Machine Learning Based Novel Framework for Malware Detection

Sonali K, Vijayshri K, Shwetambari C, Ujwala K.

Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale, Pune, Maharashtra, India

Email sonali.kothari@sitpune.edu.in,

Abstract

Recent technology developments in computer systems change human life from real to virtual worlds. Cyber criminals' attention has transferred from a real to a virtual realm as well. This is because it is easier to conduct crimes on the computer than in real life. Malicious software (malware) is unwanted software that is often used by hackers to launch cyber-attacks. Malware versions continue to grow by using advanced obfuscation as well as packaging approaches. Novel approaches which are entirely distinct from traditional ways must be applied to effectively tackle emerging-malware strains. Traditional AI (Artificial Intelligence) approaches, notably ML (Machine Learning), are unsuccessful in recognizing all new & complicated malware variants. DL (Deep Learning) approach which is considered separate from normal ML techniques may be a potential solution to the difficulty of recognizing all sorts of malware. In this research, the fusion model proposed depends upon deep learning and machine learning technique that must classify malware variants. The main influence of this work is to propose new fusion architecture which has integrated a custom DNN model and machine learning-based random forest classifier in an optimized way It is planned to test the proposed method using the Microsoft BIG 2015 dataset. Based on the experimental findings, it has been predicted that the proposed technique has useful in classifying malware with high accuracy and that it has surpassed the existing method

Keywords. Malware; Malware Classification; Malware Detection; Malware Variants; Deep Neural Networks; Deep Learning; Machine Learning.

1. INTRODUCTION

Today, it is feasible to conduct all of one's social interactions, financial transactions, and physical measurements over the Internet. All of these advancements entice cybercriminals to conduct crimes online rather than in the real world, and they've been effective. Cyberattacks cost the global economy billions of dollars, according to current academic and commercial studies. [1][2]. Malware [3][4]is often used by cybercriminals to initiate cyberattacks. Presently, no. of illegitimate and criminal applications is rapidly increasing. Many of these apps are malicious software designed to aid in the expansion of the organization. Criminals make use of malware to gain control of computers, steal personal and secret information, and utilize this information in some other way to make a profit. Malware forensics has become an integral aspect of computer forensics in today's world [5][6]. An example of malware would be a piece of software that causes harm to a victim's computer. Viruses, worms, trojans, ransomware, & rootkits are just a few examples of malicious software. Distributed Denial of Service (DDoS) assaults may be initiated by malware types, as well as the resulting disruption to computer systems can be devastating.

To stay undetected in the victim's system, malware sample types use tactics such as encrypting as well as packaging [7]. Research spread by leveraging human trust as an infectious vector. There are several ways in which malware may propagate, such as opening email attachments, installing fraudulent apps, and viewing and downloading files from false websites[8]. Proposed work is focused in identifying malicious software as soon as it is downloaded on the computer systems to keep system safe. When a suspicious file is examined, malware detection is used to determine if it is malicious or not. Once the file has been determined to be malicious, the classification of the malware may be determined. However, even though both static & dynamic methodologies may be employed for analysis, various ML also have been developed to automate classification phases & malware analysis to limit the number of samples needing close human study. The difficulty of determining if a file is malicious or not is classified as a problem of classification. The virus is classified using a variety of ML methods [10][11], including SVM (Support Vector Machine), DTs (Decision Trees), and NB (Naive Bayes), [12] among others. In this ML method, the dataset is often comprised of files, with the label indicating if a file is malicious or not malicious. This dataset has been classified into two parts i.e., training & testing, respectively. The training dataset is utilized for constructing a specific model for a given task based on the information in the dataset. It is required to use a cross-validation approach to enhance the evaluation of the model. The model is used to test the dataset once it has been trained on the training dataset. In this case, the model is completely oblivious to the labels. To determine the label for each file, it makes informed estimates about the label. The accuracy of the classification is then determined by counting the number of files that were correctly categorized.

