Academic City University College – Accra Ghana
Society for Multidisciplinary & Advanced Research Techniques (SMART) Africa
Tony Blair Institute for Global Change
FAIR Forward – Artificial Intelligence for All - Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Accra Bespoke Multidisciplinary Innovations Conference (ABMIC)

& The Africa AI Stakeholders' Summit 14th December, 2021

A Mult-Task System for Detecting and Classifying Malware Signatures Using Random Forest Classifier

Taylor, O.E., Ezekiel, P.S. & Emmah, V.T.
Department of Computer Science
Rivers State University
Port Harcourt, River State, Nigeria.
E-mails: taylor.onate@ust.edu.ng, ezekielpromise27@gmail.com, victor.emmah@ust.edu.ng

Proceedings Citation Format

Taylor, O.E., Ezekiel, P.S. & Emmah, V.T. (2021): A Mult-Task System for Detecting and Classifying Malware Signatures Using Random Forest Classifier. Proceedings of the Accra Bespoke Multidisciplinary Innovations Conference. University of Ghana/Academic City University College, Accra, Ghana. December 2021. Pp 73-84. www.isteams.net/ghanabespoke2021. DOI https://doi.org/ 10.22624/AIMS/ABMIC2021-V2-P6

A Mult-Task System for Detecting and Classifying Malware Signatures Using Random Forest Classifier

Taylor, O.E., Ezekiel, P.S. & Emmah, V.T.

ABSTRACT

The rapid increase in the use of information technology has made cyber-attacks a major concern in the use of internet by users globally. These attacks are carried out in different forms, some are carried out as phishing, man in the middle, malicious applications and so on. In this study we will focus on malware attack. Malicious applications have been a major challenge in the use of applications on windows operating system. These malicious attacks are being carried out in different forms. Some of these attacks are trojan, ransom, keylogger etc. The need to detect and classifier these malicious attacks in windows operating system is an important task. So therefore, this paper presents a smart system for detecting and classifying eight categories of malware attack on windows operating system using random forest classifier. The system starts by collecting signatures of malware attack on windows from Virus Share, Virus Sign and Github respiratory. The collected malware signatures went through the following stages of pre-processing (First stage, Second Stage, and Third Stage). The first stage has to do with creating a pandas. Dataframe using the malware signatures. The second stage has to with data cleaning and the third stage has to do with data transformation. The result of the Random Forest Classifier shows a promising performance in terms of accuracy, precision, f1-score, and recall. The result shows that the Random Forest Classifier has an accuracy of about 100% for each of the matrix evaluation.

Keywords- Malware signatures, Random Forest Classifier, Windows operating System, Matrix Evaluation

1. INTRODUCTION

The exponential expansion in malware attacks has become one of the significant ultimatum to Internet security. A new ultimatum report from Symantec shows records of thousands of malicious attacks on a regular schedule. The presence of malware in the Internet of things (IoT) and cell phones expanded. As indicated by the most recent report of threats from Kaspersky Lab in 2019, eliminate the quantity of clients that experienced Android malware dramatically multiplied to 1.7 million worldwide. Availability between an IoT devices and computers is set up through a cloud administration. The complex IoT equipment and programming conditions, gives more rooms for adversary attacks. As per the digital monetary threat report of 2019, the vast majority of the clients in China, Brazil, Vietnam, India, Russia, Germany, and the US were assaulted by banking malware. Nearly 889,452 clients of Kaspersky solutions on Lab were attacked by banking Trojans in 2018, an increment of 16% contrasted with 2017 when more than 767,000 clients were hit [1].

Analysis in malware is a quickly developing field requesting a lot of consideration in view of innovative improvement advances in interpersonal organizations, distributed computing, portable climate, smart lattice, Internet of Things (IoT) and Industrial Internet of Things (IIoT), and so on Most malware recognition frameworks depend on highlight vectors, which address the fundamental elements of malware. These component vectors separate into static examination and dynamic examination. Static examination works by dismantling the code of the malicious applications, without executing it.

