Journal of Computational Analysis and Applications VOL. 33, NO. 4, 2024
ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

Classification of Malware using Machine learning and Deep learning Techniques

[1] Dr.Khyati Rami ,[2] Prof. Prashant Halvadiya , [3] Prof. Mayank Devani

[1] Assistant Professor, Department of Computer Application, SAL Institute of Technology and Engineering Research, Ahmadabad (Gujarat)

[2] Assistant Professor, Department of Computer Application, SAL Institute of Technology and Engineering Research , Ahmadabad (Gujarat)

[3] Assistant Professor , Department of Information Technology , SAL College of Engineering , Ahmadabad (Gujarat)

ABSTRACT

The threats impose by the cyber-attacks due to malicious software (malware) have been increasing drastically with the evolution of information technology. Since people use web applications on a daily basis these malware attacks have become challenging. There have been various attacksaffecting confidentiality, integrity and availability of data which has become a major security concern. Though the manual inspection and classification methods seemed to bring up some light to this facet, these methods are no longer considered effective, since they are time consuming and inefficient. With the high-rate malware spreading, it is a necessity to come up with some novelty approach to classify them as malware or benign software. So, this is wheremachine learning comes up as a novelty approach in malware classification. In this paper, a malware dataset was used on several machine learning classifiers like Support Vector Machinery(SVM) and Gaussian Naive Bayes classifiers were used and Recurrent Neural Network (RNN) andConvolutional Neural Networks (CNN) were used as the deep learning classifiers. Although there are many other methods for malware classification,amachinelearningapproach could be efficient and effective in detecting malicious software. Thus, the primary objective of this paper is to provide an insight to the machine learning approach in malware classification by depicting, which is the best classifier of the listed, that can effectively classify malware based on their accuracy or precision. In conclusion, based on the results this recognizes Recurrent Neural networks as the best approach that recorded the highest accuracy.

Keywords

Malware Classification, Machine Learning, Deep Learning, Binary Classification

1. INTRODUCTION

In today's hyper-connected digital world, cybersecurity threats are escalating rapidly in both volume and sophistication. Among these threats, **malicious software (malware)** stands out as one of the most prevalent and dangerous. Malware refers to software intentionally designed to disrupt, damage, or gain unauthorized access to computer systems, networks, or data. As technology advances, malware authors continuously evolve their tactics to bypass traditional

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

detection systems, making **accurate and timely malware classification** a critical component of modern cybersecurity.Traditional malware detection approaches, such as **signature-based** and **heuristic-based** methods, are increasingly inadequate in identifying new or **zero-day attacks** due to their dependency on known patterns. These limitations have driven researchers and security professionals toward more intelligent and adaptive solutions, particularly those offered by **Machine Learning (ML)** and **Deep Learning (DL)**.Machine Learning algorithms can learn patterns from previously labeled malware and benign files, enabling the model to generalize and predict the nature of new, unseen samples. Techniques such as **Random Forests, Support Vector Machines (SVMs), and Decision Trees** have shown promising results in malware detection and classification tasks. With the evolution of technology, the growth of malwares has posed an immense threat to the internet security. A malicious software(malware)canbedefinedasanycodeadded, altered, or deleted from a software system with the aim of causing detrimental effects or destabilize the intended function of the system [1]. A malware can be used to breach the security policy of a computer system in terms of confidentiality, integrity, and availability. The execution characteristics of the program defines the malware classification. It is alsoclassified based on its payload, method of vulnerability exploitation and the propagation method [2]. Based on this, malwares can be classified as virus, worm, Trojan horse, Adware ransomware and spyware. In the early days of malware evolution, it came into being as a prank or an experiment gone wrong. However, there are several malwares that were created in laboratories like Darwin game, creeper, pervading animal, and Rabbit Virus, but never released out of thelabs[2].ItwastheElkCloner(in1981),whichwasthefirst virus to be released out and infect a personal computer. Followed by it, Brain; the first Microsoft PC virus came into being as a more harmful virus than Elk Cloner [2]. Followed by them, the malware has systematically conquered the computer world. Zeus in 2007, Koobface in 2008, Stuxnet in 2010, Crypto locker Trojan in 2013, and Wannacry in 2017 have trembled the world throughout the last decade.Therefore, it is a necessityto develop a method for effectively identify malware before they cause damages to the computer systems.To keep up with the malware, cyber-defenses are being constantly improved by the researchers and security analysts. Endpoint protection is one such essential element where it provides a suite of security programs like firewall, email protection, URL filtering, anti-spam, and sandboxing [3]. The last layer of defense is mostly provided by the Antivirus software which is responsible for detection and removal of malware installed on the endpoint device. Typically,antivirus software relies on a signature based or heuristic method. A signature can be defined as an algorithm or a hash that identifies a specific malware uniquely while a heuristicmethod defines a set of rules that determined by analysts after behavior analysis of malware. However, this method is not so effective in detecting malware. The major challenge in such traditional approaches is that the new variant of malware uses bypassing techniques which makes the code obfuscate. This will lead to failures in detecting new types of malwares. A malware analysis is typically composed of 2 parts namely discovery phase and the classification phase.In the discovery phase malware is caught and identified. Feature vector selection methods are being used for malware classification

