IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Mobile App Malware Detection Using Deep Learning

[1]Roja Pravallika,[2]Dr. Shivakumar B R

[1]M. tech student, [2]Associate Professor

[1]Depratment of ISE,

[1]Bangalore Institute of Technology, Bangalore, India

Abstract: Latest smartphone stages built on new operating systems, such as iOS, Android, or Windows Phone, have been a vast achievement in recent years and built-up various new opportunities. Unfortunately, 2011 also indicated us that the new machineries and the privacy-related documents on smartphones are also gradually fascinating for attackers. In this project mobile app malware detection, which contains a Recurrent Neural Network (RNN) classification algorithm is implemented. The malware detection data set is analyzed that contains 35 features to study the viability of classification algorithm and malware detection is made by using LSTM method to achieve more accuracy. The unique malware dataset patterns obtained by this method is converted to their binary form and it is applied to the deep learning algorithm to classify the malware and benign files. In this method the malware can be predicted with 98.89% accuracy using Long Short-Term Memory (LSTM) which outperforms Malware detection-based methods.

Index Terms - Malware detection, Recurrent neural network, Feature selection, Long short-term memory, malware dataset

I. INTRODUCTION

Android mobile phone applications have become target for many malicious software variants, including malicious malware types such as viruses, ransomware and spyware. Android is an open-source operating system platform. Since mobile application store a huge amount of personal data such as contacts, banking and accounts along with other documents in the device. In this case, the attackers will steal the information through malware and misuse the information or damage the content of the devices. Malware attacks are typically delivered in the form of links, SMS (Short Message Service) and Email and if user clicks on the link or open the file, it may lead to malware attacks, which are affecting the genuine users worldwide causing huge damages which sometimes are irreversible and poses a potential threat on a regular basis.

There are many popular antivirus and anti-malware products online which help to protect devices from these threats. But most of the antivirus or anti-malware vendors use an old signature-based [5] malware detection method. In this case, the device files will be stored in the database with their hash values. Their hashes represent the integrity of a file. Whenever an antivirus product scans a computer, it will check for the hashes that match their database. If it detects any hash of a file that matches the malware hash in their database, that file is considered as malware or malicious and the user will be notified for further processing or it will remove that file from the system. On the other hand, the malware developers create new malware variants which are capable of evading these old detection methods on a regular basis. Hence most antivirus products fail when there is an attack from an advanced malware variant and keep the users at a huge risk. The overview of Mobile Malware Detection is shown in Figure 1.1. To find out whether a particular application is malware or benign, the first step is data preprocessing which involves lots of data to train our deep learning model and the data is typically stored in a storage system like a file or in the database systems, the raw data is available inside a file or database system that has to be preprocessed before passing it to the deep learning model for training purpose. The processing involves methods like loading the data into deep learning model and handling the missing value inside the data. The second step is feature selection. It is the most useful step for the selection the best features and can be used to compute impurity-based important feature, which in turn can be used to discard irrelevant features selected attributes of the given dataset are to be build the model and to increase the overall complexity of the model. Thus, feature selection becomes an essential part of building deep learning model. The third step is the LSTM (long short-term memory) model. LSTMs are a special type of RNN. It is an ensemble-based learning algorithm to obtain the predictive results in an efficient manner. It can be used for classification as well as regression. LSTM works on a large collection of dataset attribute values which handles the missing values and maintains the accuracy. It will not overfit the model. It also handles larger datasets with higher dimensionality. The fourth step is training and testing. The purpose of training and testing is to discover the predictive relationship by using the model. The model starts learning the behavior of the data and structure itself and this model is built on the data it discovers in the training dataset. Once the model is built, test the dataset to predict the accuracy. The fifth step estimates the predicted value once the model is trained based on the testing and the accuracy is predicted.

