khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Detection of Cyber Malware Attack Based on Network Traffic Features Using Neural Network

Ventje Jeremias Lewi Engel[1*], Evan Joshua[2], Mychael Maoeretz Engel[2]
[1]Computer Engineering Department
Institut Teknologi Harapan Bangsa
Bandung
[2]Informatics Department
Institut Teknologi Harapan Bangsa
Bandung
[*]ventje@ithb.ac.id

Abstract-Various techniques have been developed to detect cyber malware attacks, such as behavior based method which utilizes the analysis of permissions and system calls made by a process. However, this technique cannot handle the types of malware that continue to evolve. Therefore, an analysis of other suspicious activities – namely network traffic or network traffic – need to be conducted. Network traffic acts as a medium for sending information used by malware developers to communicate with malware infecting a victim's device. Malware analyzed in this study is divided into 3 classes, namely adware, general malware, and benign. The malware classification implements 79 features extracted from network traffic flow and an analysis of these features using a Neural Network that matches the characteristics of a time-series feature. The total flow of network traffic used is 442,240 data. The results showed that 15 main features selected based on literature studies resulted in F-measure 0.6404 with hidden neurons 12, learning rate 0.1, and epoch 300. As a comparison, the researchers chose 12 features based on the nature of the malware possessed, with the F-measure score of 0.666 with hidden neurons 12, learning rate 0.05, and epoch 300. This study found the importance of data normalization technique to ensure that no feature was far more dominant than other features. It was concluded that the analysis of network traffic features using Neural Network can be used to detect cyber malware attacks and more features does not imply better detection performance, but real-time malware detection is required for network traffic on IoT devices and smartphones.

Keywords: cyberattacks; malware detection; neural network; network traffic feature

1. Introduction

As the adoption of society towards technology increases, the number of IoT (Internet of Things) devices and smartphones usage has been increasing and widespread. Security threats on IoT devices and smartphones also increase. Various cyberattacks can be committed on IoT devices and smartphones, ranging from taking access rights, destructing the data, thieving important information, and recording personal activities of users when using IoT devices and smartphones [1]. Most of these cyberattacks enter the system through malicious software or malware that are successfully planted on IoT devices and smartphones.

Malware is an application that has a negative purpose, such as corrupting data, stealing important information, disrupting device performance, and taking over the system. This threat continues to increase every year. In 2017, it is found around 3.5 million new malwares only on Android smartphone devices [2]. One of the suspicious activities of malware is the use of network traffic – can be applied as a medium for sending confidential information in the form of PINs, bank account information, personal messages, and passwords to malware makers [3]. Malware can also utilize network traffic as a backdoor for other malwares to enter.

The network traffic on IoT devices and smartphones has the same basis as network traffic in general, which contains packets that have a header and data section [4]. Data is obtained and processed at the application layer, while headers are added at each layer. The size of each data and header varies with the specified limits. The packet contains the data that the sender wants to send from source to destination. The header contains the destination IP address, sender's IP address, source port, destination port, and several other related information. Most network traffic features are time-series.

In general, malware detection system classifies applications into adware, general malware, and benign [5]. Adware is a type of malware that displays advertisements

on running software. Adware aims to increase revenue for software developers so that the advertised company pays for the adware. Each type of general malware is confirmed to have a negative purpose, such as damaging or stealing data. Benign is a normal type of application that does not have dangerous purposes; it runs according to what the application developer has written in the documentation section.

There are several efforts in detecting mobile malware that have been carried out using various approaches. Behavior-based approach that uses permissions and system calls as features, produces accuracy that is still relatively low with an average of 60%. Specifically, Simple Logistic 65.29%, Naive Bayes 65.29%, SMO 70.31% and Random Tree 54.79% [6]. Other studies using network traffic features using the Neural Network (NN) method to detect malware on smartphones have successfully detected malware botnets with a precision level of around 88.3% [7]. This result is much higher compared to the Naive Bayes and Logistic Regression methods, each of which has a value of 7% and 32% [7]. In addition, the NN method successfully outperformed the Support Vector Machine (SVM) method in classifying network traffic [8]. NN method is often used as a classification method because of its robust characteristics. It can even be used for quality classification [9]. Detecting malware through network traffic analysis – which is mostly in time-series data – suits with the NN machine learning method.