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

process and this can be classified as static analysis and dynamic analysis. These differ from each other based on the way features are extracted. Static analysis of malware uses a method of capturing information from binary programs without executionanddynamic analysis of malware, monitors the behavior of malware at run time in an isolated system [4]. However, malware detection methods based on dynamic analysis is more robust to obfuscation methods in comparison to static analysis methods. A hybrid approach can be reached by the combination of these two methodologies. So, it is a necessity to implement a novelty approach in detecting malicious software and that is where the machine learning techniques comes in. This research aims at analyzing theusage of machine learning techniques to examine Malicious software. This paper further exemplifies the categorization of feature representation used to generate training data, while comprehending algorithms used tounderstand a perfect prediction model. As the primary objective, this paper intends to conceptualize the most effective and efficient machine learning approach in classifying malware, which basically divides them into malware or benign that involves a binary classification approach.

2. Background

2.1 Malware Overview: Malware, short for **malicious software**, encompasses various types of harmful software such as **viruses, worms, trojans, ransomware, spyware, and rootkits**. Its primary goal is to compromise system integrity, steal sensitive data, or disrupt normal operations. As malware continues to evolve in complexity and stealth, traditional defense mechanisms often fall short in detecting newly emerging threats.Malware can be broadly categorized into:**Static malware**: Identified based on file structure, binary patterns, or headers.**Dynamic malware**: Behavior is analyzed during execution in controlled environments (e.g., sandboxes).Modern malware also uses **code obfuscation**, **encryption**, and **polymorphism** to evade traditional signature-based detection, making it increasingly difficult to track using conventional techniques.

Bugra Cakir et al. [1], has used ashallowdeep learning-based method for feature extraction. Here for the classification a Gradient Boosting Algorithm has been used and the performance of the model has been evaluated using k-fold cross validation split without sacrificing a validation split, which reached an accuracy of 96%. However, Matilda Rhode et al. [10], has used a recurrent neural network to predict whether an executable is malicious or benign which produced an accuracy of 94%. The tested dataset was obtained from VirusTotal which consists of 1000 malicious and 600 trusted windows 7 executables along with along with 800 trusted samples from the system files of a fresh Windows7 64-bit installation. Huan Zhou in his paper proposes a deep learning methodology for malware detection using static and dynamic combined features to classify a portable execute file (PE) is malicious or not [11]. In the research by Ivan Firdausi et al. [12], uses several machine learning algorithms like k-Nearest Neighbors (kNN), J48 Decision Tree, Support VectorMachine (SVM), Naïve Bayes and Multiplayer Perceptron Neural Network

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

(MLP). However, the best accuracy of96.8%, precision of 95.9% along with a false positive rate of 2.4% was obtained for J48 decision tree and the dataset used consists of malware and benign data in the format ofWindows PE [12]. Furthermore, Muhammad Furqan Rafique et al. [13] propose a deep learning malware detection technique. The features are extracted using two types of convolutional neural networks and finally most important and distinctive features are selected using a SVM and a wrapper- based mechanism. The dataset BIG 2015 published by Microsoft on the Kaggle platform, is finally trained using a Multilayer Perceptron.

3. Machine Learning Approaches:

Machine Learning (ML) models are widely employed in malware classification due to their ability to analyze complex patterns within engineered features. These features are derived from **static analysis** (examining malware without execution) and **dynamic analysis** (observing behavior in a controlled environment). ML models rely heavily on well-processed and relevant features, as their performance depends on the quality of the input data.

3.1 *Feature Extraction in ML*

Feature extraction is a critical step in building effective ML-based malware classifiers. It involves converting raw malware data into structured formats suitable for algorithms.

