IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

Approach for Detection of PE Malwares using Ensemble Learning and Deep Learning

Priyanka Patil[1], Madhuri Gedam[2]

Department of Computer Engineering, Shree L.R.Tiwari College of Engineering Miraroad, India[1]

Department of Computer Engineering, Shree L.R.Tiwari College of Engineering, Miraroad, India[2]

Abstract: Security breaches are very common where the safety of the users is put to threat. Hence it is necessary that a threat to the system is identified which can be done with the help of malware detection. In order to explore, infect, steal data or virtually behave as the attacker wants with the help of a file or code delivered through a network is known as a malware. A PE malware typically is a malware code which is propagated through a PE file downloaded on the device which may result in loss of information and replacement of such malicious codes.

Such malware creators get away with it easily due to traditional methods of testing which are unreliable and time consuming. The current thesis aims to deploy a prototype that uses the concepts of feature extraction and use the Portable Executable file at a later stage. These features extracted are fed to algorithms based on ML (machine learning) and deep learning so that the overall system of the model is enhanced when the feature undergoes layers of neutral networks. The model undergoes pre-processing techniques which is then fed to algorithms for training.

Keywords: PE files, malware, machine learning, deep learning

I. INTRODUCTION

In recent history, security breaches caused by viruses, Trojans, ransomware and other malware have risen, and malware infections have made the news more than ever before. Almost every week there is a new technical issue that can be seen as the failure of the security community to manage and detect malicious content. According to ITU (International Telecommunication Union) statistics, by the end of 2020, two-thirds of the world's population will have an Internet connection.

All internet users are usually in a vulnerable state due to cybercriminals using various hacking methods. These methods are primarily used to target and disrupt the normal operation and execution of a software service on a computer system. These cybercriminals exploit various types of malware that lead to exploitation of basic services and damage the user's device. The malware used by such hackers is specially created by programmers and is used with the aim of damaging the user's computer system and getting personal profit from it. Such malware can be classified into different categories based on the function they intend to perform. Common viruses include worms, adware, spyware, and Trojans to implement. (1)

With the development of technology, an explosive growth of malware has been observed. This attack software is called malware and is usually code that is infused into a user's computer with the potential of harming and defaming the user. Without the consent of the user, this code is intentionally entered into his system. This is widely and publicly available on the Internet and is involved in a number of data breaches that a hacker can do; leaving the system of the user prone to attack. These malicious codes are executed in a hidden form which the user trusts, allowing access to personal information and resources on the system of the user. Some individuals also produce such malware sheer out of their own interest and try to destroy the system of the user further with such malicious codes. Such codes spread to other systems very fast...

The most shocking malware was discovered in 2016; in which the malware was named "Mirai malware" (2), which was able to destroy user systems using botnets. These botnets can access 300,000 Internet of Things devices and conduct Distributed Denial of Service (D-DoS) attacks. After one year, an attack similar to it was found to be carried out on a Windows operating system infected with the Wannacry ransomware (3)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

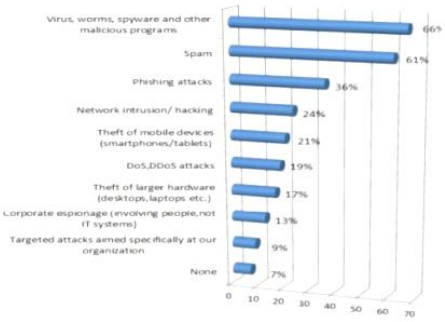


Figure 1: Malware attacks on system (3)

The reason for this attack is to infuse such malicious codes in the system of the user and get gains from them for code decryption. After getting the gains from users, the hacker hard drives to the original stage by decryption of the system. The existing devices cannot detect invisible malware; hence, decrypting such malware becomes quite a tedious task for developers. Therefore, since 2011, ML (machine learning) and Deep Learning techniques have been used for detection of such activities. As the working system of the nature being complicated in nature, the process of encrypting and securing the files have become a very crucial job for the software developers as the system of the user gets exposed to violations. The prime goal of the malware attacker is to contravene the user system and gain crucial information of the valuable assets of the user for using it against them mainly for defamation thereby creating a very stringent environment for the user.