The weakness of the previous research is that the NN method is carried out on all network traffic features, despite there are several network features that has a more important role than other network traffic features. For example, the network destination port is more important than the length of the header contents. Second, the use all network traffic features results in the increase of the internal errors that carried in the data. Third, features with large values automatically weigh higher, for example the port values commonly used are much smaller, when being compared to the value of data flow across the network [5].

The difference between this study and previous research is the network traffic dataset, the combination of features, and the iteration of the NN configuration applied. The dataset applied in this research obtained from the Canadian Institute for Cybersecurity, University of New Brunswick [10] combined with sample data collected at the Harapan Bangsa Institute of Technology Computer Laboratory (ITHB). A total of 1900 android applications with a percentage of 20% malware and 80% benign. Malware is divided into two types including adware and general malware. The combination of features is carried out based on literature studies to obtain the intersection of network traffic features that are frequently used in malware detection system. The iteration of the NN configuration is conducted by programming that concern to learning rate, epoch, and parameter evaluation. The purpose of this study is to obtain the configuration of the NN model to detect cyber-type malware attacks and to investigate the combination of network traffic features that can result in high precision, recall, and F-measure in the detection of mobile malware using NN.

2. Methods

a. Research Flowchart

The research steps are arranged in the form of a flowchart, which begins with preprocessing. The preprocessing conducted is the normalization of features that will be used by dividing the features' values by the maximum value of each feature. Hence, this process will minimize features, so that a feature does not dominate other features.

Next, the learning stage applies the Neural Network method with backpropagation algorithm and the testing phase uses feed-forward method. In the initial phase, the weights will be randomly assigned in accordance with the previous provisions and they are stored in the file weight. Learning outcomes will give new weight values. The test will use the weight in the previous learning file. The test output is divided into 3, namely benign, adware, or general malware.

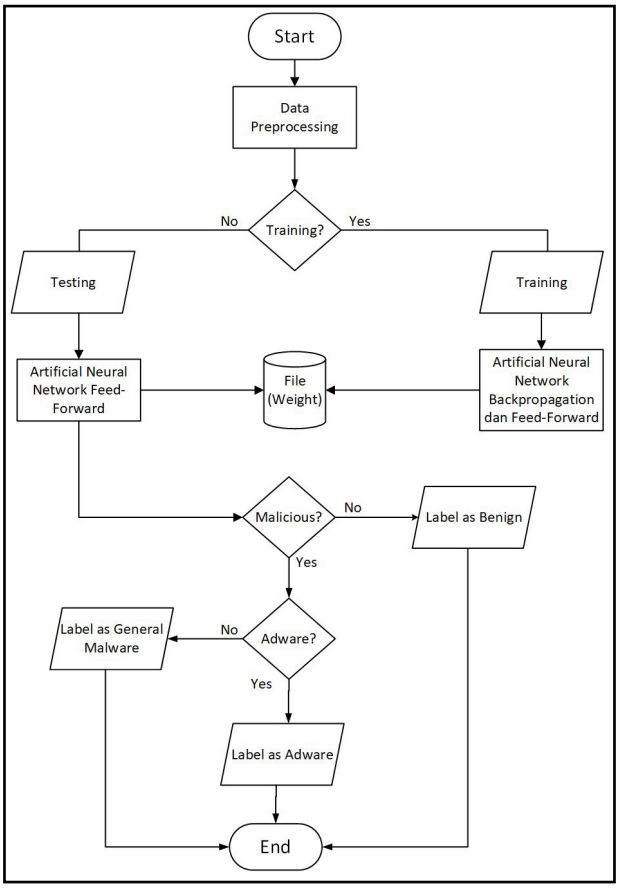

Figure 1. Research Flowchart

b. Neural Network Architecture

Neural Network (NN) or often also called Artificial Neural Network is one of machine learning techniques.

