2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) | 978-1-6654-4058-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/HORA52670.2021.9461302

Neural Network Based Android Malware Detection with Different IP Coding Methods

Esra Calik Bayazit
Fatih Sultan Mehmet Vakif University
Marmara University Institute of Science, Computer Engineering Department
Istanbul/Turkey
ecalik@fsm.edu.tr

Ozgur Koray Sahingoz
Biruni University
Computer Engineering Department
Topkapi/Istanbul/Turkey
osahingoz@biruni.edu.tr

Buket Dogan
Marmara University
Department of Computer Engineering, Faculty of Technology
Istanbul/Turkey
buketb@marmara.edu.tr

***Abstract*— Due to the COVID-19 epidemic that has affected the whole world, internet use has increased more than in previous years. Almost all operations and transactions are done over the internet, especially with the use of cellular phones and tablet PCs. This growth results in many security deficits that need to be solved by security admins and end users. Malicious software (malware) is generally preferred for attacking the computer systems and recently for cellular phones. As a mobile operating system, Android is the main player of this sector with about 72% market share worldwide. Therefore, malware attacks especially target these devices, for reaching the maximum number of victims. The situation is getting more and more devastating with around 12,000 new Android malware attacks every day. This is one critical problem that needed to be solved by setting up an android malware detection system. Machine learning algorithms are frequently preferred in data mining-based security applications which contain lots of features in datasets. Artificial Neural networks are one of the mostly preferred learning models for training the system. Therefore, in this paper, it is aimed to implement a neural network based android malware detection system by using an up-to-date dataset presented by the Cyber Security Institute of Canada as CICMalDroid2017. Ip Addresses are one of the features in this dataset, and we focus on two different IP coding methods, as IP Splitting to Four Numbers, IP Transform to integer number, and no IP Address. In experimental study we reached a good level of accuracy rate as 98.4% by splitting an IP address to four numbers.**

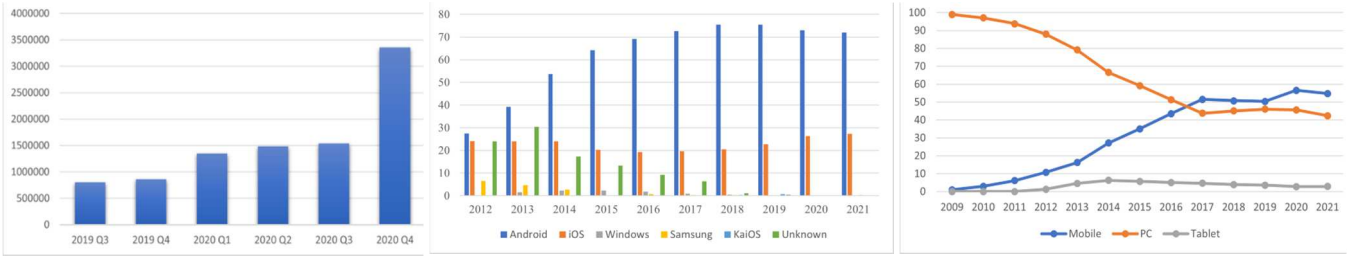
Keywords—Machine Learning, ANN, Android System, Malware Detection

I. INTRODUCTION

With the recent digital transformation, the amount of big data has been increasing exponentially. Android systems have a large 72.2 % market share in this digital transformation [1]. For this reason, it is very important to develop effective techniques to analyze and detect malware application threats. Machine learning algorithms play important roles in various fields in the analysis of big data. The number and quality of features in big data directly affect the correct detection and classification performance.

In addition to the various optimization methods that increase the performance rate in machine learning algorithms used to obtain the optimal solution, the efficiency of the features is also one of the most important factors that should be taken into consideration. For example, fake IP addresses are the most qualified feature in network attacks that affect system performance. Regarding this, every device connected to the Internet network must have a certain address called IP address to exchange data. An IP address appears as a label consisting of four numbers separated by dots, assigned to all devices connected to the Internet. Although the IP address is a numeric tag, it cannot be directly processed by machine learning algorithms. It appears that IP addresses are not quantitative variables and are not considered in machine learning detection processes due to conversion processes [2]. However, the IP address is one of the features that should not be ignored in the detection of attacks. Malware attacks can invade the system on vulnerable operating systems and contaminate other systems on the network. This type of malware often adversely affects system performance and causes notifications. The multitude of usage areas of Android systems makes it even more important to ensure the security of these systems.