- **Static Analysis**: Extracts features from the malware without executing it. Common features include **PE header information**, **file size**, **hash values (MD5, SHA256)**, and **API call imports**.
- **Dynamic Analysis**: Requires executing the malware in a sandboxed environment to observe real-time behavior. Captured features include **API call logs**, **memory usage patterns**, **registry changes**, and **network traffic packets**.
- **N-grams**: Involves extracting opcode sequences (machine-level instructions) from malware binaries and treating them similarly to text tokens for pattern recognition.

3.2 *Machine Learning Algorithms*

(a) Decision Tree (DT)
A tree-structured algorithm that splits data based on feature thresholds using **information gain** or **Gini index**. Decision Trees are interpretable and effective for small datasets but are prone to overfitting.

(b) Random Forest (RF)
An ensemble of multiple Decision Trees built using **bagging** to enhance robustness. Random Forest offers high accuracy, reduces overfitting, and provides feature importance rankings. It is highly suitable for handling high-dimensional malware data.

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

(c) Support Vector Machine (SVM)
SVM classifies malware by finding the optimal **hyperplane** that separates classes. It is effective for binary classification (malware vs benign) and uses the **kernel trick** for handling non-linear data. However, it becomes computationally expensive on large datasets.

(d) k-Nearest Neighbor (k-NN)
An instance-based algorithm that classifies malware based on the **closest neighbors** in the feature space. While simple, it is slow on large datasets and highly sensitive to feature scaling.

(e) Naïve Bayes (NB)
A probabilistic classifier based on **Bayes' theorem**, assuming feature independence. It performs well with text-like features such as **opcode sequences** and **API call logs**, and it is fast and memory-efficient. However, the independence assumption can limit accuracy.

(f) Gradient Boosting (XGBoost, LightGBM)
An advanced **boosting ensemble method** that iteratively improves weak learners. It delivers superior performance on large-scale datasets, effectively handles class imbalance, and processes high-dimensional data efficiently.

3.3 Feature Selection for ML

Feature selection optimizes model performance by reducing noise and dimensionality. Common techniques include:

- **Chi-square Test**: Evaluates the statistical significance of categorical features.
- **Mutual Information**: Measures dependency between features and target labels.
- **Recursive Feature Elimination (RFE)**: Iteratively removes less significant features.

This step improves accuracy, reduces overfitting, and speeds up training.

3.4 Evaluation Metrics

Performance assessment of ML models in malware classification uses:

- **Accuracy**: Overall correctness of predictions.
- **Precision & Recall**: Precision measures correctness of positive predictions, recall measures coverage of actual positives.
- **F1-score**: Harmonic mean of precision and recall.
- **ROC-AUC**: Evaluates classification ability across thresholds.
- **Confusion Matrix**: Summarizes classification outcomes (True Positives, False Positives, etc.).

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

Convolutional Neural Network (CNN): A CNN consists with one or more convolutional layers with fully connected layers on top and it has pooling layers, and weights [8]. Being able to train easily than feed forward neural networks and their ability to train with standard back propagation makes it a highly attractive architecture to be used [8]. Fig 1 depicts an image of a typical CNN

Recurrent Neural Network(RNN): RNN is commonly used for sequential attribute recognition of a dataset and predicting the next likely scenario using patterns [9]. Long ShortTerm Memory (LSTM) is and type of RNN that uses backpropagation, but it uses memory blocks connected to layerstolearnsequencedata. AtypicalstructureofaLSTMis denoted in Fig2.