Neural networks are included in supervised learning, with the resulting model in the form of weight [11]. The weights are used at the test stage and the output is mapped to the activation function to determine which label the output refers to.

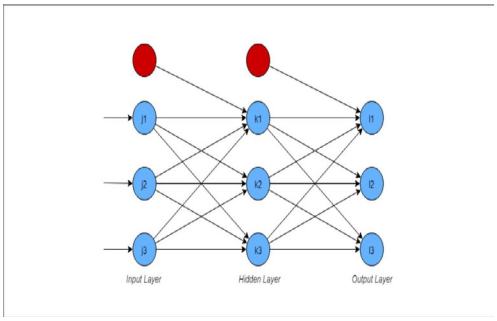

Figure 2. Neural Network Component

As shown in Figure 2, There are 3 main layers in the Neural Network, namely the input layer, hidden layer and output layer. It is also drawn several circles of various colors according to their role. The blue circles are called nodes or neurons, while the red circles are bias – benefit to increase the flexibility of the model.

The input layer acts as the layer that receives initial input. The input obtained is processed to produce output on the hidden layer. The hidden layer is situated between the input and output layers and is useful for supporting neural networks learning complex features. The hidden layer itself can contain several layers. Each layer in the hidden layer may have different number of neurons. The hidden layer will produce output which then subjects to an activation function, to be mapped to the class in the output layer.

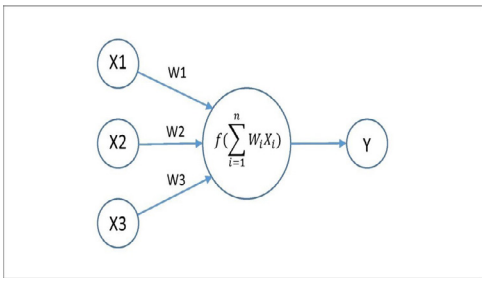

Figure 3. Calculation Process of Output at Layer

Figure 3 shows that each neuron has a weight according to the number of connections with other neurons. Output calculation is influenced by the weight and input values, which the results will then be processed with an activation function. According to Stevanovic [7], this mechanism makes Neural Network able to read and analyze simultaneously many features of network traffic for detection of malware with a high degree of precision.

In this study, three layers will be used, including the input layer, hidden layer and output layer. The input layer has a number of neurons according to the number of features used. In the hidden layer, only one layer will be used with the number of neurons tested, such as 4, 5, 6, and 12. The output layer will produce output in the form of 3 classes, namely benign, adware, and general malware. The test will apply several combinations of Neural Network parameters including learning rate, hidden neurons and the number of epochs. The learning rates tested are 0.1, 0.05, and 0.01 with the number of epoch 100, 200, and 300.

c. Dataset

Dataset used is a pcap (packet capture) file that contains network traffic packets with a total of 79 features. The pcap file was earned from a total of 1900 android applications with a percentage of 20% malware and 80% benign. The malware dataset is divided into three groups, including 250 adware applications, 150 general malware applications, and 1,500 benign applications. In the training data, there are 2,312 network traffic flows from general malware, 149,871 for adware, and 201,609 for benign, while in data testing there are 1,626 general flow malware, 24,271 flow adware, and 62,551 flow benign. The total flow of network traffic used is 442,240 data. By using the CICFlowMeter application, the pcap file is converted to CSV file, so that one flow means one line of data.

