Enhanced Android Malware Detection and Family Classification, using Conversation-level Network Traffic Features

Mohammad Abuthawabeh and *Khaled Mahmoud
King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Jordan
*k.mahmoud@psut.edu.jo

Abstract: *Signature-based malware detection algorithms are facing challenges to cope with the massive number of threats in the Android environment. In this paper, conversation-level network traffic features are extracted and used in a supervised-based model. This model was used to enhance the process of Android malware detection, categorization, and family classification. The model employs the ensemble learning technique in order to select the most useful features among the extracted features. A real-world dataset called CICAndMal2017 was used in this paper. The results show that Extra-trees classifier had achieved the highest weighted accuracy percentage among the other classifiers by 87.75%, 79.97%, and 66.71%for malware detection, malware categorization, and malware family classification respectively. A comparison with another study that uses the same dataset was made. This study has achieved a significant enhancement in malware family classification and malware categorization. For malware family classification, the enhancement was 39.71% for precision and 41.09% for recall. The rate of enhancement for the Android malware categorization was 30.2% and 31.14% for precision and recall, respectively.*

Keywords: *Information Security, Android Malware, Network Traffic Analysis, Conversation-level Features, and Machine Learning.*

Received February 19, 2020; accepted June 9, 2020
https://doi.org/10.34028/iajit/17/4A/4

1. Introduction

Nowadays, smartphones are not only for making phone calls as it was before. It is now a tool for holding personal information, health care, payment, and more e-services. As a result, the number of smartphone users in 2019 has increased by 5.9% more than in 2018 [30]. According to a report prepared by International Data Corporation (IDC) [17], the Android operating system is the most popular operating system for smartphones in 2019. It has an 86.7% market share more than any other smartphone's operating system. Android is a Linux based open-source operating system developed by Google [23]. It was invented in 2003, whereas the first Android smartphone was invented in 2008 [28].

"Google Play" is an official market store offered by Google [13].Google Play offers more than two and a half million applications [29]. However, this store is not the only source of Android applications; many other unofficial third-party application developers exist. Along with such a massive number of applications, the number of potential security and privacy issues by malware is increased [18, 26].

In order to reduce the risks of malware and other malicious applications, Google released a machine learning ecosystem under the name of "Play Protect" [12]. It is designed to detect malware before and after uploading applications to the market. In spite of this control process, more than 132 thousand malware was detected in the first quarter of 2018 [8], two million Android users were infected by "False-Guide" botnet in 2016 [25] as well as, half-million users were infected by thirteen different malware from applications that were uploaded to Google play market [35]. Unfortunately, Android smartphones still a target for cybercriminals.

The risks of malware are growing, and so are the efforts to mitigate their risks. In this respect, security researchers employ two methods to detect malware: the static-based method that aims to analyze the malware without running it and the dynamic-based method that monitors the malware behavior inside an isolated environment (i.e., monitoring the generated traffic of malware) [5]. Both methods can be used by machine learning to enhance malware detection.

The contribution of this study is

a) Distinguishing the most effective network traffic features.
b) Determining the best machine learning algorithm (out of three classifiers) for detecting, categorizing and classifying of malware.
c) Providing a comparison between this study and related studies that used the same dataset.
d) Enhancing the used dataset "CICAndMal2017".

This paper is an extension of [1]. In the previous conference paper, a model was designed to detect the existence of Android malware as well as categorize them. In this extension, the authors apply the same model to do malware family classification besides detection and malware categorization. The model was tested and validated using the same dataset.

This remainder of this paper is organized as follows. Section 2 introduces the dataset that is used in this study known as CICAndMal2017. Section 3explores the results of some recent related works. Section 4describes the proposed model. Section 5 presents the experimental environment and shows the results. Section 6 compares the results of this study with another study using the same dataset. Finally, section 7 concludes this research.

2. CICANDMAL2017 Dataset

After reviewing the most comprehensive and coherent set of related publications, we found that the Canadian Institute for Cybersecurity (CIC) [19] provides a competent real-world dataset called CICAndMal2017. CIC build their dataset in a way that reduces the drawbacks and shortcomings of the earlier dataset.