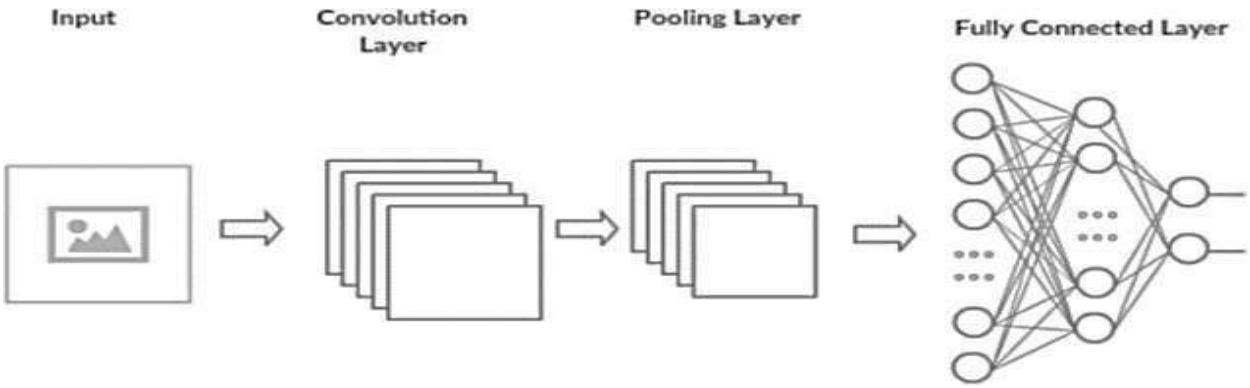

Fig1:ConvolutionalNeuralNetworkStructure

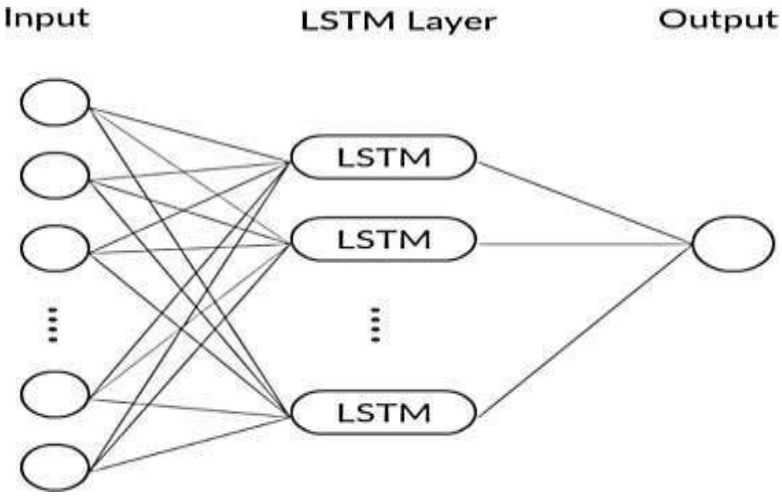

Fig2:LSTMNetworkStructure

4. METHODOLOGY

4.1 DatasetSelection

The machine learning models must be trained before it isbeing tested on a real-life scenario. For the training purpose a recognized data set should be used. However, the problemface by most researchers is that finding out a high-quality dataset. The scarcity of good datasets has

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

set a drawback in this domain. The data set used here is taken from the Kaggle data set repository[14].Thisdatasetconsistsof35columns and consists of 50000 unique samples of benign and 50000 unique samples of malware which altogether makes a dataset of 100000 samples.