II. RELATED WORKS

One of the prime objectives of the thesis is to develop a systematic model which can identify as well as classify the evaluated Portable Executable files as malignant or benign. To do this, extensive literature review of existing works has been conducted so that the general operation of the malicious system can be understood. Study of such literature revealed that several researchers were engaged in their own research to fix such system breaches by detecting such malware. This section highlights research based on machine learning to detect PE files. Eskin, Schultz et.al (4) devoted their work to developing an automated framework that can detect malicious PE files using static features such as PE Header and String. To do this successfully, they used the malware dataset and divided it into two more subsets, which were majorly classified as training and testing subsets. A classification was performed during the training phase followed by the development of a model which can separate malware and good software files from the dataset. Instead, the test phase used classifiers to compute the unknown binaries. In the later stages, the author proposed to run the model using three machine learning based algorithms such as NB, multiple NB and Ripper. They ran these algorithms on a dataset containing a total of 4,266 samples, of which 1,000 were considered benign. The author achieved an overall accuracy of 97.1% with his proposed system. Later researchers found that the accuracy achieved was relatively higher compared to existing techniques; such as signature-based authentication systems.

Rathore & Agarwal(5) explores Deep Learning and multiple machine learning algorithms for malware detection. In addition, other alternative techniques were used to build these models which included cross-validation, and solution to the issues were sought to deal with un-balanced data issues. In both supervised as well as non-supervised learning, malware classification was done by a vector using an opcode frequency feature. The resulting dimensionality was reduced using characteristic reduction techniques such as a Single-Layer auto-encoder and a three-layer stacked auto-encoder. A DNN that is Deep Neural Network and Random Forest (RF) algorithm were then used to identify the Portable Executable files in the user system. The performance of RF algorithm was better than DNN models as per the findings but the results suggested that for malware detection, Deep Learning might not be useful.

In another research paper (6) , the author developed a heuristic recognition model based on the metamorphic encryption technique and used static features for analysis. These static properties contained opcodes and API files that were extracted using IDA pro. In the next step, all static functions were extracted using the same opcode. To achieve higher accuracy, the author proposed to implement the exact concept using machine learning-based six algorithms, including Random Forest (RF) algorithm and NB algorithm. The author arrived at the conclusion that the overall improvement of the system model directly depended on the algorithm classification which was used to detect malware in the system. This model achieved bias accuracy with a higher percentage of precision and recall factors.

DOI: 10.17148/IJARCCE.2023.124124

Ucci and Aniello (7), used static analysis to extract sequences from Portable Executable file headers to propose their work. The author applied it to a dataset containing four thousand seven hundred and eighty three samples and was able to classify them based on classification algorithms. He used random forest as a classification algorithm and showed that it gives ninety-six percent accuracy. However, the work with this model was later extended to a multi-domain study and was faster in the computation compared to other observation system. In another work, (8) the author proposed a another detection method which would classify several malware variants. He proposed to achieve this by divining signatures. However, his research emphasized analyzing the static properties of a PE file, which involved extracting strings. N-grams and API calls. In later stages, the author divided the material into stages of training, testing and cross-validation. In the training phase, he applied the concepts of unsupervised learning-based machine learning algorithms and tested the system model against certain hyper parameters.

In 2017, the authors (9) presented a Markov model to compare detection and classification results obtained from dynamic and static analysis. The author suggested applying this model to different versions of the malware family. However, his research provided a wide research area for the application of discovery methods that were also based on data mining techniques. His research also included the study of different characteristic dimensions using classification and grouping algorithms.

A similar work was followed (10) where the author proposed solutions for malware detection using hybrid methods. In later stages, data mining algorithms were used to classify malware as benign and good. (11) Evaluate three types of malware detection methods: signature-based, behaviour-based and heuristics.

H. Zhang (12) conducted a similar experiment using two mechanisms out of the three mechanisms. Further, the study did not consider data mining or machine learning initiatives to detect and classify malicious files. Younas (13) provided an overview of malware detection and analysis methods. They examine detection methods using specific sources from two perspectives: feature extraction and clustering or classification. Malware analysis frameworks based on data mining could be used to improve malware detection accuracy by reduction of false positives, according to the report's findings. The use of malware research methods was strongly influenced not only by the different classifiers used, but also by the features extracted from these observations. They also tried to argue that collecting classification patterns instead of just class labels can improve detection accuracy and that training requires an equal distribution of malicious and benign files.