According to the "McAfee Labs Threats Report: April 2021 report, the attackers are targeting Android devices [3]. Figure 1(a) shows the 2020 significant increase in new Android malware threats. Accordingly, as shown in Figure 1 (b), when the period from 2012 to April 2021 is considered, it is seen that the use of Android operating systems always has the largest share compared to other operating systems [1]. For this reason, every feature that will ensure the security of Android systems



a) McAfee Lab Threat Report b) Operating System Market Share Worldwide c) Platform Market Share from

Figure 1 Worldwide statistics

978-1-6654-4058-5/21/$31.00 ©2021 IEEE

should be evaluated. On the other hand, it is clearly seen in Figure 1 (c) that PC usage has evolved into mobile device usage over the years [4].

As shown in Figure 1 android devices become the target of attackers depending on the intensity of use clearly reveals the need to protect against all threats. In this case, every feature that will increase the detection performance rate should be available. Every device that connects to the internet has an IP address. The use of this feature in detecting system attacks is among the factors that affect the success rate. With the suggested study, it is aimed to increase the detection performance rate by ensuring the use of IP addresses in machine learning algorithms. There are 84 features in the Adware-Benign dataset created. In the study, the performance examination results obtained by using the ANN algorithm were obtained with two different IP coding methods and no IP. IP address information is one of the features that should be considered in detecting attacks. In network attacks, it attacks the system and affects the system performance. IP packets are often sent from a fake address so attackers can hide themselves. In fact, one of the most common ways to attack a network is IP address spoofing. In network attacks, it invades the system and affects the system performance. The dataset created includes two different IP addresses: Source IP address and Destination IP address. The conversions of these IP addresses are provided by two different methods. In addition, malware detection was made with the data set where the IP address was not available. Classification success rates are compared by using the mentioned methods on up-to-date data sets. Table 3 shows the performance examination results.

The rest of the paper is organized as follows. Section II is the literature survey about malware detection. Then in Section III, a detailed explanation is given about the Adware-Benign data set created from the CICAndMal2017 data set used in the study. In Section IV, our malware detection system's methodology and their results are depicted. Finally, conclusions and future works are drawn.

II. LITERATURE SURVEY

Due to the increased number of Internet connected devices, the computer security either as networks security or as end user security, such as phishing type of attacks [5, 6], is a major concern of the security admins. These experts can protect their systems with the use of firewalls; however, this cannot be sufficient for prevention of all types of network attack. Therefore, intrusion detection systems are accepted for a better approach for the detection of attacks [7, 8]. But malicious software detection is another type of security breach of the computer systems. In the literature there are some solutions for this type of attack [9].

Usman et al. proposed a new hybrid approach based on Dynamic Malware Analysis, Cyber Threat Intelligence, Machine Learning (ML) and Data Forensics in their study in 2021. In the study, it was aimed to define the behavior of IP addresses, to prevent cyber-attacks due to the importance of behavior of an IP address in security threat behavior and the problems of repetitive IP attacks. They have demonstrated the malicious behavior of IP addresses at runtime. Malware families have been classified using the Decision Tree machine learning algorithm methods. IP reputation is predicted at the pre-acceptance stage with zero-day attacks, and it is predicted using the Decision Tree (DT) technique through behavioral analysis. According to the results of the study, with the help of DT, the false alarm rate was reduced as it returned a 2% error rate and 99% F-measurement score [10].

Shao's thesis in 2019 determined the best method of IP address encoding for analyzing the F-1, precision, recall and accuracy scores of machine learning algorithms and for network attack detection. They also examined the processing speeds according to the classification algorithms used in the 21 features that they considered in the study. The study shows that the best method of encoding an IP address is to divide the IP address into four numbers, and the DT, RF and SVM accuracy scores for this method are 0.9562, 0.9631 and 0.9296, respectively [11].