2. LITERATURE REVIEW

According to [12], a new deep-learning-based architecture may be developed to identify malware variants using a hybrid model. Researchers have proposed a novel hybrid architecture in which 2 wide-ranging pre-trained network systems are seamlessly integrated. Four phases are involved in the construction & training of the suggested DNN architecture, as well as testing and evaluating a trained Deep Neural Network. The suggested approach was evaluated on datasets from Malimg, Malevich, as well as Microsoft BIG 2015.DL methods [13] based on RNNs are examined in this study for the ability to detect malware in cloud VMs. Research focuses on LSTMs and Bidirectional RNNs, two significant RNN designs (BIDIs). For this study, the authors used a pool of 40,680 harmful & benign samples to test a hypothesis. Malware operating in an open online cloud environment with no constraints was used to gather the process-level characteristics. This mimics realistic cloud provider conditions & captures the genuine behavior of stealth and sophisticated malware. Different assessment measures provide detection rates of more than 98% for LSTM & BIDI models. To fully grasp the significance of input data representations, an analysis study is also conducted. In some cases, the order of the inputs does have an impact on the efficiency of the trained RNN models, according to research findings. A novel [14,15] methodology for detecting malware in real time and adapting to new characteristics was suggested in this paper. API-Pair was accepted as a new dataset & trained using the Maximum Entropy model, which was able to achieve both weighting & adaptive learning goals at once. After that, a clustering technique was used to eliminate characteristics that were unconnected or ambiguous. An LSTM -based detector was also developed to provide real-time detection.

[16] provides an ML framework for finding & detecting DGA domains to lessen the danger. For a year, they gathered real-time threat data from traffic. A 2-level model & prediction model is part of the suggested ML framework. As part of a two-tiered paradigm, the author first identifies the algorithms responsible for generating DGA domains, & then utilizes the clustering technique to identify them. The hidden Markov model is used to build a time-

series model that predicts incoming domain information (HMM). As a result, created a DNN model to help the suggested ML framework deal with the massive dataset steadily gathered over time. Outcomes from rigorous testing show that the suggested system & DNN model are accurate.

3. PROPOSED METHODOLOGY

Proposed method has offered a fusion model of DL & ML-based architecture for malware classification illustrated in fig.1 that have used in conjunction with other studies. Malware data will be acquired from the Microsoft BIG 2015 dataset for use in the proposed methodology. To process malware samples further, research has first been transformed into grayscale images also then transmitted to a deep learning system for processing. Once the image acquisition phase is done the proposed approach would extract high-level malware features from malware pictures via convolution layers of the proprietary architecture of DNN, as well as the system trained in a supervised manner after the image acquisition section is accomplished. At the end of the process, a classification using machine learning-based random forest classifiers have employed to categorize malware variants. In figure 1 all processes of research first data collection, that is Microsoft BIG 2015 dataset, after this data preprocessing which is called data acquisition, converts the collected data grayscale image. This collected dataset is split into train, test, and validate. For the feature extraction purpose CNN model is used and classification used machine learning approaches. after the trained model calculate the performance of the proposed model. Finally, get predicated results.

- Data Preprocessing: Data preprocessing is the procedure of digitizing information from the world around so it may be processed, presented, & stored in a computer. This project aims to create a grayscale picture of binary files. It has read in a vector of eight-bit unsigned numbers from the beginning of the malware binary file. This has been followed by the conversion of the binary value of each component into its decimal counterpart. After being transformed into a 2D matrix, the resultant decimal vector is translated into an image in grayscale. The size of the malware binary file has a significant impact on the width & height of the 2D matrix.
- Feature Extraction by DNN [17]: Extracting as much information as possible from the available data sets is crucial to creating an effective solution. The task of deconstructing the features learned by a custom DNN.
- Custom Deep neural network: Deep neural networks (DNN) were used for a variety of tasks, and the substantial gains in performance gained using DNN for those tasks led to the use of DNN for the picture classification task. To extract features, we made use of a DNN variation called a Deep Convolutional Neural Network (DCNN)[18][19].