As a starting point, the CIC collected more than four thousand malware applications from different resources, such as VirusTotal [34] and Contagiodumpst [24]. Moreover, more than six thousand benign applications published during 2015, 2016, and 2017 and uploaded to Google play market were collected. However, CIC has only managed to install 5 thousand of them (malware 429 and benign 5,065) on real android smartphones to conduct a real-world environment. Finally, CIC connected Android smartphones to a hotspot computer to capture network traffic in Capture Packet (PCAP) format using TCPDUMP software [32]. The CIC offered the dataset in two formats: CSV files generated by CIC Flow meter (2126 CSV file) and PCAP files (more than 20 gigabytes of captured network traffic). In this paper, the PCAP files are used.

To deal with some advanced malware that uses the time delay technique to escape dynamic analysis, the CIC captured the network traffic in three different times: After malware installation directly, fifteen minutes before rebooting and fifteen minutes after rebooting.

CICAndMal2017 has multiple levels of labeling. At the first level, the PCAP files are grouped into two categories: benign or malware. In the second level, malware types are categorized into four categories:

- Adware: Adware automatically displays advertising materials and aims to collect the highest number of clicks or views on unwanted advertisement banners [2].
- Ransomware: A malicious application aims to block access over computer resources. For example, it can encryptusers'files to extort them to pay money for decrypting their files or unlocking their devices [21].
- Scareware: This type of malware tries to scare users to let them purchase unnecessary and potentially dangerous software applications [14].
- SMS malware: A malware that makes unauthorized calls or/and sends SMS messages without user consent. The malware owner can operate the infected devices as a premium channel for SMS services [16].

Note that these different malware categories can do other malicious activities. For example, they can steal users' financial information and sending them to command and control servers [11, 33].

At the third level, each malware that was categorized in one of the above four categories is classified into its family type. Some of these types associated with its malware category are shown in Table 1.

Table 1. Malware category and family types.

Category	**Family Type**		
Adware	Ewind	koodous	Kemoge
	Dowgin	Mobidash	Youmi
	Feiwo	Selfmite	Shuanet
	Gooligan		
Ransomware	Charger	Pletor	LockerPin
	Jisut	PornDroid	Svpeng
	Koler	RansomBO	WannaLocker
	Simplocker		
Scareware	AndroidDefender	FakeAV	FakeApp.AL
	AndroidSpy.277	FakeJobOffer	FakeAV
	AV for Android	FakeTaoBao	FakeApp
	AVpass	Penetho	VirusShield
SMS Malware	BeanBot	Jifake	FakeNotify
	Biige	Mazarbot	SMSsniffer
	FakeInst	Nandrobox	FakeMart
	Plankton		

3. Related Work

In [15], a new method was introduced to detect android malware using edge computing and traffic clustering. First, the authors sent the android devices traffic to the edge server. Second, the edge server extracts mobile traffic content features (i.e., extracted plaintext from HTTP flows) and traffic behavior (i.e., packet intervals) and sent them to the cloud platform. Finally, they calculated the similarities between applications and clusters to detect the malware automatically. They used similarity methods: TF-IDF algorithm and cosine similarity. For evaluating their method, they used 400 android application. Note that the data set was not published online because of privacy concerns. The final average accuracy for their model was 96.9%.

In [7], the author used the Long Short Term Memory (LSTM) based deep learning framework on detecting malware of type ransom ware. Two categories of CICAndMal2017 dataset were selected in this study: benign and ransomware. Furthermore, they selected the top 19 flow-level features using 8 feature selection algorithms such as Chi-Square and information gain. The accuracy, recall, and F1-score results of their model were 97%.

In [9] a part of the CICAndMal2017 dataset was used; the researchers choose one PCAP file for each malware family. Their chosen samples were taken randomly. Features were extracted from PCAP files using two steps. The first step: a Java program was developed to separate network flows using the flow-level technique. Then, fifteen features were extracted, using a python program. One of the features was the minimum size of the sent packet within a flow. Three supervised machine-learning classifiers were used. The classifiers were K-Nearest Neighbours, Random forest and Decision Tree. They classify instances into two categories: malware and benign. Then, classifying malware instances into three categories: Adware, Ransom ware, and Scareware. The authors use three measures: recall, precision, and F-score.

For malware-benign classification, the results show that the Random Forest classifier has obtained the highest results by 92% for F-score and 95% for precision as well as recall. The other classifiers gained more than 85% of all used measures. Formal ware classification, the selected classifiers achieved more than 80% for the chosen measures. Similar to malware-benign classification, the Random Forest classifier gained the highest results by 84% for recall, precision and F-score. The researchers did not show the results if the full dataset was used.

