gram.

In phase 2, classification for single classifiers is implemented to solve the same problem, classifying malware and benign. Large dimensional feature spaces of n-gram dataset (especially 3-gram, 4-gram and 5-gram) are tend to overfit the classification model, which may lead to poor classification performance. In order to avoid overfitting, relevant feature descriptors with smaller size should be deployed. In this study, Chi-Square (CS) feature selection technique is applied to select only relevant and useful features for classification. SVM classifier with binary classification approach is used considering that the data set contains only two classes. Two SVM kernels, namely Linear and Radial Basis Function (RBF) are deployed as the SVM learning algorithms because the distribution of the dataset is not known whether it is linearly separable or non-linearly separable problems. Thus, each of n-gram dataset will be classified using these two kernels. SVM by default do not produce class probability output. However, probability calibration method can be used to convert the classification output to class probabilities.

In phase 3, combination of class probability output from multiple single classifiers for each size of n-gram sequence are used for the ensemble method. The propose combination for ensemble method is per kernel type, linear ensemble will combine class probability output from classification using linear kernel and RBF ensemble will combine class probability output from classification using RBF kernel. The combination method plays a crucial role to achieve a strong generalization ability. There are several combination methods to combine multiple single classifiers. In this study, two widely used combination methods are chosen to determine the final decision of the ensemble, namely Product Rule (PR) and Mean Rule (MR). PR is efficient when the classifiers have small errors and multiply class probability outputs of single classifiers as in formula (1). Otherwise, MR is more efficient when the classifiers contain large errors and combine class probability outputs by averaging the outputs of single classifiers as in formula (2).

$$H(x) = \frac{1}{T}\sum_{i=1}^{T} h_i(x) \quad (1)$$

$$H(x) = \frac{1}{T}\prod_{i=1}^{T} h_i(x) \quad (2)$$

4. Results and Discussion

The dataset contains 240 malware and 240 benign mobile applications. The evaluation of the proposed method is performed using 100 repeated train-test split procedure where the size of the train set is 80% (384 applications) and test set is 20% (96 applications). The accuracies are evaluated by averaging the results from all 100 runs.

Table 1 shows the average classification accuracy (%) for 1-gram to 5-gram. The optimum result is illustrated with bold characters. The highest accuracy for single classifiers n-gram is achieved by 2-gram features with accuracy 96.1 % for both kernels. In this case, we believe that 2-gram features have sufficient information to classify the mobile malware and benign. From Table 1, it shows that the classification performance started to gradually decrease after 2-gram for both kernels.

Table 1. The average classification accuracy (%) of single classifiers for n-gram features

	1-gram	2-gram	3-gram	4-gram	5-gram
Linear	91.6	**96.1**	95.5	95.2	95.2
RBF	91.7	**96.1**	96.0	95.7	95.8

The experiment was not continued for both kernels after 5-gram based on the trend where the larger the n-gram, the worse the classification performance. Moreover, producing 6-gram datasets and above requires larger memory and longer processing times. Even though many previous researchers reported that linear kernel is efficient in larger dataset, result from the experiments demonstrate that results from RBF kernel outperformed linear kernel for all n-gram datasets except 2-gram.

Table 2. The average classification accuracy (%) of the linear ensemble and RBF ensemble

	PR	MR
Linear	95.9	96.0
RBF	**96.6**	96.5

Table 2 shows the average classification accuracy (%) for ensemble method. Best result from 2-gram single classifier with 96.1% accuracy for both kernels are compared with the proposed ensemble method. The results show that the proposed RBF ensemble outperform the best single classifier. The best performance is achieved when PR combination method is applied to RBF ensemble with 96.6% accuracy. However, linear ensemble method gave a slightly lower accuracy than 2-gram single classifier. It may be due to single classifiers other than 2 grams which may not help in classification. Overall, RBF ensemble method outperformed linear ensemble for both combination methods, PR and MR. This performance maybe contributed by the combination of strong single classifiers using RBF kernel.

5. Conclusion

This research proposed an anomaly technique for mobile malware detection approach through an ensemble of n-gram system call sequence. The classification result shows that bagging ensemble classifier which uses product rule combination method gives the highest accuracy in classifying between malicious and benign mobile application. Thus, proving that a combination of multiple classifiers is better than considering only single classifier.