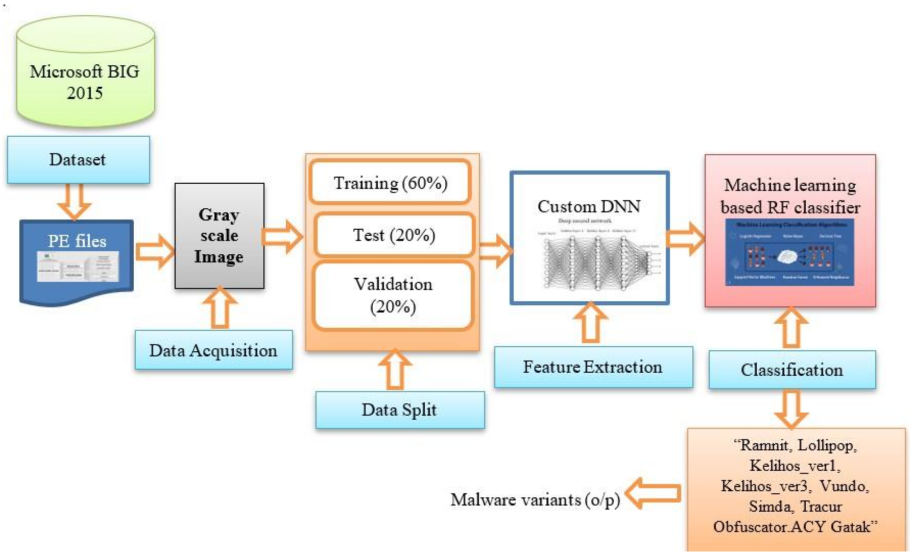


Figure 1. Block architecture of proposed methodology

- Oversampling using SMOTE: The dataset must be balanced to improve the minority class's forecast accuracy. The class imbalance issue is addressed using SMOTE (Synthetic Minority Oversampling Technique)[20]. SMOTE uses feature space to generate synthetic samples of the minority class. Synthetic instances are generated by first multiplying the difference in feature vectors between the minority class instance and its closest neighbor by a random value between 0 and 1. To create a synthetic instance of the minority class, the outcome of multiplying the feature vector is added to it.. Consider the fact that the feature vector of the under-investigation minority class sample is denoted by fi, and that $fnear$ is K-nearest neighbors of fi. Newly created synthetic sample, denoted by f_{new}, may be symbolized by the equation. (1).

$$f_{new} = f_i + (f_i - f_{near}) \times R \qquad (1)$$

Here, R is a random no. among 0 & 1.

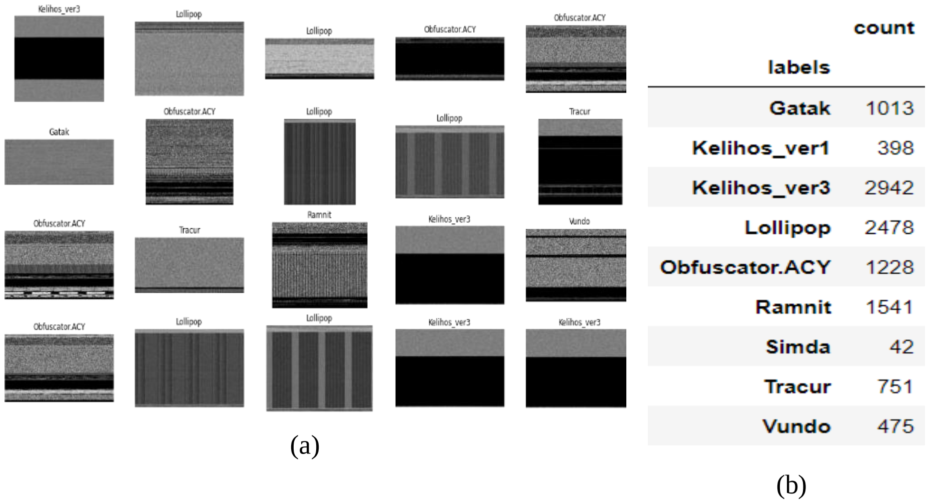
- Random Forest Classifier (RFC): Random forests are a kind of machine learning regression approach for classification. This method is driven by assembling web data into a multitude of decision trees (DTs) during the training phase, also then outputting class i.e., mode of classes that are produced by individual trees. It is the most accurate algorithm currently in use.