III. OVERVIEW OF METHODOLOGIES USED

This section of the research study briefs on the methodologies used to implement the proposed system.

Method Proposed for Use

The current study uses a combination of the concepts of ML (machine learning) and Deep Learning which classifies model capable to learn and understand from it's past experiences as well as to make predictions. Next after the implementation of this model, the functioning of this supervised model takes place. The model therefor makes use of mapping functions to generate output variables.

Workflow of the proposed system is as below:

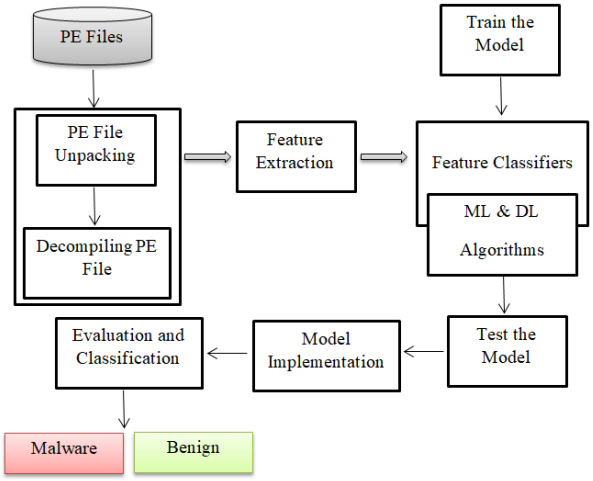


Figure 2: Workflow of the proposed system

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023
DOI: 10.17148/IJARCCE.2023.124124

The algorithms used in the study are machine learning and deep learning

Machine Learning Algorithms:

1. Adaboost: An acronym for adaptive boosting, it is used by assigning heavy weights in order to improve the implementation and operation of an algorithm. Each operation during the training phase is assigned a weight. Weak origin algorithms are given booster weights so that they are strengthened iteratively. Steps involved with Adaboost are as below:

- Initializing specific weights to respective training points
- Each classifier calculates the error rate by voting for the weakest classifier. The final step involves updating of weights at every training point and the malware file is then used for the purpose of classifying. This procedure is repeated until an optimum accuracy and least error is gained.

2. Random Forest: Random forest algorithm is a set of decision trees which act as an ensemble algorithm. At each node of the decision tree, a choice must be made, where the random forest algorithm helps make a prediction with that choice. The earliest decision is made in the root leaf node, or whether the given data in the leaf node is red or blue, and this decision determines the decision making execution order. The second option is dependent on whether to underline the data numbers on the leaf nod if they are red.

3. Gradient Boosting: A well-known approach utilised in the application of classification and regression issues is the gradient boosting algorithm. The overall weight of all the classifiers employed in the algorithm is calculated via boosting. This approach has been successfully implemented using a collection of weak estimating methods that are frequently employed in decision trees. Gradient Boosting combines a number of weak classifiers into a single, much stronger and larger data set, improving the performance of the prediction model even further. Its main goal is to lower the loss function, provided that the loss function is differentiable.

4. Ensemble Learning: This method creates a set of first-stage classifiers to train the data in different ways. In addition, a fresh overall classification model is developed using all the decisions of these first-stage classifiers. Finally, the model learns to label and predict objects using this stacking method. Random forest and Adaboost are used as two group learning classifiers in this thesis implementation.

Deep Learning Algorithms:

1. CNN: Following the concept of Supervised Learning, the CNN model receives input in the form of images and for the purpose of further classification, it further extracts them. (14)CNN consists of four layers:

A convolutional layer in the first layer is in charge of spotting the existence of visual features. Matching templates and dragging a window that depicts an image are used to complete the process until a feature map is obtained.

The second layer is the activation layer, which improves a network's training during the back propagation process by introducing nonlinearity into the dot product.

The third layer is the pooling layer in which the output matrix obtained is reduced to a smaller size

The fully connected layer, which is the last layer, utilises an activation function to apply relevant labels to images after receiving the yield layer from the layer above.

LSTM:

The fundamental purpose of the LSTM, an extension of the RNN model, is to retain memory data and previously created storage. Its applicability is therefore ideal for resolving time-related issues. The LSTM's operating principle is as follows:

it has an internal memory cell made up of three components: an input port, an output port, and a forget port. The vanishing gradient issue can also be solved by this LSTM construction.