duration	total_fpack	total_bpack	total_fpktl	total_bpktl	min_fpktl	min_bpktl	max_fpktl	max_bpktl
1020586	668	1641	35692	2276876	52	52	679	1390
80794	1	1	75	124	75	124	75	124
998	3	0	187	0	52	-1	83	-1
189868	9	9	1448	6200	52	52	706	1390
110577	4	6	528	1422	52	52	331	1005
261876	7	6	1618	882	52	52	730	477
14	2	0	104	0	52	-1	52	-1
29675	1	1	71	213	71	213	71	213
806635	4	0	239	0	52	-1	83	-1
56620	3	2	1074	719	52	52	592	667
7552	1	1	52	64	52	64	52	64
5008461	1	2	52	135	52	52	52	83
59125997	2	4	780	664	52	52	728	477
155610	1	2	40	92	40	40	40	52
5220	1	1	40	52	40	52	40	52
19609	1	2	52	719	52	52	52	667
573997	121	273	7352	345069	52	52	770	1390
128911	1	1	68	161	68	161	68	161
0	1	0	83	0	83	-1	83	-1
48236088	10	9	1009	4326	40	40	315	1390
1570	1	1	68	148	68	148	68	148
46176642	5	5	352	642	40	40	172	465
1134	1	1	68	148	68	148	68	148
391249	14	12	1987	7755	52	52	987	1064
29824	1	1	70	268	70	268	70	268

Figure 4. Screenshot of CSV Datasheet File Contents

d. Feature Combination Analysis

The combination of features that will be used in the Neural Network is chosen based on the analysis, obtained from the literature study. The results of the literature study can be observed in Table 1.

Table 1. Key Features Network Traffic Literature Study Results

Numb	Feature Name	Reference
1.	Source port	[12][13]
2.	Destination port	[12][13]
3.	L3 / L4 Protocol identifier	[12][13]
4.	Total number of packets	[7][12][13][14][15]
5.	Total number of bytes	[7][12][14]
6.	Mean of number of bytes per packet	[7][12][15]
7.	Standard deviation of number of bytes per packet	[7][12]
8.	Number of packets per second	[7][12]
9.	Number of bytes per second	[7][12][13]
10.	Flow duration	[7][12][13][15]
11.	Mean of inter-arrival time (IAT)	[7][12]
12.	Standard deviation of IAT	[7][12]
13.	Ratio of number of packets OUT/IN	[12][13][14]
14.	Ratio of number of bytes OUT/IN	[7][12]
15.	Ratio of IAT OUT/IN	[7][12]

Table 2. Researcher's Selected Features

Numb	Feature Name	Category
1.	Forward packets	Packet based
2.	Total forward packets	Packet based
3.	Forward packet length max	Byte based
4.	Active mean	Time based
5.	Backward packets / second	Packet based
6.	Forward IAT standard deviation	Time based
7.	Max packet length	Packet based
8.	Total backward packets	Packet based
9.	Total length of backward packets	Byte based
10.	Backward IAT standard deviation	Time based
11.	FIN flag count	Flow based
12.	Packet length variance	Byte based

As a comparison, researchers chose 12 features according to researchers' understanding regarding malware. Adware variant has characteristics that interrupts the application to display the advertisements which is actually malicious code. This causes a lot of flow in the forward and backward packages. The twelve features selected by the researchers did not overlap with the features of the literature study results, and are informed in Table 2.

e. Objective and Evaluation

From the analysis of dataset, it was found that class imbalance occurred in malware label data, which was only 20% compared to benign (80%) [10] resulting in an evaluation computed with ordinary accuracy metric to be insufficient. Therefore, in this case, F-measure was used as a metric instead of accuracy. The F-measure is used to help in drawing conclusions about which Neural Network parameters are best implemented. The advantage of the F-measure is able to consider precision and recall into a single unit that is interconnected with one another. Table 2 shows the confusion matrix used to obtain the values of True Positive, False Positive, True Negative and False Negative.