Malware is perhaps the most significant threat in security on internet users. Malware can be said to be any sort of noxious codes that influence the respectability, secrecy and the usefulness of the computerized framework [2]. Malware functionalities are used in categorizing malware into various categories i.e., Malware functionalities are used in categorizing malware into various categories i.e., Backdoors, Trojans, Viruses, and Worms. These classes further gap into families based of the sort of variations. Malware programmers send numerous jumbling methods like dead-code addition, subroutine reordering, and code rendering, to make variations of a current malware family to dodge location. The most difficult aspect of web security is finding malware variations. The likenesses between numerous malware variations like Nuwar, Kelihos and Storm propose they were created by the equivalent malware coders [3].

Lately, machine learning has accomplished uncommon outcomes in the fields of vision and natural language processing and computer vision. Numerous scholars have likewise utilized Artificial intelligence techniques to address malware recognition and issues of classification. Despite the fact that analysis of malware techniques dependent on Artificial Intelligence have accomplished promising results, these strategies frequently require a great deal of time and assets in highlight designing. Malware perception is a significant part of malware examination. Basically, all current static survey techniques dependent on malware perception are gotten from grayscale pictures. Notwithstanding, a solitary low request include portrayal might be adverse to finding stowed away highlights in a malware family. For some datasets, particularly uneven datasets, existing order strategies don't generally function admirably for all families.

2. RELATED WORKS

In the paper [4], they carried out a survey on utilizing different techniques based on machine-learning in identifying and categorizing various classes of malware. The machine learning methods utilized are choice tree classifier, and neural network and multi support vector machine. Their exploratory outcome gave the following accuracy results 90.2%, 98.4%, and 99.33%. The research [5], considered the issue of malware recognition and grouping dependent on the analysis of images. They convert executable records to pictures and apply deep learning (DL) models in recognizing images. They also fine tune existing DL models that have been pre-prepared on large dataset that comprises of images. Their test result gave the 98.4% for deep learning model.

In the paper [6], a deep learning system in categorizing malware was proposed. The proposed model was prepared on a CNN-based engineering to detect malware on tests data. They transformed binaries of malware to grayscale pictures and therefore train a CNN in classifying the images. An experiment is being carried out on two malware grouping datasets, Malimg and Microsoft malware, exhibit that their technique accomplishes better result when compared to other algorithms. Their proposed method accomplishes 98.52% and 99.97% accuracy on the Malimg and Microsoft datasets separately.

The paper [7], introduced a gradual learning technique dependent on multiclass support vector machine (IMCSVM), which can further be develop by learning new malware test. Their proposed model can further be developed to categorize the nature of known malware classes by limiting the errors of the outcome with a better information to classify the various classes of malware. They also apply the gradual learning strategy into analysing malware, and their experimental results show the benefits and viability of their proposed model. In their paper [8], they proposed a system to identify and classify malware dependent on portrayals images of the binaries of malware. They utilized principal component analysis in extracting features of the malware class on a smaller subspace. They utilized support vector machine in classifying the extracted features. The outcome shows the accuracy of help vector machine on three dataset 0.998, 0.911 and 0.997 individually.

The paper [9], proposed a structure in characterizing eleven groups of malwares. This was accomplished by removing their unmistakable API columns from the report's scalable version of cuckoo sandbox. They applied diverse techniques in machine learning that will be used in grouping the malware families. Their experimental result shows that K-Nearest Neighbour had the most promising accuracy, which is around 95.8%. In the paper [10], a system in recognizing idea float in malware categorization models was proposed. The proposed system utilized a method called Transcend. Their proposed system was can utilized in a detecting the drift concept t on two contextual research on Android and Windows malware, raising a warning on malicious application. The paper [11], proposed a novel classifier to recognize variations of malware families and improve malware recognition utilizing CNN-based deep learning design. Their proposed system converts binaries of malware into colourful pictures that are utilized by the adjusted CNN design to distinguish and recognize malware families. Their experiment result has shown that the IMCFN stands apart among the deep learning models incorporating other CNN models with an accuracy of 98.82% in Malimg malware dataset and over 97.35% for IoT-android versatile dataset.