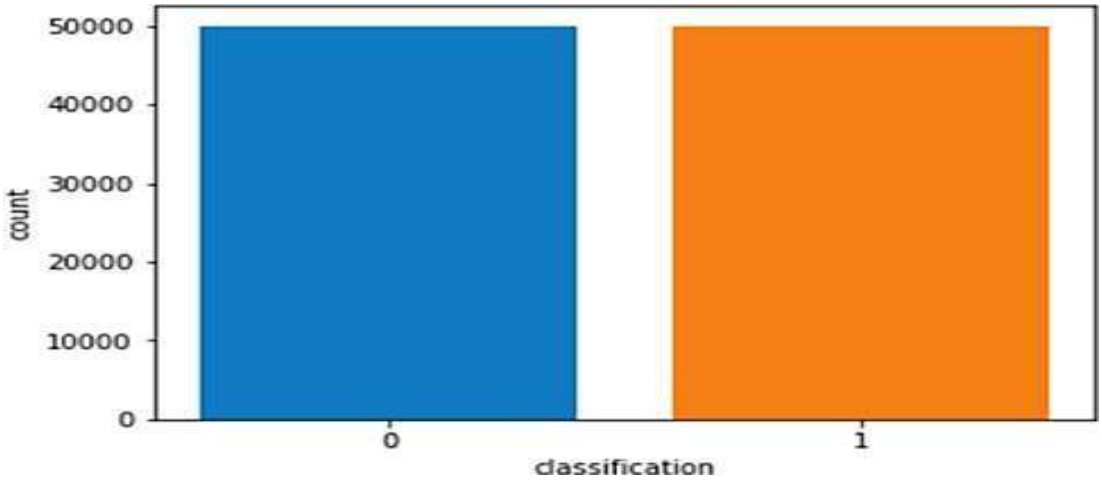

Fig3:DatasetDistribution

Data preprocessing and feature selection: Before training the machine learning models the dataset should be preprocessed and features should be selected. The feature selection was carried out only for the machine learningmodels and it was done using Extra Tree Classifier. The deep learning models were trained without feature selection. The dataset was imported in the format of data frames, and it was then shuffled. Since the first 500000 data samples were malware and the next 500000 samples were benign, thedataset was shuffled to increase the randomness and prevent the overfitting during training. Then the features and targets were selected from the shuffled dataset. At the basic level 33 columns were selected as data (features) dropping the „hash" column and the column with the name „classification" was selected as the target (labels). Since data consist of valuesfrom different ranges, it had to be normalized. As the classificationisaddressedinabinaryform,thestrings„malware" and „benign"in thetargetwereconverted to„0"and„1" respectively.

Training: The pre-processed dataset with selected features is then used to train the desired machine learning algorithm. Selection of the machine learning algorithm should be done, based on the scenario it is being applied for. Several, machine learning and deep learning models are discussed earlier in the literature. The dataset with data and targets is then feed into the respective machine learning model and the model is allowed to train. If deep learning methods like Neural Networks are being used the model can be trained for several epochs to increase the level of training. Since there is no separate dataset for testing the dataset should be split intotrain data, test

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

data, train target and test target respectively. In this context 2 machine algorithms namely SVM and Naive Bayes were used. As the deep learning algorithms Convolutional Neural Networks and Recurrent Neural Networks were used

Testing: This is the final phase of a machine learning approach. This is where the testing dataset is feed into the machine learning model to predict the accuracy of the model which provides an idea of how well the model reacts for unseen data. When feeding the data into the trained machine learning model,the data of the testing dataset should be in the same format of the training dataset. Since there is noseparated testing dataset, 10% of the training dataset is used for the testing purpose

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 31, 128)           512
_________________________________________________________________
activation_1 (Activation)    (None, 31, 128)           0
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 31, 16)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 27, 64)            5184
_________________________________________________________________
activation_2 (Activation)    (None, 27, 64)            0
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 27, 8)             0
_________________________________________________________________
dropout_1 (Dropout)          (None, 27, 8)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 216)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 217
=================================================================
Total params: 5,913
Trainable params: 5,913
Non-trainable params: 0
```

Fig4:ProposedCNNArchitecture

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 33, 64)            16896
_________________________________________________________________
dropout_1 (Dropout)          (None, 33, 64)            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 33, 128)           98816
_________________________________________________________________
lstm_3 (LSTM)                (None, 33, 64)            49408
_________________________________________________________________
dropout_2 (Dropout)          (None, 33, 64)            0
_________________________________________________________________
lstm_4 (LSTM)                (None, 128)               98816
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 264,065
Trainable params: 264,065
Non-trainable params: 0
```

Fig5:ProposedRNN Architecture

4.2 PerformanceEvaluation parameters

Evaluating the performance of a machine learning model is a fundamental aspect of machine learning. This helps inrefining the parameters and selecting the best and most appropriate model from which are being tested. The most commonly used methods for evaluation of machine learning models, is calledmatrix [15]. Furthermore, metrics like Coefficient of determination (R2 score), accuracy and precision are also being used in determining the classifiers performance

Confusion Matrix: A confusion matrix consists of 2 rowsand 2 columns which gives the number of false positive (FP), false negative (FN), true positive (TP) and true negative (TN).The Fig 6 depicts an image of a confusion matrix. In a confusionmatrixanobservedclassisrepresentedbyeachrow while each column represents a predicted class [15]. Integer numbers are the entries of a confusion matrix, and the total of TP, TN, FP, and FN becomes equal to the number of test data [15].

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

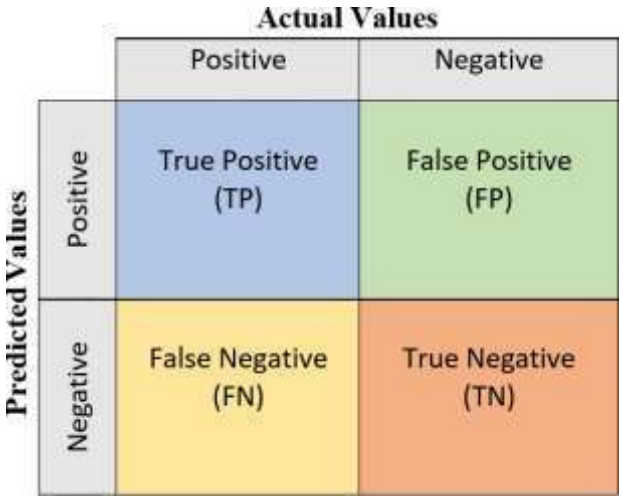

Fig6:ConfusionMatrix

Accuracy: Accuracy can be considered as a proportion oftotal number of predictions that were correctly identified. The following equation can be used to determine the accuracy.