Table 3. Confusion Matrix

Prediction Value	True Value	
	TRUE	***FALSE***
TRUE	True Positive (TP)	False Positive (FP)
FALSE	False Negative (FN)	True Negative (TN)

Table 4 Neural Network Results Using the Literature Study Features vs. Researcher Features vs. Combined Features

Numb.	Combination of Features	Number of Features	Hidden Neuron	Learning Rate	Epoch	Training time	Testing time	Precision	Recall	F-measure
1	**Literature Study Features**	15	12	0.1	300	28m 50s	6s	47.58%	97.88%	0.6404
2	**Researcher Features**	12	12	0.05	300	27m 43s	4s	55.89%	82.39%	0.6660
3	**Combined Features**	27	12	0.1	300	30m 3s	4s	47,40%	98.26%	0.6395

Table 5. Comparison of Feature Combinations in Hidden Neurons of 12

Hidden Neuron = 12	Learning Rate	Epoch	Precision	Recall	F-measure
Literature Study Features	0.1	100	47.34%	98.07%	0.6386
		200	47.44%	98.14%	0.6396
		300	**47.58%**	**97.88%**	**0.6404**
	0.05	100	63.83%	49.85%	0.5598
		200	64.46%	49.78%	0.5618
		300	65.30%	49.75%	0.5648
	0.01	100	70.69%	42.65%	0.532
		200	71.30%	42.57%	0.5331
		300	71.68%	42.38%	0.5326
Researcher Features	0.1	100	0.00%	0.00%	0
		200	0.00%	0.00%	0
		300	0.00%	0.00%	0
	0.05	100	53.93%	85.13%	0.6603
		200	54.94%	83.45%	0.6626
		300	**55.89%**	**82.39%**	**0.6660**
	0.01	100	0.00%	0.00%	0
		200	0.00%	0.00%	0
		300	0.00%	0.00%	0

Formulas (1), (2), and (3) are employed for determining the value of precision, recall, and F-measure, respectively.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F\text{-}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

3. Results and Discussion

The implementation and testing environment is conducted in cloud computing since the CSV data that must be processed is quite large, both for training and testing. Weight configurations on the Neural Network are randomly generated. Then, the first training process is carried out – the weights are updated. The training process is conducted continuously until the specified epoch is finished. After that, testing is carried out with feed forward. Table 4 shows a comparison of Neural Network results with features obtained from literature studies and features earned from researchers' knowledge.

Complete test results for each combination of features are given in the supplement of this article. The highest F-measure was achieved for hidden neurons number 12. These results are consistent with Stevanovich's research [7, 12] which states that the more hidden neurons used, the Neural Network performance tends to be better until it finds a saturation point. This is different for learning rate. Comparison of learning rate and epoch for each combination of features in hidden neurons totaling 12 can be seen in Table 5.

A higher learning rate does not guarantee that the F-measure results will also be better. In the combination of researchers' features, the best results are achieved when the learning rate is 0.05 only. The combination of literature study features does achieve the best results with maximum configuration of Neural Network parameters (learning rate 0.1 and epoch 300). Technically, learning rate is the magnitude of change given to the weight which is changed according to the error value. Whereas, the epoch indicates the number of iterations performed by the computer. Learning rate that is overly high or low might result in new weights at further deviation than the expected weights. From Table 5, it is shown that in the combination of researchers' features, there are several learning processes that produce a value of 0 for precision, recall, and F-measure. This is assumed that the model produced with these parameters experienced underfitting when the learning rate is 0.01 and 0.1.

The F-measure score of the combination of 12 researchers' features is greater than the combination of 15 features of literature studies (0.6660 > 0.6404). This shows that using more features does not necessarily improve the accuracy of malware detection on the Neural Network. It is obvious that the two sets of feature combinations do not intersect, but have slightly different F-measure values. It means that there are still combinations of features that are likely to produce F-measure values better than both. For this reason, researchers merged the two combinations of features and conducted testing and training once again. The results of the merged combination of 27 features earned the highest score of F-measure on the number of hidden neurons 12, learning rate 0.1, and epoch 300; the resulted F-measure is 0.6395 (see Table 4). This score is lower than the results of a combination of literature study features. These results once again show that more features do not necessarily improve detection accuracy. This is because the more features used might result in more internal errors which were involved in the learning process. Each feature has internal errors, such as errors due to measurement or errors due to rounding values [13]. Another factor is that each feature has its own contribution in malware detection and there is a possibility that features that are combined together have the effect of eliminating each other, so that the detection accuracy might decrease [15].