4. RESULTS AND DISCUSSION

This section describes the dataset in use for malware detection, and its visualization, and highlights the present research work's analysis of trials. This study used the Microsoft BIG 2015 dataset to apply the methods suggested in this study in python 3.0. Multiple performance metrics are used to assess the outcome.

Dataset Description

Microsoft BIG 2015 is being used to acquire malware data in the beginning. This year's Microsoft BIG 2015 dataset[1] involves 10868 malware samples divided into nine separate classes, which are called Gatak, Kelihos_ver1, Kelihos_ver3, Lollipop, and Obfuscator.ACY, Ramnit, Simda, Traceur, and Vundo. The many kinds of malware are represented by respective grayscale graphics in Figure 2 (a). (The dataset known as Microsoft BIG 2015 is where these photographs originated). Figure 2 (b) shows the count of images of the malware dataset with each category. The dataset contains nine categories of data. The gatka count is 1013, lollipop count is 2878 similarly other labels contain count values of data.


Kelihos_ver3
Lollipop
Lollipop
Obfuscator.ACY
Obfuscator.ACY
Gatak
Obfuscator.ACY
Lollipop
Lollipop
Tracur
Obfuscator.ACY
Tracur
Ramnit
Kelihos_ver3
Vundo
Obfuscator.ACY
Lollipop
Lollipop
Kelihos_ver3
Kelihos_ver3

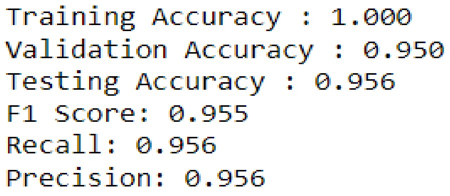
(a)

	count
labels	
Gatak	1013
Kelihos_ver1	398
Kelihos_ver3	2942
Lollipop	2478
Obfuscator.ACY	1228
Ramnit	1541
Simda	42
Tracur	751
Vundo	475

(b)

Figure 2. (a) Sample dataset images (b) Count of images in each malware category

Studies were done in a Windows 10 environment with an Intel Core i7 CPU operating at 4.8 GHz and 32 GB of RAM. Python programming language was used to implement the proposed design. Assessment procedures are carried out sequentially to ensure the random selection of training, validation, and test datasets. The selection rates of the data available for the training, validation, or testing phases are set at 60%, 20%, and 20%, respectively. Figure 3 illustrates the confusion matrix for the proposed RFL. A confusion matrix is a visual demonstration of ground-truth labels compared to model predictions. Using the confusion matrix, you can see how many different classes of cases are represented in each row and column. On the x-axis, the predicted label for Metrix is shown, while on the y-axis, the actual label is shown. Using the confusion matrices, the accuracy rates of each malware version are shown below. No doubt about it offered approach works.



```
Training Accuracy : 1.000
Validation Accuracy : 0.950
Testing Accuracy : 0.956
F1 Score: 0.955
Recall: 0.956
Precision: 0.956
```

Figure 3. Evaluation Metric

- Evaluation metric of the proposed model: Figure 3 shows the proposed model performance with multiple parameters. The results show the proposed DNN+RFC model achieved training accuracy is 100% which is very good and much better than existing work. The validation accuracy of the proposed model is 95%, the training accuracy is 95.6% and the f1 score, recall, and precision are 95% approx. All three last parameters achieved similar results. These outcomes clearly show proposed model is much better in comparison to the base work.

Figure 4 and Table 1 show comparison outcomes of the base and proposed model. Furthermore, the results were compared to the most recent findings in the field.

Figure 5 shows the distribution graph of the malware dataset with each category. All potential data values (or intervals) are shown in a data distribution as a function or a listing. It also informs you how often each value is used. In the above figure, the x-axis shows the malware variant that is shown in the graph very clearly, and the y-axis shows the number of counted images.