When using LSTM, follow these instructions to find malware in a Portable Executable file.

- The initial step starts with the raw data obtained by extracting the header files
- In the next step, the headings are processed using an LSTM network
- In the last step, the classification of invisible files is tested

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

IV. IMPLEMENTATION DETAILS

The area involves steps involved in implementation of data. Repository data is collected and used for training as well as testing. One of the main languages used in implementation is Python. Post implementation, the prototype is tested and classification of the files is done as good ware or malware.

A. Dataset Used

Kaggle and Virus Share Repository are used for collection of data. In order to construct and develop a clean dataset from the raw dataset, Portable Executable were acquired from.exe and.dll repositories.. The files tested entity were labelled as "malware detection" after the train and test set. These labels were then transformed into numeric type and utilised to build a confusion matrix. The dataset utilised is divided into two halves, 80:20, with 80 percent used for training and the remaining 20 percent used for When using LSTM, follow these instructions to find malware in a PE file used for testing purpose. The training and testing processes are then integrated to generate a single database after all missing values have been found.

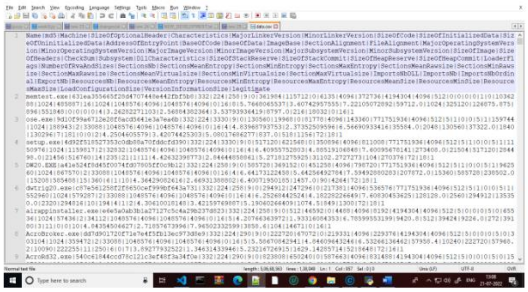


Figure 3: Workflow of the proposed methodology

B. Data Pre-Processing

Post the beginning of the implementation, data collected from various repositories is verified if it can be further used for the next step or not. This is performed to avoid over time consumption for a single task. At this stage all the irrelevant data is discarded so that the performance is not hampered. Hence this is known as data pre-processing which is performed to filter out all the redundant data.

Various methods used for pre=processing of data is as below:

- Principal Component Analysis (PCA) is a technique used to decrease the input file's overall dimensionality in order to prevent the issue of over-fitting. A dataset is rotated around a predetermined axis to create linear combinations of the old data.
- Estimators: Also referred to as standardisation, these are in charge of gathering distinct features from a dataset and transforming them into a distribution function using a scalar number.
- In order to achieve optimal outcomes, unstructured raw data is transformed into meaningful data by removing the unhelpful information.
- Data cleaning is done to remove irregularities from a raw data set.

C. Feature Extraction

The Portable Executable input files are later removed to produce only the necessary features to finish the whole malware detection procedure after the data processing stage is finished. Implementing the built-in Python tool pefile allows for this extraction. The three main headers used during the function extraction process are the DOS header, job header, and file header. Figure below provides an overview of the Portable Executable file.:

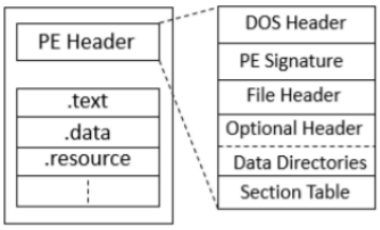


Figure 4: Overview of PE Header

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

D. Feature Selection:

For the purpose of classification, feature selection takes place after the data has been cleaned and filtered. At this point, the final folder configuration size for all dataset samples was set to 0, and the three PE heads with the most inconsistent properties were removed. Finally, all relevant features were selected and entered for training and testing.

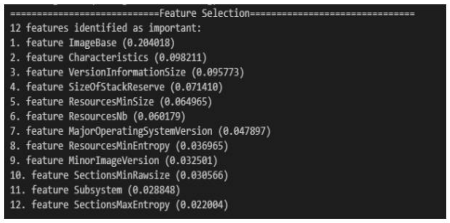


Figure 5: Selected features of the model

E. Data Split

Once the process of visualizing the dataset is observed; the system model thus undergoes the process of data split; wherein the dataset is split into ratios of training and testing. For the purpose of implementation of the proposed thesis; the ratio is divided to be as 80 percent and 20 percent respectively. After this split; the system model is then sent for testing purpose on machine learning algorithms.