Mahdavifar et al. in 2020, developed a new dataset of Adware, Banking, SMS, Riskware and Benign attacks, which contains a total of 17,341 static and dynamic data belonging to five different Android application categories, which they call CICMalDroid2020. In the study, the malware category was classified by applying a semi-controlled technique on the data set. For this purpose, they used a deep neural network that takes the frequencies of dynamic behaviors as input. They used the pseudo-label [12] method to train deep neural networks in a semi-controlled manner. The reason for using the so-called label is explained as follows. Semi-supervised learning is the most beneficial machine learning technique if there is a limited number of labeled data for each class. Machine learning models have been found to be insufficient to solve real-world problems with intrinsic confusion and big data in the manual feature extraction process. "Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), and k-Nearest Neighbor (k-NN) machine learning techniques were compared with the proposed model and semi-controlled Label Spread on CICMalDroid2020. In the classification of Android applications according to the malware category, the F1 score was 97.84% and the false positive rate was 2.76% [13].

In the study Kim et al. proposed a multilayer deep learning software detection model that uses entity or similarity-based feature extraction methods to reflect the features of the Android application, extract and improve features, and achieve an effective feature representation. Seven types of static property extraction including permissions, components, environment, strings, Dalvik opcode strings, API call sequences, and shared library function opcode properties. Each type of feature was used to train the first network of the respective deep neural network, respectively. The training results of the first network were then used to train the final network. In the data set consisting of 13,075 malicious software and 19,747 benign software, it was determined that the applied model reached 98% detection accuracy [14].

Android malware detection studies have been a research point in recent years. It is seen in the reviewed literature studies that different methods are used to detect Android malware. For this reason, it is important to develop methods that increase the success rate of attack detection.

III. DATASET

The data set used in this study was taken from the website of the University of New Brunswick, Canadian Institute for Cybersecurity [15]. CICAndMal2017 dataset created by Lashkari et al. collected more than 10,854 samples (4,354 malware and 6,500 benign) from several sources. 426 malware and 5,065 benign obtained by performing dynamic analysis on real devices. The benign software was created by collecting from the most popular free applications published in the Google Play market in 2015, 2016, 2017. The malware families collected are categorized as adware, ransomware, scareware, and SMS malware. These examples are attacks belonging to a total of four families and have a being label. The number of examples according to the types of attacks in the CICAndMal2017 dataset is shown in Figure 2.

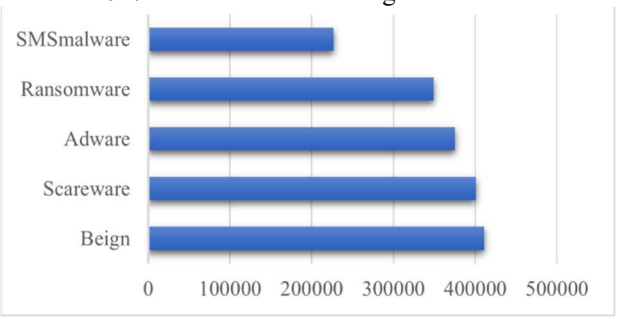


Figure 2 Attack Numbers by Families

- **Adware Malicious Applications:** It consists of 104 applications including Ewind, Dowgin, Gooligan, Feiwo, Shuanet, Kemoge, Youmi, Koodous, Mobidash and Selfmite families.
- **Ransomware Malicious Applications:** Charger, Pletor, Jisut, PornDroid, Koler, RansomBO, LockerPin, Svpeng, Simplocker, WannaLocker consists of 101 applications.
- **Scareware Malicious Applications**: It consists of 102 applications including AndroidDefender, FakeApp.AL, AndroidSpy.277, FakeAV, AV, FakeJobOffer, FakeTaoBao, Penetho, FakeApp families.
- **SMS Malware Applications:** It consists of 99 applications including Bean Bot, Ji Fake, Bilge, Mazarbot, FakeInst, Nandrobox, FakeMart, Plankton, FakeNotify, SMS sniffer families.
- **Benign Applications:** It consists of 1700 benign applications obtained from Google Play market in 2015-2016.

CICAndMal2017 dataset has a total of 84 features and a label (Attack, Normal). These features are shown in Table 1. The features marked in bold in the table were not used by removing them from the dataset with the methods described in section IV.