3. DESIGN METHODOLOGY

In this section we present the design methodologies

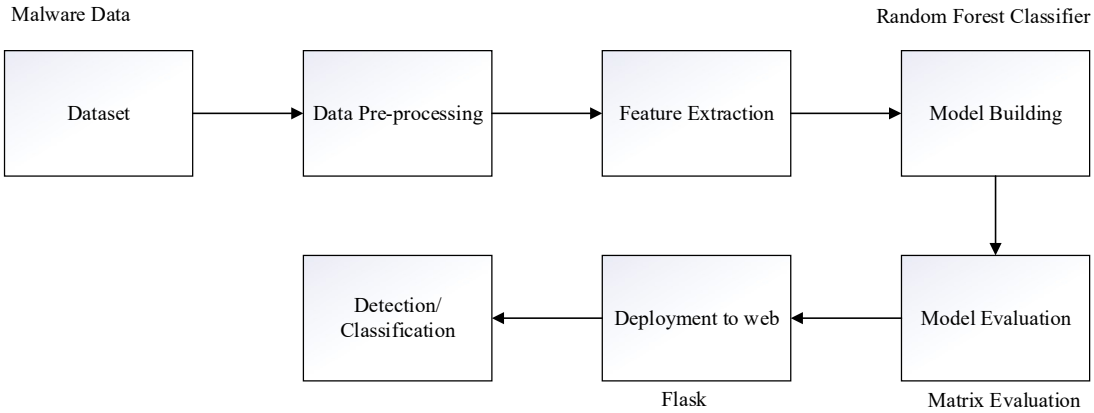

Figure 1: Architecture of the proposed system

Dataset: The dataset is a collection of different signatures of malware carried out on windows. The malware data was collected from Virus Share, Virus Sign, and Github respiratory. The collected malware data is made of 8 categories. The categories are backdoor, downloader, keylogger, miner, rouge, ransom, trojan and worm. The dataset sample can be seen in figure 2 below.

	Signatures	Category
0	0329c190b694c5ba92fcc4c7388d306f	backdoor
1	03b918c00c0689a272059fe340d49781	backdoor
2	0731b597e61c2fd74577239fc53c794b	backdoor
3	0cc1903c9931c6f102ae6a06aee5cc00	backdoor
4	0dd35f87b7bd22843ba334c1eb57fba2	backdoor
5	0e75c0e85710606d0a4caea65352c6f0	backdoor
6	101534723ab369c5ee0f73bedb2f3ae0	backdoor
7	10595f4fb22182d8fb3af855240ac7a0	backdoor
8	18798b6904059c9408888fa05da02fe0	backdoor
9	1cd12a8269d6ed7af46c6d82dbf0db28	backdoor
10	1d45b20988560b3322224bddc654e2a0	backdoor
11	1eb18d5802287167cd44b42de58b75e0	backdoor
12	2033a6d7d02690c31fa53d8717fc7ffb	backdoor
13	245f708cd7231ffcefc9096a81938ba0	backdoor
14	246135411475813c72a0c595232d7fd0	backdoor

Figure 2: Dataset of the first 15 rows.

The dataset shows the signatures of malware and the categories on which they fall on.

Data Pre-processing: The dataset was pre-processed by collecting the various malware signatures of each of the 8 categories and using the collected files in creating a csv file. The dataset was pre-processed by removing null values, and converting the data into arrays.

Feature Extraction: Feature extraction technique was used in selecting the most important features of the dataset. CountVectorizer function was used in selecting this important malware features.

Model Building: The model was trained using Random Forest Classifier. 80% of the malware data was used as input for the Random Forest Classifier, and 20% of the data was used for testing . In other to get a better training accuracy, we changed the number of estimators until will finding a better training result.

Mean Squared Error

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (fi - yi)^2 \quad (1)$$

Where
N represents the number of points in the data,
fi is the value returned by the model and
yi is the original value for data point i.

Classification Problem

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2 \quad (2)$$

Where pi represents the relative frequency of the malware categories and c represents the numbers of categories.