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

Precision: The precision is defined as a proportion of predicted positive cases that were correctlyidentified, and the equation below can be used to determine the precision [15].

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

Coefficient of determination (R2 score): This is a statistic usedinstatisticalmodelswhereprimaryobjectiveistopredict andprovideameasurementofthefutureoutcomeofthetested model [15].

$$r = \frac{n\sum(xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (3)$$

Here,r=TheCorrelationcoefficient,n=numberinthegiven dataset, x = first variable in the context, y= second variable

RESULTS

The Table 1 and Table 2 depicts the results obtained after the testing phase. The Table 1 consists of results without feature selection and Table to provides a comparison of results for machine learning classifiers with and without featureselection. In this study the main performance evaluation parameters that were included was accuracy, precision, R2 score, and parameters related to the confusion matrix. Based on these parameters several factors can be highlighted. According to the table 1, CNN and RNN which are deep learning algorithms depicts a prominent accuracy and precision over the traditional machine learning algorithms.

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

However, it was obvious that when using the SVM, it had a clear advantage over the Gaussian Naïve Bayes algorithm. Moreover, the false positive and negative rates of theGaussian Naïve Bayes algorithm were at a higher level which indicates that it is not that suitable for malware classification. This also gives a clear indication that proper usage of deep learning algorithms can result in great results.

Table1.Resultsofthe classifierswithoutfeatureselection

	Accuracy	Precision	FP	FN	R2
Gaussian NB	69.56%	62.22%	53.54%	7.06%	-0.21
SVM	98.61%	97.41%	2.55%	0.24%	0.94
CNN	99.67%	99.99%	0.28%	0.38%	0.96
RNN	99.96%	99.93%	0.06%	0.02%	0.99

The table 2 depicts the impact of feature selection and not selecting features on machine learning algorithms. Thistabular depiction provides a clear idea that, through proper feature selection accuracy and precision. This was very much obvious in Gaussian Naïve Bayes algorithm, which initially had an accuracy of 69.56% and precision of 62.22% resultedin an accuracy of 72.51% and precision of 65.08%. It also resulted in reducing the false positive rate from 53.54% to 40.13%. In addition to this, even the SVM algorithm too showed some promising results. According to the table 2, the SVM algorithm has obtained an accuracy of 98.61% and precision of 97.41% when it was trained without feature selection. After the feature selection SVM resulted in an accuracyof99,94%andprecisionof99.23%.

Table2.Resultsofclassifierswithandwithoutfeature selection

	Accuracy	Precision	FP	FN	R2
GaussianNB without featureselection	69.56%	62.22%	53.54%	7.06%	-0.217
Gaussian NB with featureselection	72.51%	65.08%	40.13%	14.69%	-0.099
SVM withoutfeature selection	98.61%	97.41%	2.55%	0.24%	0.9443
SVM withfeature selection	99.44%	99.23%	0.39%	0.72%	0.9775

However, the feature selection was done onlyfor the machine learning classifiers while deep learning classifiers weretrained without feature selection. Based on the results it was the RNN

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

that had the better accuracyand R2 score and lowest percentage of False Positive and False Negative rates. TheFig. 9 provides a graphical representation of the results depicted in the Table 1. Fig.7 and Fig.8 presents how the validation accuracy and Accuracy varied during CNN and RNN training and testing phase.