In this study, it is aimed to examine the effect of IP Addresses on artificial neural networks classification. 375,564 data belonging to the adware category and 410,548 data belonging to the benign category of the CICAndMal2017 dataset were combined. As a result, an up-to-date dataset consisting of 786,112 data in total was used. The pre-processing phase of the features of the up-to-date Adware-Benign dataset created and is explained in Section IV.

Table 1 CICAndMal2017 Dataset Feature

Feature	**Feature**	**Feature**
Flow ID	Fwd IAT Min	Avg Bwd Segment Size
Source IP	Bwd IAT Total	Fwd Header Length. 1
Source Port	Bwd IATMean	**Fwd Avg Bytes/Bulk**
Destination IP	Bwd IAT Std	**Fwd Avg Packets/Bulk**
Destination Port	Bwd IAT Max	**Fwd Avg Bulk Rate**
Protocol	Bwd IAT Min	**Bwd Avg Bytes/Bulk**
Timestamp	**Fwd PSH Flags**	**Bwd Avg Packets/Bulk**
Flow Duration	**Bwd PSH Flags**	**Bwd Avg Bulk Rate**
Total Fwd Packets	Fwd URG Flags	Subflow Fwd Packets
Total Backward Packets	**Bwd URG Flags**	Subflow Fwd Bytes
Total Length of Fwd Packets	Fwd Header Length	Subflow Bwd Packets
Total Length of Bwd Packets	Bwd Header Length	Subflow Bwd Bytes
Fwd Packet Length Max	Fwd Packets/s	Init_Win_bytes_forward
Fwd Packet Length Min	Bwd Packets/s	Init_Win_bytes_backward
Fwd Packet Length Mean	Min Packet Length	act_data_pkt_fwd
Fwd Packet Length Std	Max Packet Length	min_seg_size_forward
Bwd Packet Length Max	Packet Length Mean	Active Mean
Bwd Packet Length Min	Packet Length Std	Active Std
Bwd Packet Length Mean	Packet Length Variance	Active Max
Bwd Packet Length Std	FIN Flag Count	Active Min
Flow Bytes/s	SYN Flag Count	Idle Mean
Flow Packets/s	**RST Flag Count**	Idle Std
Flow IAT Mean	PSH Flag Count	Idle Max
Flow IAT Std	ACK Flag Count	Idle Min
Flow IAT Max	URG Flag Count	
Flow IAT Min	CWE Flag Count	
Fwd IAT Total	**ECE Flag Count**	
Fwd IAT Mean	Down/Up Ratio	
Fwd IAT Std	Average Packet Size	
Fwd IAT Max	Avg Fwd Segment Size	

IV. METHODOLOGY

In the proposed study, binary classification has been carried out with different IP transform methods using the Artificial Neural Network (ANN) learning model. The effects of features on performance rates were analyzed with the created up-to-date data sets. With the feature selection, the size of the dataset can be reduced as well as the classification success can be increased, overfitting during training can be eliminated, and the training time of the models can be shortened [16]. Features selection process in a machine learning model directly affects

the success of the model in classification. The main reasons for choosing the ANN algorithm in the study are listed below.

• They can generate information for unseen outcomes. unsupervised learning is involved.
• They can make pattern recognition and classification.
• They can complete missing patterns.
• They have fault tolerance. They can work with incomplete or ambiguous information. In faulty cases, they show graceful degradation.
• They can operate in parallel and process real-time information [17].