Model Evaluation: The model was evaluated using classification report, and confusion matrix. The confusion matrix shows the number of prediction results of the classification problem. It shows the summary of number of correct and incorrect prediction with a count value broken down by down. The confusion matrix is a technique for summarizing the performance of a classification algorithm. This is because classification accuracy alone can be misleading if an unequal number of observations in each class.

Deployment to web: The model was deployed to web for easy testing and execution. The web application was built using flask framework. Flask comprises of both HTML and CSS file embedded in it as class.

Classification: The web based application was used in detecting and categorizing the different malware class.

4. RESULT AND DISCUSSION

This paper presents a smart system for detecting and classifying eight categories of malware attack on windows operating system. The system starts by collecting signatures of malware attack on windows from Virus Share, Virus Sign and Github respiratory. The collected malware signatures undergone the following stages of pre-processing (First stage, Second Stage, and Third Stage). The first stage has to do with creating a pandas. Dataframe using the malware signatures. The was done by creating a pandas table for each of the eight categories of malware attack (backddor, downloader, keylogger, miner, rouge, ransom, trojan and worm) on windows, and finally appending the independent tables to be od one table so that we can reduced the memory space and carry out further pre-processing and analysis.

Figure 3 shows a chart of the eight categories of malware attack on windows operating system. The second stage has to do with the data cleaning. This was achieved by using pandas to check for rows that have some Nan or missing values, and drop such rows if it cannot be filled with another value. Finally, the third stage of the pre-processing has to do with the transformation of categories in numerical values. This was archived using LabelEncoder function in python to encode the eight categories to be of the form 0, 7. The can be seen in figure 4 and 5 below.

After these stages of pre-processing, we applied CountVectorizer function in selecting or extracting the most important features of the dataset. CountVectorizer was also used in transforming the various signatures to arrays which will be of the form 0,1. Finally the pre-proposed and extracted data was used as an input data in the Random Forest Classifier. The Random Forest Classifier was trained on a number of 1000 nodes. The proposed Random Forest Classifier achieved an accuracy result of about 100%. The model was evaluated using classification report and confusion matrix.

The classification report shows the performance of the model in terms of accuracy, precision, f1 score and the total number of predictions, whereas the confusion matrix shows the number of true predictions versus correct predictions. This was done for each of the categories of malware attack on windows. The result of the classification report can be seen in figure 6 and 7 below. After evaluation, the proposed model was saved and deployed to web using python flask framework. Figure 8 shows the homepage of the proposed system built on flask framework for easy detection and classification of malware attacks on windows. The result of the categorized result can be seen in figure 9.