In [19], CICAndMal2017 dataset was used. CIC researchers extracted network traffic flow-level features. Two algorithms for feature selection were used: Information Gain (IG) and Correlation-based Feature Selection (CFS). The two algorithms select nine features. Three machine learning classifiers were used to evaluate their model, namely: Decision Tree, Random Forests, and K-Nearest (KNN). The classifiers categorized malware in three scenarios: malware binary detection, malware category classification and malware families' characterization.

The results show that network traffic flow-level features are useful for binary detection, but not for the other scenarios. For clarification, the three classifiers gained 85% precisionon average and 88% for recall measure for binary detection. On the other hand, malware category classification achieved less than 50% for precision and recall, and less than 20% for precision and recall for the family classification.

In [38], the Decision Tree (J48) algorithm was used to detect malware traffic. They used 700 samples; 200 samples are malware from Drebin [6] and Contagiodumpst datasets, and 500 benign samples from Google play market. Network traffic of these samples was captured on a real smartphone using "tpacketcapture" [31]. The authors extracted seven features from the captured traffic. Finally, they calculated the accuracy for Drebin and Contagiodumpst datasets, and the results were 98.4% and 97.6%, respectively.

In [20], a new model was proposed to detect and categorize Android malware based on network traffic features. The authors collected the generated network traffic of 1500 benign applications as well as 400 general malware and adware. Next, they used feature selection algorithms such as IG and CFS to select the most useful features. Finally, supervised machine learning classifiers were used to detect and categorize malware. The proposed model achieved more than 90% average accuracy and precision.

In [4], network traffic features of Android malware are prioritized based on IG and Chi-Square tests. Next, network traffic features were minimized using a proposed algorithm to enhance the detection accuracy and reduce the time for training and testing phases. Statistical analysis techniques were used to rank features. The proposed algorithm finds that 9 out of 22 features are adequate for higher detection accuracy. Likewise, the study results show that it can reduce the time for training and testing phases 50% and 30%, respectively.

In [27], the researchers proposed a technique to detect Android malware based on network traffic using Decision Tree classifier. First, they selected 100 malware samples that generate network traffic from the Genome project dataset [37]. Second, they captured malware and benign applications network traffic using a real smartphone by "tpacketcapture" application. Third, the authors analyzed sixteen features and picked eight among them. Finally, Decision Tree was used to classify network. The accuracy of this proposed technique was more than 90%.

4. Proposed Model

In this section, an enhanced model based on the conversation-level feature is presented. The objective of this model is to increase the accuracy for mainly three scenarios: malware binary detection, malware categorization and malware family classification.

The proposed model is a sequence of four phases; analysis and feature extraction, data cleaning, feature extraction and finally training and testing. Figure 1 shows these phases. A detail description of each phase in this model is given below.

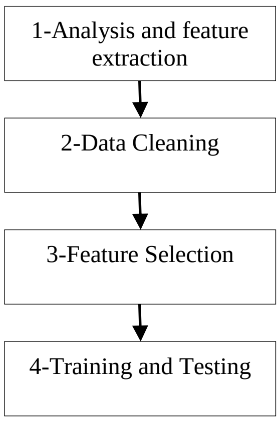

Figure 1. The four phases of the proposed model.

- *Phase* 1: Analysis and Feature Extraction

In this phase, the CICAndMal2017 dataset is analyzed in order to extract the most useful conversation-level features. The network traffic can be viewed as a set of conversations. Each conversation is represented as a statistical summary of network flows with the same two-tuples (source IP and destination IP) [22].The idea behind the conversation-level approach is to capture who is talking to whom. It is widely used in peer-to-peer applications and botnet detection. An essential advantage of this approach is that it can detect malware that uses port randomizes technique.

- *Phase* 2: Data Cleaning

This phase aims to clean the dataset from any unuseful instances and features. Some instances may affect the detection accuracy negatively, such as instances with Google DNS IP address or internal routing network traffic. Therefore all internal routing and Google DNS instances are removed. Moreover, all flow identification features will be removed, such as source IP and Destination IP.

Next, the cleaned dataset will be divided into two sub-datasets:a training set and a testing set (80% - 20% respectively).