Table 1. Comparison between Base (DNN) and proposed model

Parameters	Proposed (DNN+RFC)	Base (DNN)
Accuracy	100	94.47
F1 Score	95.5	71.06
Recall	95.6	70.28
Precision	95.6	83.06

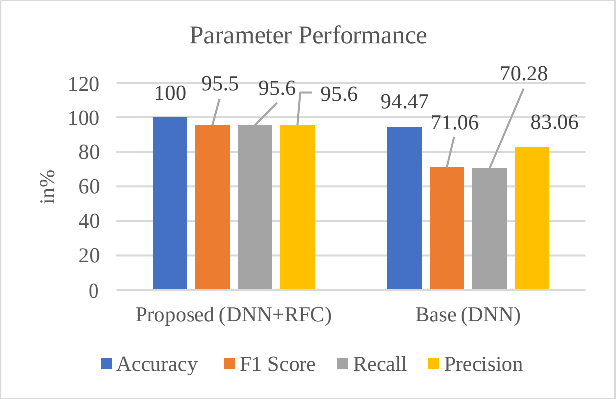


Figure 4. Graphical Comparison between base & proposed model

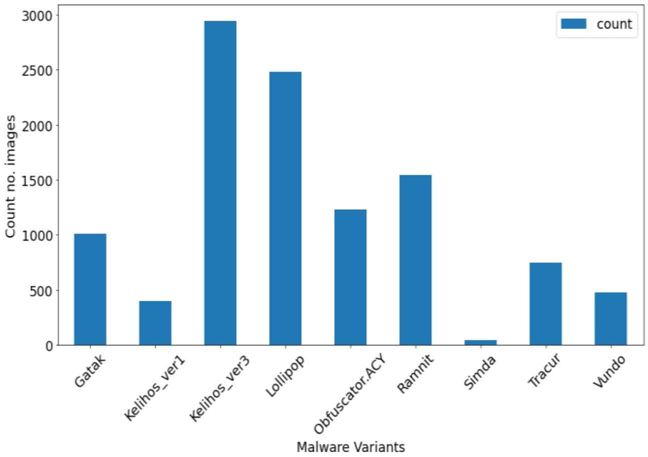


Figure 5. Dataset Distribution graph of each category

5. CONCLUSION AND FUTURE WORK

Even though a substantial amount of research has been conducted on the detection & classification of malware, determining the best effective approach for detecting malware variants remains a serious problem in the world of information security. Malware identification is made more difficult by obfuscation and packing methods that hide the source code. This paper proposed novel machine learning (Random forest classifier) and DL-based (Custom Deep neural network) architecture to efficiently detect malware variants. The fusion Model is used in the suggested design. Numerous datasets were used in the beginning to gather malware data. After that, pre-trained networks are used to extract the features. Lastly, a supervised learning approach is utilized to train DNN design. Fusion models may be created by optimally joining two pre-trained network models, which is the primary contribution of the proposed technique Microsoft's Big 2015 dataset is used to test the suggested machine learning & deep learning techniques. The proposed Fusion model is then compared to each model.

6. REFERENCES

[1] R. Lyer, “The Political Economy of Cyberspace Crime and Security,” *New York,NY, USA*, 2019.

[2] R. G. and S. P. Agarwal, “A comparative study of cyber threats in emerging economies,” *Globus, Int. J. Manag. IT,* vol. 8, no. 2, pp. 24–28, 2017.

[3] A. Singh, A. Handa, N. Kumar, and S. K. Shukla, “Malware classification using image representation,” 2019, doi: 10.1007/978-3-030-20951-3_6.

[4] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, “Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm,” *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3036491.

[5] E. Al Daoud, I. Jebril, and B. Zaqaibeh, “Computer virus strategies and detection methods,” *Int. J. Open Probl. Comput. Math.*, 2008.

[6] M. Al-Janabi and A. M. Altamimi, “A comparative analysis of machine learning techniques for classification and detection of malware,” 2020, doi: 10.1109/ACIT50332.2020.9300081.

[7] O. Aslan and R. Samet, “A Comprehensive Review on Malware Detection Approaches,” *IEEE Access*. 2020, doi: 10.1109/ACCESS.2019.2963724.