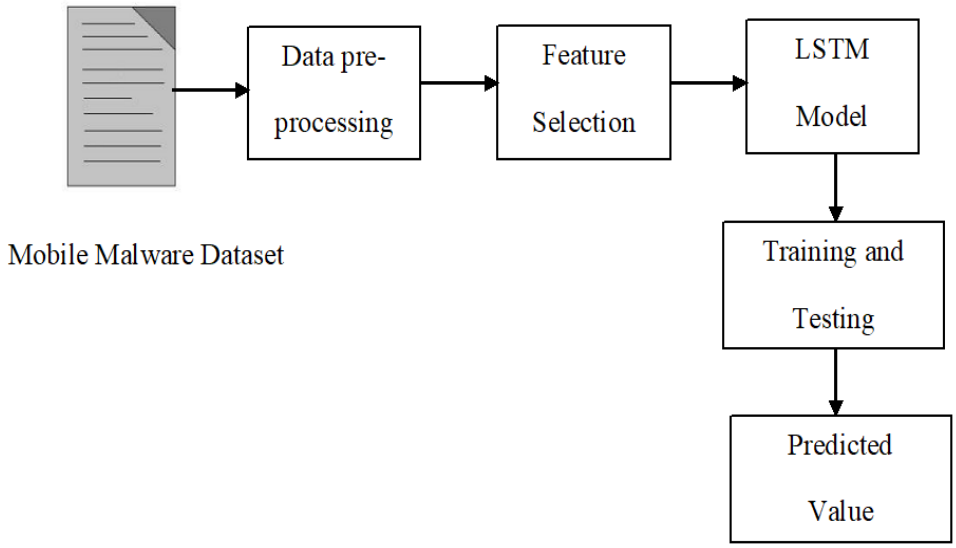

Figure1: Overview of Mobile Malware Detection

II. LITERATURESURVEY

A.Hemalatha, Selvabrunda [1] have proposed the machine learning classifiers to classify the previous portable malware, with the mixed kernel function which is unique to support vector machines, and Selected basic information like data content time and in order using various functions on the network-based. The result is performed on the given dataset taken from MalGenome. implementation was done based on the mixed kernel function on SVM, the result obtained from the given dataset was 96.89 % of accuracy by using the SVM function of the unique mixed kernel, which is simple and compared with existing models. To find the best TRP values by using random forest multilayer perceptron to classify and select the features and predict the accuracy based on the TRP values to advance the malware recognition rate in a wide range to classify higher dataset attributes.

Authors Mohammed K. Alzaylaee, Suleiman Y. Yerima, SakirSezerIgor Muttik Ensemble [3] have collected simpler base models which are used to predict the accuracy. For classification of the model random forest algorithm is applied, proposed ensemble learning method to find out the characteristics of the application which supports the 179 features. This model is used to classify and detect the mobile apps whether the application is good malware or bad malware. For classification of high accuracy prediction by utilizing a decision tree, ensemble learning is often applied to reinforce investigating Android malware detection. By extracting the dataset features of malware and benign from a deep learning method classifier can be used to experiment with situations. Furthermore, a comparative analysis is formed to Naïve Bayes, Decision Trees, Random Trees, and straightforward Logistic.

LSTM approach is proposed with the useful resource of the use of Hoch Reiter and Schmid Huber [4] in 2017. It is a feed-beforehand network with one or more hidden layers where its main purpose is to learn long-term reliance. LSTM possessed a framework like a chain and the repeating module includes a unique arrangement. There are four neural network layers that interact in a completely unique way with each other. Gibson and Patterson stated that each LSTM unit includes two classes of connections which are the connections from the preceding time-pace (unit outputs) and the connections from the earlier layer. The crucial additives of the LSTM architecture are the memory mobiliary and the gates (moreover includes the neglect mobiliary and the input mobility). The materials of the memory mobiliary are adjusted with the useful resource of the use of the input gates and neglect gates. Assuming that both of the gates are closed, the materials inside the memory mobiliary will stay unaltered between one time-paces and the next. In summary, the input gate defends the unit from unnecessary input incident, the neglect gate permits the unit to erase the past memory materials and the output gate reveal the materials of the memory mobiliary at the LSTM output unit.

Abdulrazak Yahya Saleh, Corrine Francis [6] has proposed an LSTM algorithm technique that will be used in the malware detection technique. This algorithm helps us to detect malware attributes and framework which includes 5 stages for detection of the malware family, in which the first stage consists of data preparation this consists of 138047 malware samples and 52 features from the dataset taken from that samples which are divided into malware and benign. The second stage is extraction and training, LSTM algorithm is used for extraction of the information from the dataset, and features are chosen randomly. The third stage consists of the testing stage after random selection the features which will validate the model for the training process to complete with the help of LSTM algorithm. The fourth stage is the output stage. In this stage, they have predicted the value and plot the graph after training, and the testing process is done. The final stage is performance analysis that is examined based on the training accuracy and validation accuracy by using long short term memory algorithm.

III. METHODOLOGY

A Recurrent Neural network is the derivative of a feedforward ANN (artificial neural network). It consists of three different nodes i.e., the input layer nodes, hidden layer nodes and the output layer nodes as shown in Figure 2