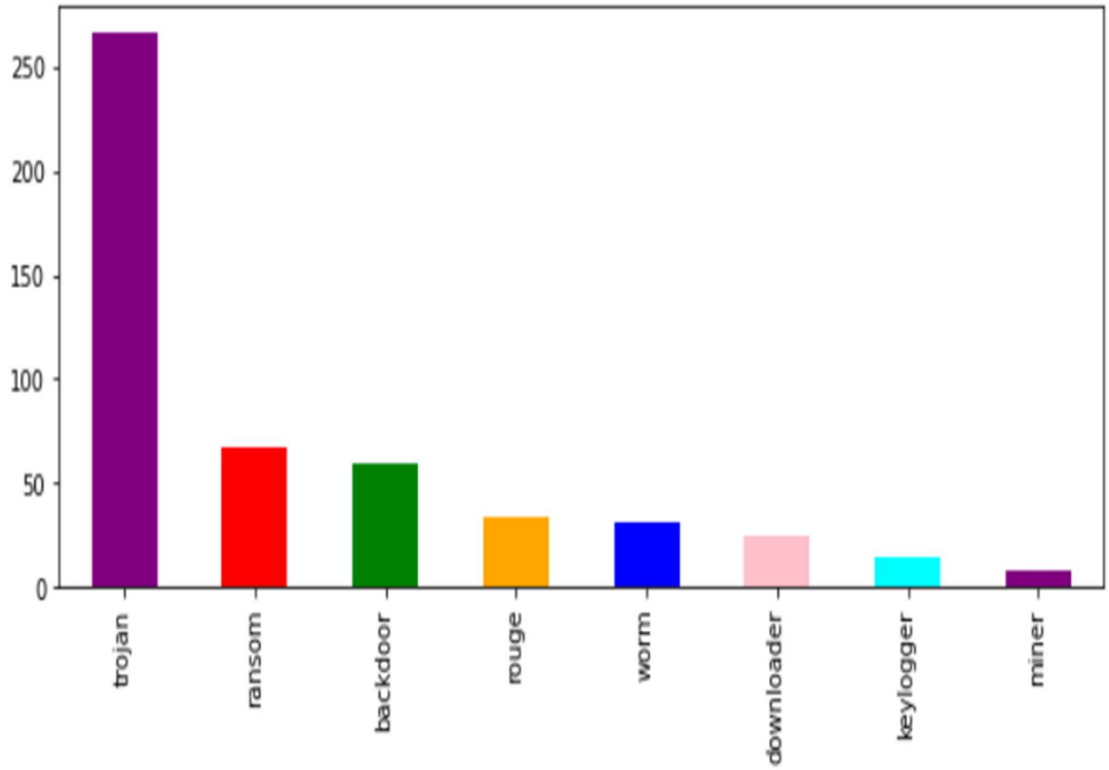

Figure 3: Histogram chart of the malware data

The chart above shows the number of times each of the malware category appears more on our training data. This shows that trojan attack appears more.

	Signatures	Category
0	0329c190b694c5ba92fcc4c7388d306f	0
1	03b918c00c0689a272059fe340d49781	0
2	0731b597e61c2fd74577239fc53c794b	0
3	0cc1903c9931c6f102ae6a06aee5cc00	0
4	0dd35f87b7bd22843ba334c1eb57fba2	0
5	0e75c0e85710606d0a4caea65352c6f0	0
6	101534723ab369c5ee0f73bedb2f3ae0	0
7	10595f4fb22182d8fb3af855240ac7a0	0
8	18798b6904059c9408888fa05da02fe0	0
9	1cd12a8269d6ed7af46c6d82dbf0db28	0
10	1d45b20988560b3322224bddc654e2a0	0
11	1eb18d5802287167cd44b42de58b75e0	0
12	2033a6d7d02690c31fa53d8717fc7ffb	0
13	245f708cd7231ffcefc9096a81938ba0	0
14	246135411475813c72a0c595232d7fd0	0

Figure 4: Training Data

Figure3 shows the signatures of the malware attacks on windows operating system. The signature columns shows the signatures of the eight classes of malware attack on windows. The category column the attack name. Here, zero represents a backdoor attack. The first 15 rows in the dataset is of backdoor attack.