Fig7:VariationofAccuracyandValidationAccuracyfor CNN Model

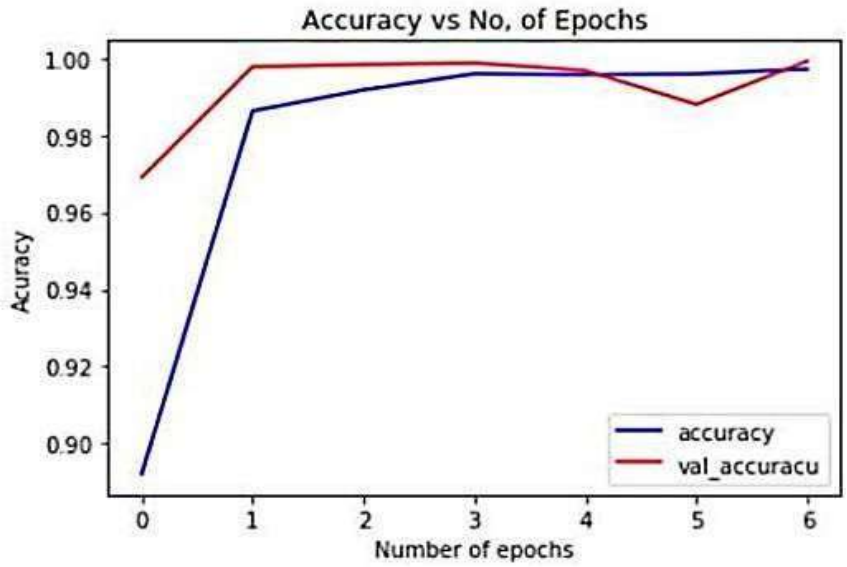

Fig8:VariationofAccuracyandValidationAccuracyfor RNN Model

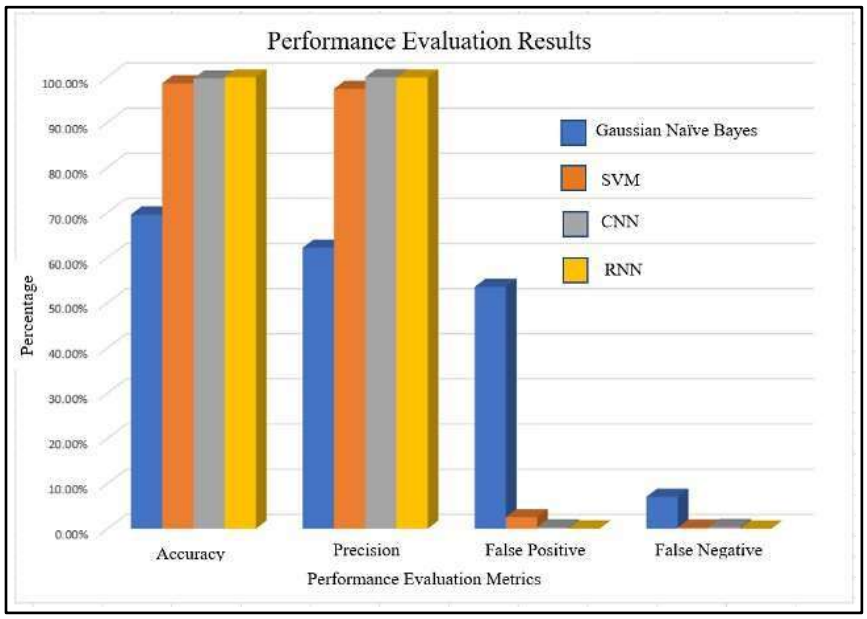

Fig9:Graphicalrepresentationoftheresults

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

Fig 9 depicts a clear graphical representation of the variation of accuracy, precision, false positive rates and false negative rates of Gaussian Naïve Bayes, SVM, CNN and RNN. Based on this graph too ,it is clear that the RNN is having the highest accuracy as well as the highest precision while GaussianNaïve Bayes is having the lowest accuracy and the lowest precision level. Apart from that it is clearly visible that, the Gausian Naïve Bayes is having the highest FP and FN rate. Another factor that can be highlighted here is that, RNN is having very low FP and FN values compared to the Gausian Naïve Bayes Algorithm. So, based on the graph it is obvious that RNN is having a clear advantage against the othermodels.

5. LIMITATIONSANDFUTUREWORK

When compared to the other detection methods, several research areas in this domain are yet to be discovered. However, for the existing research work modifications can be done to obtain greater results. The analysis of the literature discussed above outsmarted the fact that the feature selection technique needs paramount improvements in detecting malware at a best accuracy rate. This can prevent confusionsin malware identification. Modifications of the machine learning models is a necessity because over time attackersfind mechanisms to overrule the prevailing defense measures. The fine tuning of the machine learning algorithms can produce better results utilizing the features to function at an optimumlevel. Thelimitationsindatasetscreatedrawbacksin most studies. A small data set can generate incorrect predictions. Since the same dataset was used for both training and testing the results of this research is bit higher than using separatedatasets fortrainingandtesting.Generationofproper datasets of a considerable size can greatly help researchers to generate machine learning models with better results. Toavoid the overfitting problems in classifiers more attention should be given for the data preprocessing stage and in processing, clustering can be used to sort out outliers. By preventingtheinclusionofsuchnoisysamplesintraining,can also generate good results.