- *Phase* 3: Feature Selection

In this phase, the most useful features will be selected. One way to do that is through the use of the ensemble learning technique. This technique uses multiple machine learning algorithms for the same task, which can lead to better accuracy [36]. In this model, three feature selection algorithms were used; Random Forest (RF), Recursive Feature Elimination (RFE) and Light GBM. The selected features are those ones that receive more votes (i.e., each feature that is selected by at least two algorithms will be considered in the next steps).

Figure 2 shows the steps of this phase. Note that these steps will be done for each scenario separately. Figure 3 explains the voting process for a single feature.

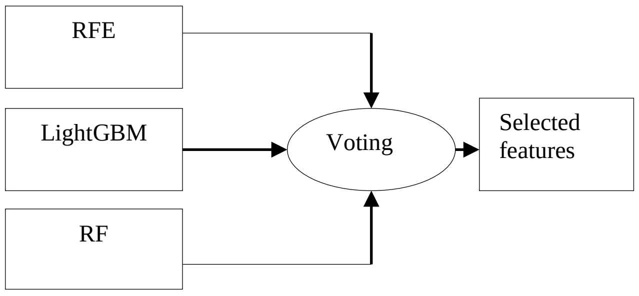

Figure 2. Feature selection steps.

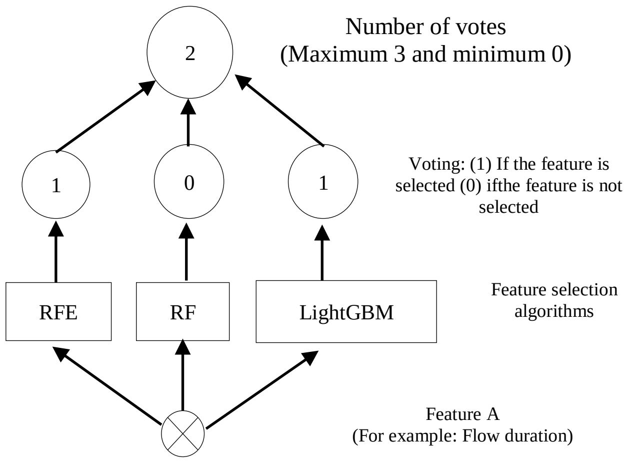
2
Number of votes (Maximum 3 and minimum 0)
1
0
1
Voting: (1) If the feature is selected (0) ifthe feature is not selected
RFE
RF
LightGBM
Feature selection algorithms
Feature A (For example: Flow duration)

Figure 3. Feature selection example.

- *Phase* 4: Training and Testing

Extra Trees, Random Forest, and Decision Tree are trained using the cleaned dataset that was prepared in the previous phase. In like manner, these classifiers were evaluated with the testing set. Three selected measures have been calculated for each classifier. These measures are weighted accuracy, weighted precision, as well as weighted recall.

5. Experiments Details and Results

All experiments have been conducted on the Microsoft Windows 10 Professional (64-bit) version with a second-generation 2.20 GHz Intel Core i7 processor and 16 GB of memory. Python 3.7.0 was chosen for data pre-processing, feature selection, and model building because of its productive and useful libraries for such tasks.

One tool that uses the conversation-level technique is the PeerShark tool [22]. The available version of this tool extracts six features only. Since it is an open-source tool, it can be enhanced to adopt new features. Therefore, a new fourteen conversation-level features were developed (E1-E14). Table 2 list these features.

After executing the cleaning phase, the number of removed instances was 798 instances for the first scenario and 456 instances for the second and the third scenario. Only two identification features were removed: source IP and destination IP. Table 3 shows the number of instances grouped by scenario number.

In the feature selection phase and after executing the ensemble learning technique, nine features were selected. Table 4 lists these selected features for the three scenarios.

Table 2. Peershark basic and extended features.

#	Feature name	Description
1	SourceIP	The source IP of the conversation
2	DestinationIP	The destination IP Source
3	NoOfPackets	Number of packets during the conversation
4	NoOfBytes	Number of bytes during the conversation
5	InterArrivaltime	Median of inter-arrival time of packets
6	DurationInSeconds	Duration time of a connection in seconds
E1	NoOfPacketSizeFWD	Number of byte per forward packet
E2	NoOfPacketSizeBWD	Number of byte per backward packet
E3	PacketPerSecFWD	The forward packet per second
E4	PacketPerSecBWD	The backward packet per second
E5	PKTFwdnum	Total number of forward packets
E6	PKTBwdnum	Total number of backward packets
E7	ByteFwdnum	Total number of forward bytes
E8	ByteBwdnum	Total number of backward bytes
E9	BytePerFlow	Number of byte per flow within the conversation
E10	PacketPerFlow	Number of packets per flow within the conversation
E11	FWDBytePerFlow	Number of forward bytes per flow within the conversation
E12	BWDBytePerFlow	Number of backward bytes per flow within the conversation
E13	FWDPacketPerFlow	Number of forward packets per flow within the conversation
E14	BWDPacketPerFlow	Number of backward packets per flow within the conversation

Table 3. Number of instances for training and testing dataset.