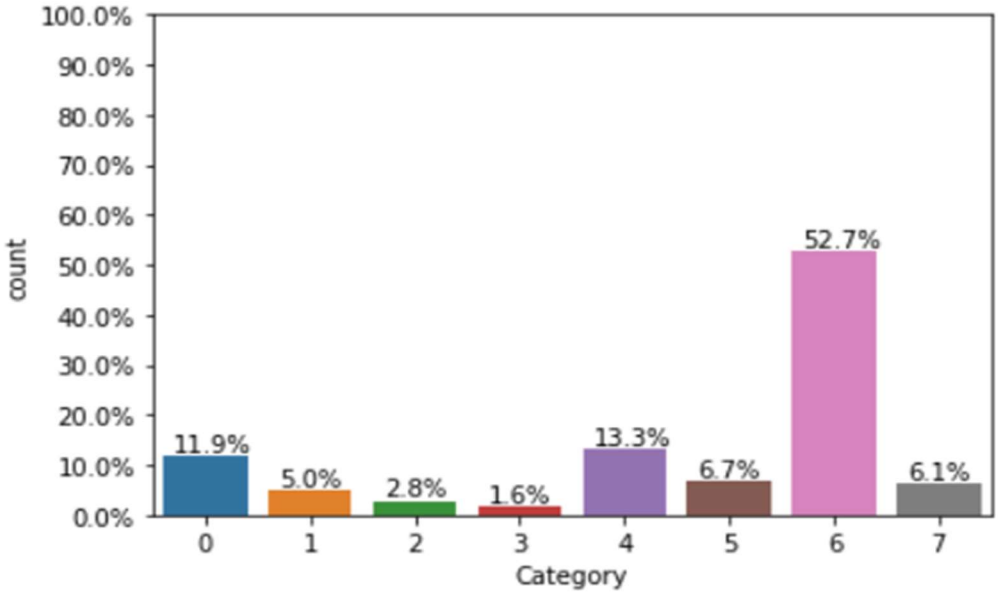

Figure 5: Malware statistics

This shows that the most attack that is being carried out is keylogger attack.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	49
1	1.00	1.00	1.00	13
2	1.00	1.00	1.00	11
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	59
5	1.00	1.00	1.00	30
6	1.00	1.00	1.00	217
7	1.00	1.00	1.00	19
accuracy			1.00	404
macro avg	1.00	1.00	1.00	404
weighted avg	1.00	1.00	1.00	404

Figure 6: classification report of the proposed model

The classification report shows the performance of the model in terms of accuracy, precision, recall f1-score and support for each of the eight categories. Here, our proposed model achieved an accuracy result of 100% approximately, same with precision, recall, and support score.

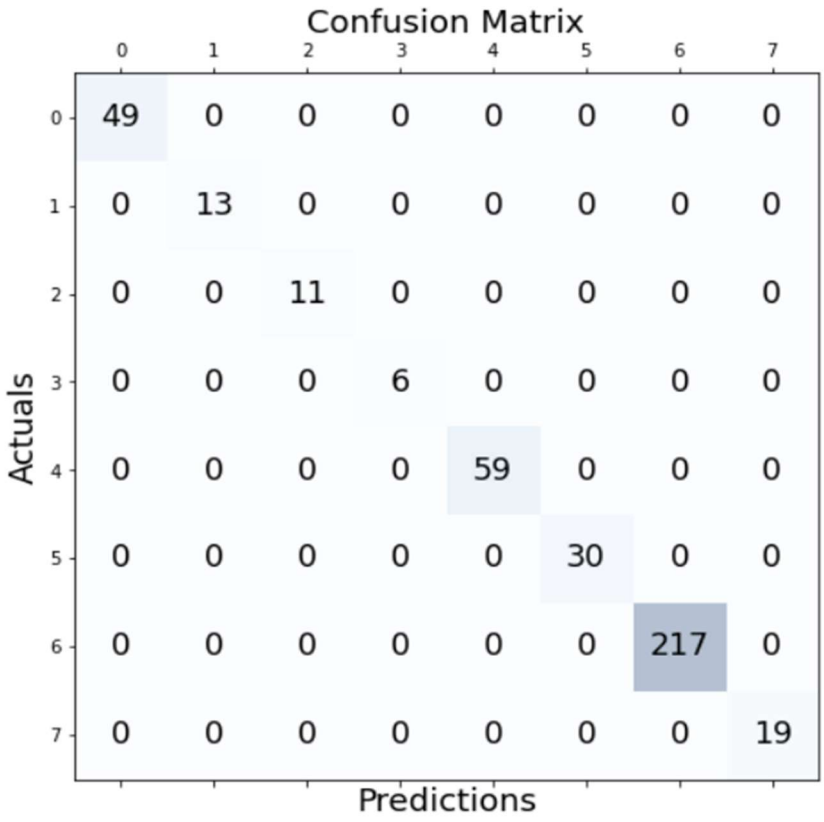

Figure 7: Confusion Matrix

The confusion matrix shows the number of true prediction vs correct prediction. Here, our proposed model predicted all the labels correctly. This shows that the number of true positive and true negative is about 100% whereas the false positive and negative are less than 1%.