V. EXPERIMENTAL ANALYSIS AND RESULTS

This section of the study highlights on the parameters that are used for evaluating the overall performance of the system model and further briefs on the results thus obtained.

A. Parameters of Evaluation

Following are the parameters used for evaluating the proposed work:

Parameter	Definition
Accuracy	how many samples were correctly classified
False Positive (FP	how many samples were classified as malware but were not
False Negative (FN)	how many samples were classified as benign software but was not
True Positive (TP)	how many samples were classified as malware and was
True Negative (TN)	how many samples were classified as benign software and was
Recall	how many positive samples were identified correctly
Precision	how many of the malware samples were identified correctly
F-Measure	uses the harmonic mean between recall and precision
Receiver Operating Characteristics (ROC)	a figure describing the relation between true positive rate and false positive rate

Figure 6: Parameters for evaluation of proposed work

- Confusion Matrix: It is a visual representation of the values that have been obtained, and it can be further displayed by contrasting the actual values with the predicted values
- Classification Table: A classification table includes information on the accuracy that was created, as well as the results of precision, recall, and F1-factor calculations.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

B. Testing Results

Following are the results obtained from machine learning algorithms and deep learning algorithms obtained by using confusion matrix:

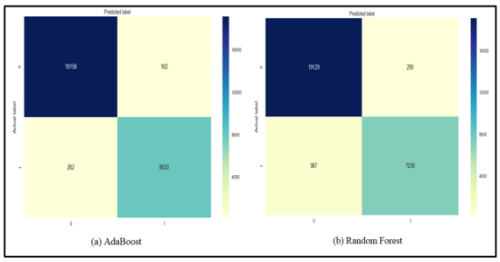
Machine Learning Algorithm:

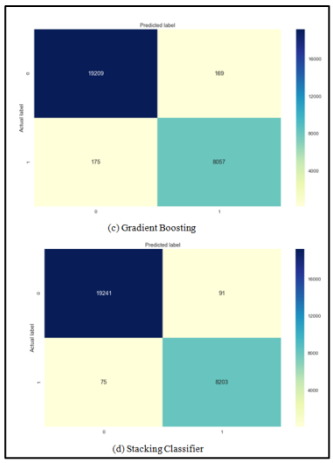
The main goal of the study is to gather malicious files from a collection of datasets and then classify them as malicious or benign. Hence three algorithms and one stacking algorithm is used which can be described from the table below:

Algorithm Used	Train/Test	Values	Accuracy
AdaBoost	Test Set	[19156,162] [262,8030]	98.46
Random Forest	Test Set	[19129,256] [987,7238]	95.49
Gradient Boosting	Test Set	[19209,169] [175,8057]	98.75
Stacking	Test Set	[19241,91] [75,8203]	**99.39**

The visual representation of the confusion matrix is as below:

;





IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

The table below displays the accuracy numbers gleaned from classification reports:

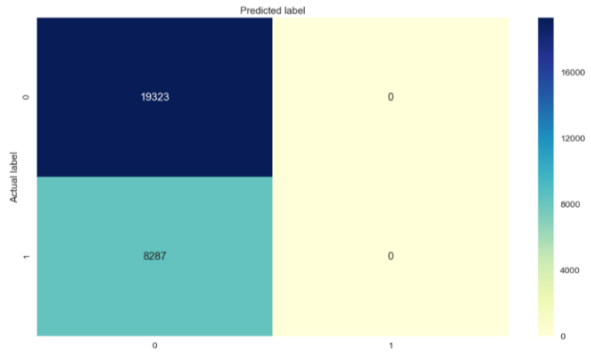
Algorithm	Precision		Recall		FI-Score	
	Positive Cases	Negative Cases	Positive Cases	Negative Cases	Positive Cases	Negative Cases
AdaBoost	0.987	0.980	0.992	0.968	0.989	0.974
Accuracy	**0.98**					
Random Forest	0.889	0.933	0.978	0.712	0.931	0.808
Accuracy	**0.99**					
Gradient Boosting	0.991	0.979	0.991	0.979	0.991	0.979
	0.99					
Stacking	0.996	0.989	0.995	0.991	0.996	0.990
	0.99					

Deep Learning Algorithm:

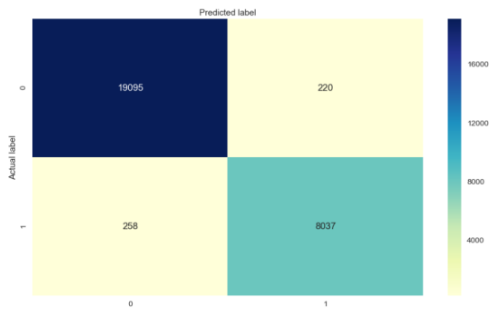
To accomplish the goal of the thesis, the deep learning algorithm employs two sets of algorithms. Following is a description of the confusion matrix's values:

Algorithm Used	Train/ Test	Values	Accuracy
CNN	Test Set	[19095, 0] [8287,0]	69.98
CNN + LSTM	Test Set	[19095,220] [258,8037]	**98.62**

The visual representation of CNN is as below:



The visual representation of CNN plus LSTM is as below:



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

Graphical User Interface:

The project model deployed through GUI can be described with the help of the snippets below:

Detection of malware software

Implementation of Deep Learning Algorithm

VI. CONCLUSIONS

The thesis's main objective is to carry out research and use machine learning and deep learning algorithms to discern between excellent and benign items when it comes to dangerous Portable Executable files. The dataset is trained and evaluated using 80% and 20% of the original data extracted from the archive, respectively, and the proper features are chosen for attribute categorization. Later, by gathering 138047 PE files and extracting attributes from them to pass via the computer process, the thesis was successfully put into practise. Future deep learning and machine learning algorithms will enable evaluation and deployment of the same model. Because it enables complete automation while employing fewer functions and obtaining high accuracy, this may be advantageous to the model.

REFERENCES

[1] The World of Malware: An Overview. Nmanya, A.P., et al., et al. Barcelona, Spain : s.n., 2018. IEEE 6th International Conference on Future Internet of Things and Cloud. pp. 420-427.

[2] Undertsanding the Mirai Botnet. April, T. and Antonakakis, M. Van Couver, Canada : 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, 2017. pp. 1093-1110.

[3] Wnnacry, cybersecurity and health information technology: A time to act. Ehrenfeld, J.M. 7, s.l. : Journal of Medical Systems, 2017, Vol. 41, p. 104.

[4] Data mining methods for detection of new malicious executibles. Schultz, M.G., et al., et al. s.l. : Proceedings 2001 IEEE Symposium on Security and Privacy, 2001. pp. 38-49. S&P(2001); IEEE 2001.

[5] Malware Detection using MAchine Learning and Deep Learning. Rathore, H., et al., et al. Goa : s.n., 2019. International Conference of Big Data Analytics. pp. 402-411.

[6] Khodamoradi, P., et al., et al. Heuristic metamorphic malware detection based on statistics of assembly instructions using classification algorithms. CSI International Symposium on Computer Architecture and Digital Systems (CADS). s.l. : IEEE. pp. 1-6.

[7] Ucci, D., Aniello, L. and Baldoni, R. Survey of Machine Learning Techniques for Malware Analysis. 2018.

[8] Gandotra, E., Bansal, D. and Sofat, S. Zero-Day Malware Detection. s.l. : Sixth International Symposium Embedded Computing and Designing System,IEEE, 2016. pp. 171-175

[9] Damodaran, A., et al., et al. Comparison of Static, Dynamic and Hybrid Analysis for Malware Detection. 2017. pp. 1-12.

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124124

[10] Mirza, Q.K.A., Awan, I. and Younas, M. Cloud intell: An Intelligent Malware Detection. 2018. pp. 1042-1053, Fut. Gen. Computer System.

[11] Zhang, Y. Li. Z, et al., et al. Spectral-based directed graph network for malware detection. s.l. : IEEE, 2020.

[12] Zhang, H., et al., et al. MALDC: A depth detection method for malware based on behavior chains. s.l. : Proc. World Wide Web, 2019. p. 1 to 20.

[13] Early Stage Malware Prediction Using Recurrent Neutral Networks. Rhode, M., Burnap, P. and Jones, K. s.l. : Computer Security, 2018, Vol. 77, pp. 578-594.

[14] SVM training phase reducting using dataset feature filtering for Malware Detection. O'Kane, P., Sezer, S. and Mclaughlin, K. s.l. : IEEE, 2013, Vol. 8(3), pp. 500-509.