	Set Name	Number of instances
First Scenario	Training set	244,594
	Testing set	61,149
Second Scenario	Training set	121,084
	Testing set	30,271
Third Scenario	Training set	121,084
	Testing set	30,271

Table 4. Selected features for all scenarios.

#	First Scenario Selected Features	Second Scenario Selected Features	Third Scenario Selected Features
1	DurationInSeconds	NoOfBytes	InterArrivaltimet
2	NoOfPacketSizeFWD	DurationInSeconds	DurationInSeconds
3	InterArrivaltimet	BytePerFlow	PacketPerSecFWD
4	FWDBytePerFlow	BWDBytePerFlow	PacketPerSecBWD
5	PacketPerSecFWD	PacketPerSecFWD	PacketPerFlow
6	PacketPerSecBWD	PacketPerSecBWD	NoOfPackets
7	PacketPerFlow	NoOfPacketSizeFWD	NoOfBytes
8	NoOfBytes	NoOfPacketSizeBWD	FWDPacketPerFlow
9	FWDPacketPerFlow	NoOfBytes	NoOfPacketSizeFWD

In the last phase, the three classifiers were trained and tested. The selected measures have been calculated for each classifier. Tables 5, 6, and 7 show the calculated measures for the first scenario, the second scenario, and the third scenario.

The listed results show that the Extra-Trees classifier achieved the highest accuracy, precision, and recall, among other classifiers for the three scenarios. On the other hand, the decision tree classifier achieved the lowest results.

Table 5. First scenario training and testing phase results.

#	Name	Weighted Accuracy	Weighted Precision	Weighted Recall
1	Extra Trees	87.75%	89.35%	85.33%
2	Random Forest	86.65%	89.00%	83.22%
3	Decision Tree	86.12%	85.76%	86.16%

Table 6. Second scenario training and testing phase results.

#	Name	Weighted Accuracy	Weighted Precision	Weighted Recall
1	Extra Trees	79.97%	80.24%	79.73%
2	Random Forest	79.91%	80.20%	79.64%
3	Decision Tree	77.13%	77.07%	77.06%

Table 7. Third scenario training and testing phase results.

#	Name	Weighted Accuracy	Weighted Precision	Weighted Recall
1	Extra Trees	66.71%	67.26%	66.85%
2	Random Forest	66.64%	67.21%	66.59%
3	Decision Tree	65.44%	65.20%	65.70%

6. Results Comparison

One main goal of this research is to enhance the detection accuracy achieved by other related studies that use the same dataset. Therefore, the achieved results in this research are compared with the results published in [19]. This published study is the only one provided by the Canadian Institute for Cybersecurity (CIC) and uses the same and full dataset.

First of all, the CIC study employs a different feature extraction technique. CIC extractsflow-level features [3] using the CICFlowMeter-V3 [10], whereas our study extracts the conversation-level features using PeerShark. A flow is considered as a statistical summary of packets with the same five features (source IP, destination IP, source port, destination port, and protocol).

To compare the results of both studies, we rely on the results that were achieved using common classifiers and measures. The selected classifiers are random forest and decision tree, while the selected measures are weighted recall and weighted precision.

Table 8 shows the results obtained in each study for each scenario after calculating the two measures. Figures 4, 5, and 6 visualize the results shown in Table 8. These results can be illustrated as follows:

- The first scenario (Binary detection): Figure 4 reveals that this study result is higher than the CIC study by (0.85%) precision and (0.7%) recall for the random forest algorithm. Also, the decision tree algorithm achieved better precision in this model by (0.66%) but less recall (1.84%).
- The second scenario (Malware categorization): Figure 5 confirms that this study achieved more significant results than the CIC for precision and recall by around (30%). This enhancement proves

that conversation-level features are more suitable for malware categorization than flow-level features.