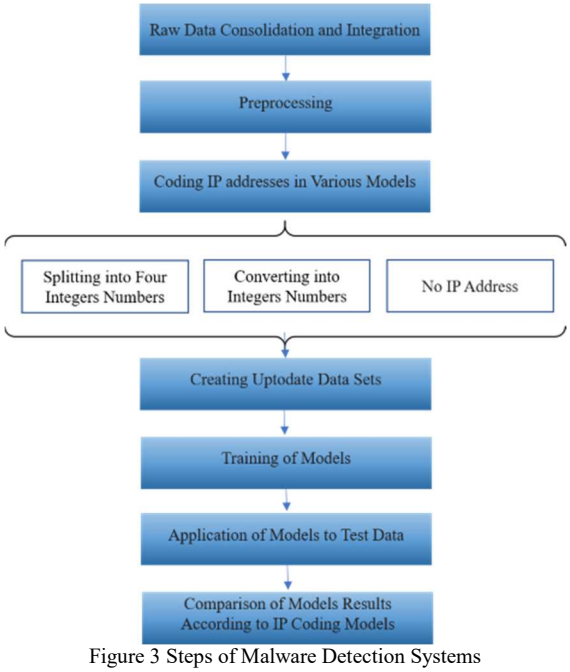


Figure 3 Steps of Malware Detection Systems

In other words, ANN are self-learning algorithms that produce better results as the number of features increases. In the proposed system, the Sequential model is used for a flat layer stack where each layer has exactly one input tensor and one output tensor. The basically used parameters for each proposed method are: three hidden layers, classifier optimizer is Adam, 50 epochs, and the loss function is binary cross entropy. The system consists of two main elements, pre-processing and classification. The values of the parameters used are shown in table 2.

Table 2 ANN Classifier Parameters

	Test 1	Test 2	Test 3
Input Layer	77	71	69
1st Hidden Layer	64	32	16
2nd Hidden Layer	32	64	64
3rd Hidden Layer	16	32	16
Output Layer	1	1	1
Layer Activation Func.	ReLu		
Output Activation Func.	Sigmoid		
Classifier Optimizer	Adam		
Loss Function	Binary Crossentropy		
Dropout	0.5		
Epoch	50		
Batch Size	32		

Performance rates were determined by analyzing the result of binary classification as non-malicious and malicious using ANN algorithm according to IP coding models. IP address coding and implementation methodology is explained in later work. The process steps performed in the proposed system are shown in Figure 3. Python 3.8.3 was used in the study, and the hardware features of the computer are shown as Table 3.

Table 3 System Properties

Property	Value
CPU	i7-8700K
# of Core	6
# of Threads	12
TurboBoost	4.70 GHz.
Cache L1/L2/L3	64K/256K/12MB
Memory Type	DDR4-2666
RAM	16 GB
Operating System	Windows- 64-bit OS
Display card	Nvidia G-Sync

The confusion matrix was run for the accuracy of classifiers, and F1 score were evaluated. Precision or recall were used to measure accuracy and F1- score was used for the imbalanced data. The formulas of the performance evaluation metrics used are given below [18].

TP: Predicted Positive, Actual Positive.
TN: Predicted Negative, Actual Negative.
FP: Predicted Positive, Actual Negative.
FN: Predicted Negative, Actual Positive.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + TN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F1\_Score} = 2 * \frac{Precision * Recall}{Precision * Recall} \quad (4)$$

$$\text{Error Rate} = 1 - Accuracy \quad (5)$$

As a result, the splitting IP address method was determined to be the most reliable approach. In the data set, Flow ID, Source IP address, Destination IP address data type is object, Timestamp data type is datetime and other remaining properties have variable values of numeric type. For machine learning, these non-digital features must be transformed. First, Flow ID features in the dataset; It is a combination of Source IP, Source Port, Destination IP, Destination Port and Protocol features. For this reason, the Flow ID feature has been removed from the entire data set created. Two new data sets were created by converting Source IP and Destination IP addresses with two suggested methods. In addition, Source IP and Destination IP addresses were removed completely and another data set was created. Flow ID has been removed from each of these up-to-date data sets. In the study, 72 features remained by eliminating unique values using the Pandas library of 84 features of the data set.

Another feature of datetime data type in the data set is Timestamp. A timestamp is a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day, sometimes accurate to a small fraction of a second. The timestamp property needs to be transformed into a form usable in machine learning classification. By using freq = 'D' parameter of Pandas library's date range function (2017-01-01 ',' 2017-12-31) it has been converted to str data type according to the frequency of occurrence in the date range.

Then it was converted to integer data type using the NumPy library. Adware and benign files are available separately as comma separated values (CSV). Both file structures, feature names and feature numbers are identical. These files were combined with the merge function to create a data set of 786,112 lines in total.