6. CONCLUSION

Despite the extensive studies and staggering progress that the machine learning approach on malware classification have gained in the recent years; yet it remains a very challenging domain. However, in this domain, it is a necessity to upgrade the existing strategies since the attackers too develop counter measures to overrule the defense strategies. Based on the results it was the Recurrent Neural Network (RNN), that recorded the highest accuracy rate of 99.96% and R2 score of 0.9989. It also had the lowest False Negative Rate of 0.02% and False Positive Rate of 0.06%. However, Convolution Neural Network (CNN), too had promising results when compared with the machine learning classifiers. The results of this research have also provided an insight to the machine learning approach in malware classification by highlighting the fact that, feature selection can be effectively used for machine learning classifiersto produceoutstandingresults. In

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

conclusion, itcan be said that Recurrent Neural Network (RNN) is a promising machine learning approach in malware classification.

7. REFERENCES

[1] B. Cakir and E. Dogdu, "Malware classification using deep learning methods," Proc. ACMSE 2018 Conf., vol. 2018-January, no. April 2018.

[2] A. P. Namanya, A. Cullen, I. U. Awan, and J. P. Disso, "The World of Malware: An Overview," Proc. - 2018 IEEE 6th Int. Conf. Futur. Internet Things Cloud,FiCloud 2018, no. September, pp. 420–427, 2018.

[3] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," J. Netw. Comput. Appl., vol.153, no. July2019,p. 102526,2020. [Online].Available:https://doi.org/10.1016/j.jnca.2019.10 2526

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran,andS.Venkatraman,"RobustIntelligentMalwareDetectionUsingDeepLearning," IEEE Access, vol. 7, pp. 46 717–46 738, 2019.

[5] V. Menger, F. Scheepers, and M. Spruit, "Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text," Applied Sciences, vol. 8, no. 6, p. 981, Jun. 2018. [Online]. Available: https://doi.org/10.3390/app8060981

[6] C. Liu, L. Wang, B. Lang, and Y. Zhou, "Finding effective classifier for malicious URL detection," in Proceedingsofthe20182ndInternationalConferenceon Management Engineering, Software Engineering and Service Sciences - ICMSS 2018. ACM Press, 2018. [Online].Available:https://doi.org/10.1145/3180374.3181 352

[7] Z.Cui,F.Xue,X.Cai,Y.Cao,G.G. Wang,andJ.Chen, "Detection of Malicious Code Variants Based on Deep Learning," IEEE Trans. Ind. Informatics, vol. 14, no. 7,pp.3187–3196,2018.

[8] B.B.Benuwa,Y.Zhan,B.Ghansah,D.K.Wornyo,andF.B.Kataka,"Areviewofdeepmachinelearning,"Int.J. Eng. Res. Africa, vol. 24, no. February 2017, pp. 124– 136, 2016.

[9] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware neural language models," 30th AAAI Conf. Artif. Intell. AAAI 2016, pp. 2741– 2749, 2016.

[10] M. Rhode, P. Burnap, and K. Jones, "Early-stagemalware prediction using recurrent neural networks," Comput.Secur.,vol.77, no.December2017,pp. 578–594, 2018. [Online]. Available: https://doi.org/10.1016/j.cose.2018.05.010

[11] R. Olson, G. James, and R. Howard, Cyber Security Cyber Security. Springer Singapore, 2011. [Online]. Available: http://dx.doi.org/10.1007/978-981-10-8536-98

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.29

[12] I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-basedmalwaredetection,"Proc. -20102ndInt. Conf. Adv. Comput. Control Telecommun. Technol.ACT 2010, pp. 201–203, 2010.

[13] A. M. M. Muhammad Furqan Rafique, Aqsa Saeed Qureshi, Asifullah Khan, Jin Young Kim, "Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique Muhammad," pp. 1–20.

[14] Saravana, "Malware detection," Apr 2018. [Online]. Available: https://www.kaggle.com/nsaravana/malware-detection.

[15] L. Zhang, Y. Zhu, P. Shi, and Q. Lu, "Performance analysis," Stud. Syst. Decis. Control, vol. 53, pp. 59–85, 2016.