[8] “Significant Permission Identification for Machine Learning Based Android Malware Detection,” *Int. J. Res. Appl. Sci. Eng. Technol.*, 2021, doi: 10.22214/ijraset.2021.37673.

[9] O. Aslan and A. A. Yilmaz, “A New Malware Classification Framework Based on Deep Learning Algorithms,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3089586.

[10] R. J. Mangialardo and J. C. Duarte, “Integrating Static and Dynamic Malware Analysis Using Machine Learning,” *IEEE Lat. Am. Trans.*, 2015, doi: 10.1109/TLA.2015.7350062.

[11] A. B. Kathole, P. S. Halgaonkar, and A. A. Nikhade, “Machine learning & its classification techniques,” *Int. J. Innov. Technol. Explor. Eng.*, 2019, doi: 10.35940/ijitee.i3028.0789s319.

[12] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, “DL 4 MD : A Deep Learning Framework for Intelligent Malware Detection,” 2016.

[13] J. C. Kimmel, A. D. McDole, M. Abdelsalam, M. Gupta, and R. Sandhu, “Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure,” *IEEE Access*, vol. 9, pp. 68066–68080, 2021, doi: 10.1109/ACCESS.2021.3077498.

[14] S. Yang, S. Li, W. Chen, and Y. Liu, “A Real-Time and Adaptive-Learning Malware Detection Method Based on API-Pair Graph,” *IEEE Access*, vol. 8, pp. 208120–208135, 2020, doi: 10.1109/ACCESS.2020.3038453.

[15] A. Alotaibi, “Identifying Malicious Software Using Deep Residual Long-Short Term Memory,” *IEEE Access*, vol. 7, pp. 163128–163137, 2019, doi: 10.1109/ACCESS.2019.2951751.

[16] Y. Li, K. Xiong, T. Chin, and C. Hu, “A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection,” *IEEE Access*, vol. 7, no. c, pp. 32765–32782, 2019, doi: 10.1109/ACCESS.2019.2891588.

[17] A. O. Salau and S. Jain, “Feature Extraction: A Survey of the Types, Techniques, Applications,” 2019, doi: 10.1109/ICSC45622.2019.8938371.

[18] A. S. Bozkir, A. O. Cankaya, and M. Aydos, “Utilization and comparision of convolutional neural networks in malware recognition,” 2019, doi: 10.1109/SIU.2019.8806511.

[19] H. Wang *et al.*, “An Effective Approach for Malware Detection and Explanation via Deep Learning Analysis,” 2021, doi: 10.1109/IJCNN52387.2021.9534115.

[20] H. Yi, Q. Jiang, X. Yan, and B. Wang, “Imbalanced Classification Based on Minority Clustering Synthetic Minority Oversampling Technique with Wind Turbine Fault Detection Application,” *IEEE Trans. Ind. Informatics*, 2021, doi: 10.1109/TII.2020.3046566.

Authors Profile

Dr. Sonali Kothari has obtained PhD in computer engineering from Sant Gadge Baba Amravati University, Amravati, India. Currently she is working as Associate Professor in Department of Computer Science and Engineering at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India.

Vijayshri Khedkar is an Assistant Professor skilled in NLP, Data Analytics & Deep Learning. A life-long learner with a strong educational background holding two Master’s Degrees (M.B.A. & M.E.) and pursuing Ph.D. in Computer Engineering (NLP) from Symbiosis International University, India.

Shwetambari Chiwhane is an Assistant Professor skilled in Artificial Intelligence and Machine Learning. She has been awarded Ph.D. in Deep Learning from Bharath Institute of Higher Education and Research, Chennai, India. She is currently associated with Symbiosis International University, Pune, India as an Assistant Professor in Department of Computer Science and Engineering.

Ujwala A. Kshirsagar is an Associate Professor skilled in IoT, VLSI and Embedded System Design. A life-long learner with a strong educational background holding Master's Degrees M.E. Digital Electronics and Ph.D. in VLSI Technology from Sant Gadge Baba Amravati University, Amravati, India. She is currently working as an Associate Professor in the Department of Electronics and Tele Communication, Symbiosis International University, India.