- The third scenario (Family classification): this model achieved higher results than the CIC model. Figure 6 expresses the enhancement gained in precision and recall by (39.71%) and (41.09%) respectively for Random forest algorithm. Also, it shows the improvement for decision tree algorithm by (38.54%) precision and (45.1%) recall.

Table 8. Training and testing phase results comparison.

Scenario	**Model**	**Random Forest**		**Decision Tree**	
		PR[a] %	**RC[b] %**	**PR[a] %**	**RC[b] %**
First scenario	New Model	86.65	89	85.76	86.16
	CIC	85.8	88.3	85.1	88
Second scenario	New Model	80.20	79.64	77.07	77.06
	CIC	49.9	48.5	47.8	45.9
Third scenario	New Model	67.21	66.59	65.20	65.70
	CIC	27.50	25.50	26.66	20.06
		[a] weighted precision		[b] weighted recall	

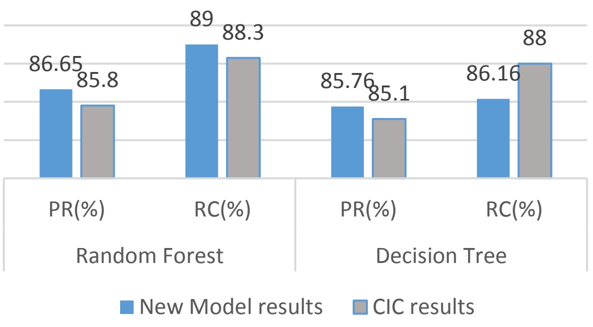

Figure 4. Binary detection comparison for both studies.

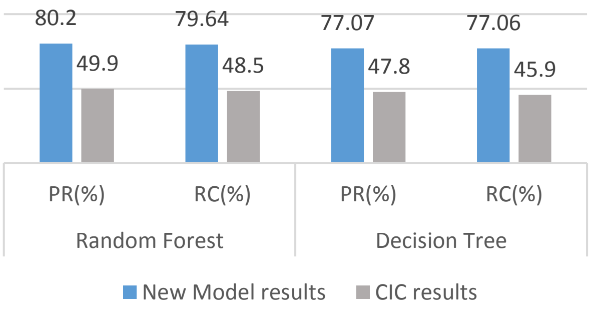

Figure 5. Malware categorization comparison for both studies.

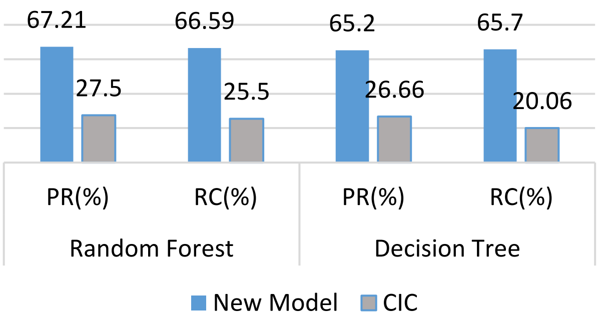

Figure 6. Malware family classification comparison for both studies.

7. Conclusions

This research introduces an enhanced model for malware detection, categorization, and family classification in the android environment. The model extracts conversation-level network traffic features from a recent and real-world dataset named "CICAndMal2017. For the process of the feature extraction phase, conversion-level features were extracted using the PeerShark tool. Multiple stages of data pre-processing have been conducted to the dataset. The most useful features were selected using the ensemble learning technique by three feature selection algorithms: Random Forest, RFE, and LightGBM classifiers. Moreover, the developed model was trained and tested using three classifiers: Decision Tree, Random Forest, and Extra-trees. Finally, this study compared the provided model results with another model that used the same dataset.

According to the final results, conversation-based features can enhance the detection, categorization, and family classification of Android malware. Furthermore, among the selected classifiers, the Extra-trees algorithm achieved the maximum accuracy results.

In comparison with a study from CIC, this model obtains better results in binary detection, and significant enhancement in malware categorization by 30.3% for precision and 31.14% for recall. Furthermore, the accuracy of malware family classification is improved by 39.71% for precision and 41.09% for recall.

8. Future Work

The proposed model comes up with some further research and implementation issues. These issues could be summarized in the following points:

1. Improve this study model by adding static features.
2. Implement a feature extraction tool that extracts most various network features for malware detection in the multilevel level (packet, flow, conversation, and connection) into CSV files.
3. Enhance the provided model by considering more factors in detecting android malware.