Figure 8: web based system for multi class malware classification

Malware Signatures	Categorized Result
0cc1903c9931c6f102ae6a06aee5cc00	Backdoor Attack
3cf3cbc5067f1a59c6b8cffec761fc07	downloader attack
0fb1fe1b6c2e5856f95b2f6d49f954b6	Keylogger Attack
1ef7d145bf7153292ea33fe7c900ece9	Miner Attack
0a7b70efba0aa93d4bc0857b87ac2fcb	Ransom Attak
1Fxu7L83m1qDUM84fvsrQN3iwEjaxeRLEy	Rouge Attack
00dd8ac493443ee53c71863c6b2669b0	Trojan Attack
0a8a4950d9d71b448fde1f741608921e	Worm Attack

Figure 9: Result of the proposed system

Here, the system detected various malware signatures and classified them into their various classes.

5. CONCLUSION AND FUTURE WORK

With the increase in the use of technology, cyber-attacks have been a major concern in the use of internet by world users. These attacks are carried out in different forms; some are carried out as phishing, man in the middle, malicious applications and so on. In this study we will focus on malware attack. Malicious applications have been a major challenge in the use of applications on windows operating system. These malicious attacks are being perpetuated in different forms. Some of these attacks are trojan, ransom, keylogger etc.

The need to detect and classifier these malicious attacks in windows operating system is an important task. So therefore, this paper presents a smart system for not just detecting malware attack but by classifying them into their various classes. The system uses signatures of eight classes of malware attacks to train a machine learning model in detecting and classifying these attacks into their various classes. Random Forest Algorithm was used in training the machine learning algorithm. The result of the Random Forest Classifier shows a promising performance in terms of accuracy, precision, f1-score, and recall. The result shows that the Random Forest Classifier had an accuracy of about 100% for each of the matrix evaluation. This work can further be extended by including more classes of malware attack and also, to use a deep learning algorithm in building a system that will detect and classify malware on images or pdf documents. Due to other transformation of malicious applications. Some of these malicious attacks are being carried out on pictures and pdf documents, that if not detected, it will cause some malfunctions in the computer system. So, the need to detected malicious files in form of images and pdf documents are important.

REFERENCES

[1]. Symantec Corporation, Symantec internet security threat report, Netw. Secur. (2019) .

[2] J. Su , V. Danilo Vasconcellos , S. Prasad , S. Daniele , Y. Feng , K. Sakurai, “Lightweight classification of iot malware based on image recognition”, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 2, 2018, pp. 664–669.

[3] S. Venkatraman , M. Alazab, “Use of data visualisation for zero-day malware detection”, Secur. Commun. Networks 2018

[4] N. Udayakumar, V.J. Saglani, A. V. Gupta3, T. Subbulakshmi, “Malware Classification Using Machine Learning Algorithms”, Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018) IEEE Conference Record: # 42666; IEEE Xplore ISBN:978-1-5386-3570-4.

[5] N. Bhodia, P. Prajapati, F. D. Troia, M.Stamp, “Transfer Learning for Image-Based Malware Classification”, arXiv:1903.11551v1 [cs.LG] 21 Jan 2019.

[6] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, F. Iqbal, “Malware Classification with Deep Convolutional Neural Networks”, 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 1-5, 2018.

[7] J. Li , D. Xue , W. Wu, J. Wang, “Incremental Learning for Malware Classification in Small Datasets”, Security and Communication Networks, 1-12, 2020.

[8] L. Ghouti, M. Imam, “Malware Classification Using Compact Image Features and Multiclass Support Vector Machines”, IET Information Security, pp. 1–12, 2020.

[9] C. San, M. Mie, N. L. Htun, “Malicious Software Family Classification using Machine Learning Multi-class Classifiers”, vol. 481(1), 423-433, 2019.

[10] R. Jordaney, K. Sharad, S. K. Dash, Z. Wang, D. Papini, I. Nouretdinov, L. Cavallaro, “Transcend: Detecting Concept Drift in Malware Classification Models”, 26th USENIX Security Symposium, 625-642, 2017.

[11] D. Vasana, M. Alazab, S. Wassan, H. Naeem, B. Safaei, Q. Zheng, “IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture”, Computer Networks vol. 171, -12, 2020.