There are different attack examples belonging to ten families in the adware data set. These attack instances are labeled as 1 Malware and all instances of benign type as 0, using the Label Map function, to be used in binary classification, all families of the Adware category. IP address information is one of the features that should be considered in detecting attacks. In network attacks, it invades the system and affects the system performance. The data set created includes two types of IP addresses, Source IP, and Destination IP. Source IPs are called packet sending IP, Destination IP is called IP address receiving packets. IP addresses aggregated as IPv4 is a 32-bit number and displayed as four separate numbers and divided into three dots. Coding models of IP addresses are mentioned below.

A. Splitting an IP Address to Four Integer Number(Method1)

To convert IP addresses, the 32-bit address is divided into four separate numbers. These four numbers have been accepted as four distinct features. After converting the Source IP and destination IP, these IP addresses are included in the data set using the *concat* function as eight different properties. Later, Source IP and Destination IP were removed from the data set. Thus, there were 77 features in the up-to-date dataset. Figure 4 shows the process of splitting an IP Address. Splitting an IP Address to Four Integer Number method is expressed as *Method 1* in classification performance examination results tables.

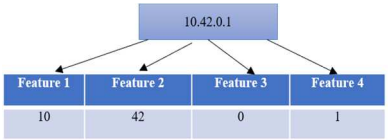


Figure 4 Splitting an IP Address

B. IP Transform to Integer Number (Method2)

Source IP and Destination IP are converted to integer numbers from the data set. Thus, there were 71 features in the up-to-date dataset. The IPv4 / IPv6 manipulation library has been imported to change IP addresses. "*ipaddress"* function provides the capabilities to create, manipulate and operate on IPv4 and IPv6 addresses and networks. The functions and classes in this module make it straightforward to handle various tasks related to IP addresses, including checking whether two hosts are on the same subnet, iterating over all hosts in a particular subnet, checking whether a string represents a valid IP address or network definition. Return an IPv4Address or IPv6Address object depending on the IP address passed as argument. Figure 5 shows the IP Address Transform to Integer Number. IP Transform to Integer Number method is expressed as *Method 2* in classification performance examination results tables.

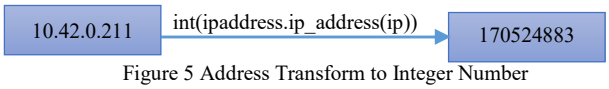


Figure 5 Address Transform to Integer Number

C. No IP (Method3)

Source IP and Destination IP were removed from the data set, and classification was made using 69 features and an attack / normal tag. No IP method is expressed as *Method 3* in classification performance examination results tables.

These 3 different methods we tested our system which are designed according to Table 2 with different test parameters. Experimental results are given in Table 4-6 all methods are shown in comparison with performance examination results using different dense.

Table 4 Classification Performance Examination Results for Method1

	Precision	F1-Score	Accuracy	Error Rate
Method 1	1.00	0.98	0.982	0.018
Method 2	1.00	0.98	0.975	0.025
Method 3	0.99	0.98	0.974	0.026

Table 5 Classification Performance Examination Results Method 2

	Precision	F1-Score	Accuracy	Error Rate
Method 1	0.97	1.00	0.984	0.016
Method 2	0.96	0.99	0.972	0.028
Method 3	0.96	0.99	0.973	0.027

Table 6 Classification Performance Examination Results Method 3

	Precision	F1-Score	Accuracy	Error Rate
Method 1	0.97	1.00	0.981	0.019
Method 2	0.96	0.99	0.971	0.029
Method 3	0.95	0.99	0.973	0.027

As seen from these results, Splitting an IP Address to Four Integer Number, which is our Method1, gave the best results in all of our experiments and best test platform is seen as designed 32-64-32 Artificial Neural Network model. Use of different ANN design can give better solution but they needed to be tested. However, use of other learning approaches can give more accurate results for us.

V. CONCLUSIONS

In recent years there is a growing security problem in all network-connected devices due to the anonymous structure of Internet. Currently trend of computers changed from desktop computer to Tablets and smartphones, which use mobile operating system that can be more vulnerable to attacks. Android, as a more preferred operating systems, is the main target of the attackers to develop some malicious software (malwares). Therefore, developing a malware detection system is critical for security admins.