References

[1] Abuthawabeh M. and Mahmoud K., "Android Malware Detection and Categorization Based on Conversation-Level Network Traffic Features," *The International Arab Conference on Information Technology*, Al Ain, pp. 42-47, 2019

[2] Ahvanooey M., Li Q., Rabbani M., and Rajput A., "A Survey on Smartphone Security: Software Vulnerabilities, Malware, and Attacks," *International Journal of Advanced*

Computer Science and Applications, vol. 8, no. 10, pp. 30-45, 2017.

[3] Alauthman M., “An efficient Approach to Online Bot Detection Based, Doctoral Thesis,” Northumbria University, 2016.

[4] Arora A. and Peddoju S., “Minimizing Network Traffic Features for Android Mobile Malware Detection,” *in Proceedings of the 18th International Conference on Distributed Computing and Networking*, Hyderabad, pp. 1-10, 2017.

[5] Arora A., Garg S., and Peddoju S., “Malware Detection Using Network Traffic Analysis In Android Based Mobile Devices,” *in Proceedings of the 8th International Conference on Next Generation Mobile Applications, Services and Technologies*, Oxford, pp. 66-71, 2014.

[6] Arp D., Spreitzenbarth M., Hübner M., Gascon H., and Rieck K., “Drebin: Effective and Explainable Detection of Android Malware in Your Pocket,” *in Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, pp. 1-15, 2014.

[7] Bibi I., Akhunzada A., Malik J., Ahmed G., and Raza M., “An Effective Android Ransomware Detection Through Multi-Factor Feature Filtration and Recurrent Neural Network,” *in Proceedings of UK/China Emerging Technologies (UCET)*, Glasgow, pp. 1-4, 2019.

[8] Chebyshev V., Sinitsyn F., Parinov D., Kupreev O., Lopatin E., and Liskin A., “IT Threat Evolution Q2 2018. Statistics,” Haettu Osoitteesta Secure. [Online]. Available: com/it-threatevolution-q2-2018-statistics/87170, 2018, Last Visited, 2020.

[9] Chen R., Li Y., and Fang W., “Android Malware Identification Based on Traffic Analysis,” *in Proceedings of International Conference on Artificial Intelligence and Security*, New York, pp. 293-303, 2019.

[10] Draper-Gil G., Lashkari A., Mamun M., and Ghorbani A., “Characterization Of Encrypted And VPN Traffic Using Time-Related Features,” *in Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, Italy, pp. 407-414, 2016.

[11] F-Secure, “Android/Kmin.” 2012. [Online]. Available: https://www.f-secure.com/v-descs/trojan_android_kmin.shtml, Last Visited, 2020.

[12] Google, “Google Play Protect.” [Online]. Available: https://www.android.com/play-protect/Last Visited, 2020.

[13] Google, “Google Play Store.” [Online]. Available: https://play.google.com/store, Last Visited, 2020.

[14] Gupta S., “Types of Malware and its Analysis,” *International Journal of Scientific and Engineering Research*, vol. 4, no. 1, pp. 1-13, 2013.

[15] He G., Xu B., Zhang L., and Zhu H., “On-Device Detection of Repackaged Android Malware via Traffic Clustering,” *in Security and Communication Networks*, vol. 2020, no .7, pp. 1-19, 2020.

[16] Hamandi K., Chehab A., Elhajj I., and Kayssi A., “Android SMS malware: Vulnerability and Mitigation,” *in Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops*, Barcelona, pp. 1004-1009, 2013.

[17] IDC, “Smartphone Market Share,” 2019. [Online]. Available: https://www.idc.com/promo/smartphone-market-share/os, Last Visited, 2020.

[18] Kashefi I., Kassiri M., and Saleh M., “Preventing Collusion Attack in Android,” *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 719-727, 2015.

[19] Lashkari A., Kadir A., Taheri L., and Ghorbani A., “Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification,” *in Proceedings International Carnahan Conference on Security Technology*, Montreal, pp. 1-7, 2018.

[20] Lashkari A., Akadir A., Gonzalez H., Mbah K., and Ghorbani A., “Towards A Network-Based Framework for Android Malware Detection and Characterization,” *in Proceedings of 15th Annual Conference on Privacy, Security and Trust, PST*, Calgary, pp. 233-242, 2018.