4. Conclusion

Detection of cyber malware attacks based on network traffic features using Neural Network results in different F-measure values for different combinations of features. A combination of features based on literature studies (15 features) produces an F-measure of 0.6404, a combination of researchers' analysis features (12 features) produces an F-measure of 0.6660, and a combination of the two combined features (27 features) produces an F-measure of 0.6395. The conclusion is that the number of features does not mean that the accuracy of malware detection will increase. Instead, an improper combination of features can reduce detection accuracy.

This research uses Dataset with 442,240 data which is a combination of existing Dataset and the results of laboratory experiments, for the learning process. It is recommended that the existing Neural Network model can be applied to detect malware in real time on IoT devices and smartphones. Additionally, further research is also needed on the analysis of the combination of network traffic features to produce even better accuracy.

Acknowledgement

This research was funded through a grant from the Ministry of Research, Technology and Higher Education *(Kementerian Riset, Teknologi, dan Pendidikan Tinggi.)*

References

[1] Kaspersky, "Mobile Malware Threatens Smartphones & Tablets," *Kaspersky Lab ZA*, 2015. [Online]. Available: https://www.kaspersky.co.za/resource-center/threats/mobile-malware. [Accessed: 18-Jul-2018].

[2] C. Lueg, "8,400 new Android malware samples every day," *G Data Security Blog*, 2017. [Online]. Available: https://www.gdatasoftware.com/blog/2017/04/29712-8-400-new-android-malware-samples-every-day. [Accessed: 18-Jul-2018].

[3] Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and Evolution," in *Proceedings - IEEE Symposium on Security and Privacy*, 2012, no. 4, pp. 95–109.

[4] B. A. Forouzan, *TCP/IP Protocol Suite*, 4th ed. New York: McGraw-Hill Companies, Inc., 2010.

[5] A. H. Lashkari, A. F. A. Kadir, H. Gonzalez, K. F. Mbah, and A. A. Ghorbani, "Towards a Network-Based Framework for Android Malware Detection and Characterization," in *Proceeding of the 15th international conference on privacy, security and trust*, 2017.

[6] P. Kaushik and A. Jain, "Malware Detection Techniques in Android," *Int. J. Comput. Appl.*, vol. 122, no. 17, pp. 22–26, 2015.

[7] M. Stevanovic and J. M. Pedersen, "An analysis of network traffic classification for botnet detection," in *2015 International Conference on*

Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2015, pp. 1–8.

[8] J. Zhang, Y. Xiang, and Y. Wang, "Network Traffic Classification Using Correlation Information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, 2013.

[9] F. Wibowo and A. Harjoko, "Klasifikasi Mutu Pepaya Berdasarkan Ciri Tekstur GLCM Menggunakan Jaringan Saraf Tiruan," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 3, no. 2, pp. 100–104, 2018.

[10] Canadian Institute for Cybersecurity, "Android Adware and General Malware Datasheet ," *University of New Brunswick*, 2017. [Online]. Available: https://www.unb.ca/cic/Datasheet s/android-adware.html. [Accessed: 18-Nov-2018].

[11] T. Rashid, *Make Your Own Neural Network: A Gentle Journey Through the Mathematics of Neural Networks*. CreateSpace Independent Publishing Platform, 2016.

[12] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, 2014, pp. 797–801.

[13] H. Lim, Y. Yamaguchi, H. Shimada, and H. Takakura, "Malware Classification Method Based on Sequence of Traffic Flow," in *2015 International Conference on Information Systems Security and Privacy (ICISSP)*, 2015, pp. 394–401.

[14] D. Jiang and K. Omote, "An approach to detect remote access trojan in the early stage of communication," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, 2015, pp. 706–713.

[15] Z. B. Celik, R. J. Walls, P. Mcdaniel, and A. Swami, "Malware Traffic Detection using Tamper Resistant Features," in *MILCOM 2015-2015 IEEE Military Communications Conference*, 2015, pp. 330–335.