6. Acknowledgement

The authors are grateful to InforsNet Research Group, Fakulti Teknologi Maklumat dan Komunikasi(FTMK), Universiti Teknikal Malaysia Melaka for the support and special acknowledgement to the Ministry of Higher Education Malaysia (MoHE) for providing financial support through the Fundamental Research Grant Scheme (FRGS/2018/FTMK-CACT/F00391).

References

[1] International Telecommunications Union (ITU), 2019. ICT Facts Figures 2019. [ONLINE] Available at http://www.itu.int [Accessed 2 February 2019]

[2] Forni A. A. & Meulen R. V. D., 2017. Gartner Says Worldwide Sales of Smartphones Grew 9 Percent in First Quarter of 2017. [ONLINE] Available at http://www.gartner.com/newsroom/id/3725117. [Accessed 30 May 2019]

[3] Denis Maslennikov and Yury Namestnikov, Kaspersky Security Bulletin 2012. The overall statistics for 2012, (Securelist), [ONLINE] http://www.securelist.com/en/analysis/204792255/Kaspersky_Security_Bulletin_2012_The_overall_statistics_for_2012#1. [Accessed 25 July 2019]

[4] M. N. Alenezi, H. Alabdulrazzaq, A. A. Alshaher, M. M. Alkharang, "Evolution of Malware Threats and Techniques: A Review," International Journal of Communication Networks and Information Security, vol. 12, no. 3, pp. 326–337, 2020.

[5] K. Tian, D. Yao, B. G. Ryder, G. Tan and G. Peng, "Detection of Repackaged Android Malware with Code-Heterogeneity Features," in IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 1, pp. 64-77, 2020.

[6] S. S. Moon, and J. J. Kyeong, "Alert Correlation Analysis in Intrusion Detection," Proceedings of the 2nd International Conference Advanced Data Mining and Applications (ADMA 2006), pp. 1049–1056, 2006.

[7] A. Abdullah, R. C. Veltkamp, M. A. Wiering, "Ensembles of novel visual keywords descriptors for image categorization," 2010 11th International Conference on Control Automation Robotics and Vision (ICARCV), pp. 1206-1211, 2010.

[8] M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat, R. Yusof and R. Ahmad, "Profiling mobile malware behaviour through hybrid malware analysis approach," 2013 9th International Conference on Information Assurance and Security (IAS), pp. 78-84, 2013.

[9] I. Burguera, U. Zurutuza, and S. 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, pp. 15-26, 2011.

[10] G. Dini, , F. Martinelli, A. Saracino, and D. Sgandurra, "Madam: a multi-level anomaly detector for android malware," In Computer Network Security, pp. 240-253, 2012.

[11] G. Canfora, E. Medvet, F. Mercaldo, and C. A. Visaggio, "Acquiring and Analyzing App Metrics for Effective Mobile Malware Detection," In Proceedings of the 2016 ACM on International Workshop on Security and Privacy Analytics - IWSPA '16, pp. 50–57, 2016.

[12] K. Sokolova, C. Perez, and M. Lemercier, "Android application classification and anomaly detection with graph-based permission patterns," Decision Support Systems, 93, pp. 62–76. 2017.

[13] Robiah Y, Siti Rahayu S., Mohd Zaki M, Shahrin S., Faizal M. A., Marliza R., "A New Generic Taxonomy on Hybrid Malware Detection Technique," International Journal of Computer Science and Information Security, IJCSIS, Vol. 5, No. 1, pp. 56-61, 2009.

[14] M. F. A. Razak, N. B. Anuar, R. Salleh, A. Firdaus, M. Faiz, and H. S. Alamri, "'Less Give More': Evaluate and zoning Android applications," Measurement : Journal of the International Measurement Confederation, vol. 133, pp. 396–411, 2019.

[15] Y. S. Yen, and H. M. Sun, "An android mutation malware detection based on deep learning using visualization of importance from codes," Microelectronics Reliability, vol. 93, pp. 109-114, 2019.

[16] L. Zhang, V. L. Thing, and Y. Cheng, "A scalable and extensible framework for android malware detection and family attribution," Computers & Security, vol. 80, pp.120-133, 2019.

[17] A. Kumar, K. S. Kuppusamy, and G. Aghila, "FAMOUS: Forensic Analysis of MObile devices Using Scoring of application permissions," Future Generation Computer Systems, vol. 83, pp.158-172, 2018.