[21] Liao Q., “Ransomware: a Growing Threat to SMEs,” *in Proceedings of Southwest Decision Sciences Institute’s Annual Conference*, Houston, pp. 360-366, 2008.

[22] Narang P., Hota C., and Venkatakrishnan V., “PeerShark: Flow-Clustering and Conversation-Generation for Malicious Peer-To-Peer Traffic Identification,” *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1-12, 2014.

[23] Nauman M. and Khan S., “Design and Implementation of A Fine-Grained Resource Usage Model for the Android Platform,” *The International Arab Journal of Information Technology*, vol. 8, no. 4, pp. 440-448, 2011.

[24] Parkour M., “Contagio malware database,” contagiodump. 2013. [Online]. Available: http://contagiodump.blogspot.com/2011/03/take-sample-leave-sample-mobile-malware.html, Last Visited, 2020.

[25] Point C., “FalseGuide misleads users on GooglePlay,” Check Point. 2016. [Online]. Available: https://blog.checkpoint.com/2017/04/24/falasegu

ide-misleads-users-googleplay/, Last Visited, 2020.

[26] Rashidi B. and Fung C., "A Survey of Android Security Threats And Defenses," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 6, no. 3, pp. 3-35, 2015.

[27] Sharma N., "Android Malware Detection using Decision Trees and Network Traffic," *in International Journal of Computer Science and Information Technologies*, vol. 7, no. 4, pp. 1970-1974, 2016.

[28] Singh R., "An Overview of Android Operating System and its Security," *International Journal of Engineering Research and Applications*, vol. 4, no. 2, pp. 519-521, 2014.

[29] Statista, "Number of available applications in the Google Play Store from December 2009 to June 2019," 2019. [Online]. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/, Last Visited, 2020.

[30] Statista, "Smartphone users worldwide 2014-2019". 2019. [Online]. Available: http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, Last Visited, 2020.

[31] Taosoftware Co. L., "tPacketCapture." 2012. [Online]. Available: https://www.taosoftware.co.jp/en/android/packetcapture/, Last Visited, 2020.

[32] Tcpdump/Libpcap, "Tcpdump and Libpcap," tcpdump. 2010. [Online]. Available: https://www.tcpdump.org/, Last Visited, 2020.

[33] Verma A., "WannaLocker - A New WannaCry-inspired Ransomware Is Attacking Android Smartphones," Fossbytes. 2017. [Online]. Available: https://fossbytes.com/wannalocker-ransomware-wannacry-android/, Last Visited, 2020.

[34] Virustotal, "Virustotal Free Antivirus Scanners," [Online]. Available: https://support.virustotal.com/hc/en-us/categories/360000160117-About-us, Last Visited, 2020.

[35] Whittaker Z., "Half a million Android users tricked into downloading malware from Google Play | TechCrunch," [Online]. Available: https://techcrunch.com/2018/11/20/half-a-million-android-users-tricked-into-downloading-malware-from-google-play/, Last Visited, 2020.

[36] Zhang C. and Ma Y., *Ensemble Machine Learning: Methods and applications*, Boston, MA: Springer, 2012.

[37] Zhou Y. and Jiang X., "Dissecting Android Malware: Characterization and Evolution," *in Proceedings IEEE Symposium on Security and Privacy*, San Francisco, pp. 95-109, 2012.

[38] Zulkifli A., Hamid I., Shah W., and Abdullah Z., "Android Malware Detection Based on Network Traffic Using Decision Tree Algorithm," *International Conference on Soft Computing and Data Mining*, Johor, pp. 485-494, 2018.

Mohammad Abuthawabeh received his master's degree from Princess Sumaya University for Technology, Jordan, in 2019. Currently, he is an information systems specialist of Jordan Anti Money Laundering and Counter Terrorism Unit, and a freelance information security researcher. His research interest includes Information security and machine learning.

Khaled Mahmoud get his BSc degree in Computer Science from Jordan University on June 1992, MSc degree in Computer Science (Artificial Intelligence) from Jordan University on 1998 and PhD degree in Print Security and Digital Watermarking from Loughborough University (UK) on 2004. This was followed by academic appointments at ZARQA Private University as an assistance Professor in computer Science. On 2018 he joined Princess Sumaya University as an academic staff in computer science department. His areas of interest include Information security, Digital watermarking, Image forgery detection, AI and Arabic language processing.