In malware detection system, different machine learning models can be preferred. In this study, preferred Neural Network approach as learning model and use three different IP_address coding for increasing the performance as Splitting an IP Address to Four Integer Number, IP Transform to Integer Number and NO IP address. Experimental results showed that Splitting an IP Address to Four Integer Number method gave better accuracies. We also designed 3 different Neural networks with 3 hidden layers, and best results are reached with the design of 32-64-32 hidden layer structure.

Use of different machine learning models can results better performance. Therefore, in the future work we will focus on this by using different learning models, especially as a more trending model of Deep Learning.

VI. ACKNOWLEDGMENT

This work has been supported by Marmara University Scientific Research Projects Coordination Unit under grant number FDK-2020-10066.

REFERENCES

[1] Statista, Mobile Operating Systems Market Share Worldwide from January 2012 to January 2021, 2021, access in May 2021.

[2] I. Sharafaldin, A. Habibi Lashkari and A. A. Ghorbani, “Toward Generating A New Intrusion Detection Dataset and Intrusion Traffic Characterization”, In Proceedings of the 4th international conference on information systems security and privacy - volume 1: ICISSP, (p. 108-116), SciTePress, doi: 10.5220/0006639801080116, 2018.

[3] McAfee Labs Threats Report: April 2021,2021, access in May 2021.

[4] Statista, Desktop vs Mobile vs Tablet Market Share Worldwide 2009 to 2021, 2021, access in 2021 May 2021.

[5] E. C. Bayazit, O. Koray Sahingoz and B. Dogan, "Malware Detection in Android Systems with Traditional Machine Learning Models: A Survey," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-8, doi: 10.1109/HORA49412.2020.9152840. 5

[6] E. Buber, B. Diri and O. K. Sahingoz, "Detecting phishing attacks from URL by using NLP techniques," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 337-342, doi: 10.1109/UBMK.2017.8093406. 6

[7] E. Buber, B. Diri and O. K. Sahingoz, “NLP Based Phishing Attack Detection from URLs”. Intelligent Systems Design and Applications. ISDA 2017. Advances in Intelligent Systems and Computing, vol 736. Springer, Cham. doi.org/10.1007/978-3-319-76348-4_59. 7

[8] G. Karatas, O. Demir and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," in IEEE Access, vol. 8, pp. 32150-32162, 2020, doi: 10.1109/ACCESS.2020.2973219. 8

[9] O. Can and O. K. Sahingoz, "An intrusion detection system based on neural network," 2015 23rd Signal Processing and Communications Applications Conference (SIU), 2015, pp. 2302-2305, doi: 10.1109/SIU.2015.7130338. 9

[10] N. Usman, S. Usman, F. Khan et al., “Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics,” Future Generation Computer Systems, vol. 118, p. 124, 2021.

[11] E. Shao, Encoding IP Address as a Feature for Network Intrusion Detection. Diss. Purdue University Graduate School, 2019.

[12] D.H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Workshop on challenges in representation learning, ICML, vol. 3, 2013, p. 2.

[13] S. Mahdavifar, A. F. Abdul Kadir, R. Fatemi, D. Alhadidi, A.A. Ghorbani,” Dynamic Android Malware Category Classification using Semi-Supervised Deep Learning” The 18th IEEE International Conference on Dependable, Autonomic, and Secure Computing (DASC), Aug. 17-24, 2020.

[14] T. Kim, B. Kang, M. Rho, S. Sezer and E. G. Im, "A multimodal deep learning method for Android malware detection using various features", IEEE Trans. Inf. Forensics Security, vol. 14, no. 3, pp. 773-788, Mar. 2019.

[15] A. H. Lashkari, A. F. A. Kadir, L. Taheri and A. A. Ghorbani, “Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification”, In the proceedings of the 52nd IEEE International Carnahan Conference on Security Technology (ICCST), Montreal, Quebec, Canada, 2018.

[16] S. Miller and C. Busby-Earle, “The Role of Machine Learning in Botnet Detection”, 11th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 359–364), Infonomics Society, 2016.

[17] SC. Wang, Artificial Neural Network. In: Interdisciplinary Computing in Java Programming. The Springer International Series in Engineering and Computer Science, vol 743. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0377-4_5, 2003.

[18] R. Kohavi and G. John, “Wrappers for features subset selection. Artificial Intelligence” 97:273–324, 1997.