[18] S. Sheen, R. Anitha, and V. Natarajan, "Android based malware detection using a multifeature collaborative decision fusion approach," Neurocomputing, vol. 151, pp.905-912, 2015.

[19] F. Idrees, M. Rajarajan, M. Conti, T.M. Chen, and Y. Rahulamathavan, "PIndroid: A novel Android malware detection system using ensemble learning methods," Computers & Security, vol. 68, pp.36-46, 2017.

[20] L. Zhao, D. Li, G. Zheng and W. Shi, "Deep Neural Network Based on Android Mobile Malware Detection System Using Opcode Sequences," In 2018 IEEE 18th International Conference on Communication Technology (ICCT), pp. 1141-1147, 2018.

[21] P. Zhang, S. Cheng, S. Lou and F. Jiang, "A Novel Android Malware Detection Approach Using Operand Sequences," 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), pp. 1-5, 2018.

[22] R. Raphael, P. Vinod and B. Omman, "X-ANOVA and X-Utest features for Android malware analysis," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1643-1649, 2014.

[23] S. Y. Yerima, S. Sezer, G. McWilliams and I. Muttik, "A New Android Malware Detection Approach Using Bayesian Classification," 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp. 121-128, 2013.

[24] S. Feldman, D. Stadther and B. Wang, "Manilyzer: Automated Android Malware Detection through Manifest Analysis," 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, pp. 767-772, 2014.

[25] S. Wang, , Z. Chen, Q. Yan, B. Yang, L. Peng, and Z. Jia, "A mobile malware detection method using behavior features in network traffic," Journal of Network and Computer Applications, vol. 133, pp.15-25, 2019.

[26] P. Vinod, A. Zemmari, and M. Conti, “A machine learning based approach to detect malicious android apps using discriminant system calls,” Future Generation Computer Systems, vol. 94, pp. 333–350, 2019.

[27] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang and B. Yang, “Machine learning based mobile malware detection using highly imbalanced network traffic,” Information Sciences, Vol. 433–434, pp. 346–364, 2018.

[28] F. Tong and Z. Yan, “A hybrid approach of mobile malware detection in Android,” Journal of Parallel and Distributed computing, Vol. 103, pp. 22–31, 2017.

[29] N. A. Anuar, M. Z. Mas’ud, N. Bahaman, and N. A. Mat Ariff, “Mobile Malware Behavior through Opcode Analysis,” International Journal of Communication Networks and Information Security, Vol. 12, No. 3, pp. 345–354, 2020.

[30] K. J. Abela, D. K. Angeles, J. R. D. Alas, R. J. Tolentino, and M. A. Gomez, “An Automated Malware Detection System for Android using Behavior-based Analysis AMDA,” International Journal of Cyber-Security and Digital Forensics (IJCSDF), vol. 2, no. 2, pp. 1–11, 2013.

[31] M. Z. Mas’ud, S. Sahib, M. F. Abdollah, S. R. Selamat, and R. Yusof, “An evaluation of n-gram system call sequence in mobile malware detection,” ARPN Journal of Engineering and Applied Science, Vol. 11, No. 5, pp.3122-3126, 2016.

[32] W. B. Cavnar, and J. M. Trenkle, “n-gram-Based Text Categorization,” in Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175, 1994.

[33] A. Jacob, and M. Gokhale, “Language Classification using n-grams Accelerated by fpga-Based Bloom Filters,” in Proceedings of the 1st international workshop on High-performance reconfigurable computing technology and applications: held in conjunction with SC07, pp. 31-37, 2007.

[34] D. Jurafsky, and J. H. Martin, *Speech and Language Processing (Vol. 3).* London, Pearson, 2014.

[35] M. A. Abdulhayoglu, B. Thijs, and W. Jeuris, “Using Character N-Grams to Match a List of Publications to References in Bibliographic Databases,” Scientometrics, 109(3), pp. 1525-1546, 2016.

[36] S. Kumar, A. Viinikainen and T. Hamalainen, "Evaluation of ensemble machine learning methods in mobile threat detection," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 261-268, 2017.

[37] A. D. Donald and G. Murali, “Selective ensemble of Internet traffic classifiers for improving malware detection,” 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 3548-3551, 2017.