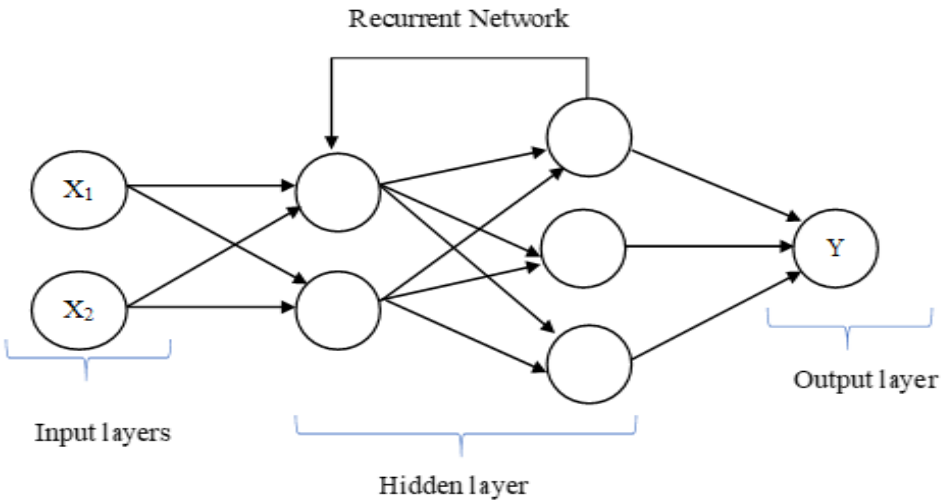

Figure2: Implementation of RNN

Apart from the input nodes, the other nodes act as the neurons which in turn uses the non-linear activation function. The algorithm uses a backpropagation supervised learning technique for training in order to predict whether an input belongs to a certain category or not. The algorithm acts as a linear classifier by drawing a straight line to classify the instances. Output of the algorithm y = (w*x) + b, that is a product of weight and feature vector x added with bias. Training involves adjusting the malware detection parameters or the weights and biases of the model in order to minimize error. It is done by the hidden layer and the final result class is calculated and displayed in the output layer. It has a lot of applications in the field of classification and regression problems. This model is suitable for datasets that are not linearly separable. Neural networks are suitable for sound analysis, handwriting recognition, and many similar applications. Simple RNN is implementation in Keras.

LSTM Architecture

LSTM has the same control flow as the recurrent neural network. It processes the data sequentially passing on information. As it propagates forward, the difference are the operations within the LSTM cells. The Figure 3 represents LSTM architecture. LSTM architecture consists of three different gates forget gate, input gate, and output gate that regulate information flow in an LSTM, these are just a layer of neurons where Xt and Ht are vectors that implies the list of numbers. Hidden state is also referred as a vector. Sigmoid operation like weighted multiplication is performed by using these vectors and activation function which is tan h in the case of RNN will get a new hidden in traditional RNN is applied. Xt is given as the input feed at the forget gate. Long-term memory is a problem that occurs in RNN when the network is in the need of making a prediction that requires context. In a regular RNN, the need of understanding the context can be handled this is only dependent on how far the memory needs to save the instruction for the context.

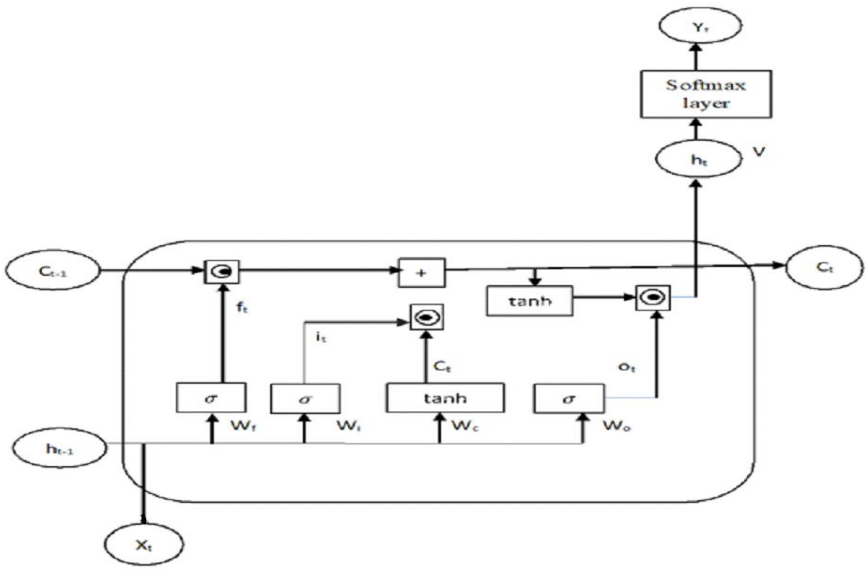

Figure3: LSTM Architecture

IV. IMPLEMENTATION

The detailed explanation of the design and the implementation of the model are described

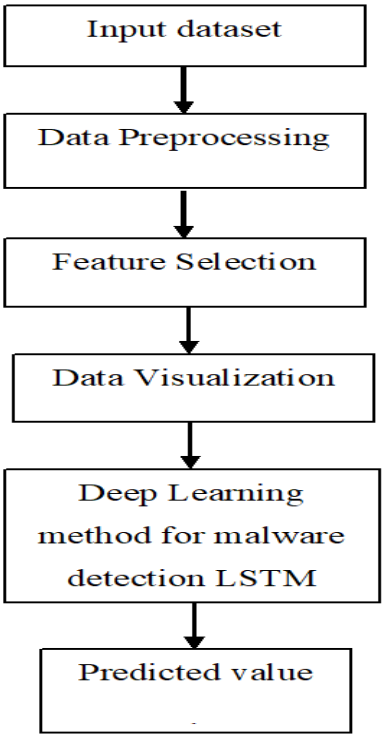

Figure4: Flowchart of Malware and Benign Prediction

The flowchart of the malware or benign prediction is as depicted in Figure4. The input dataset file includes various malware and benign samples. Input dataset consists of 35 features and load the dataset and preprocess the dataset for the best feature for the prediction to know which is malware or benign after several steps in the given flowchart. 100,000 dataset record attributes are taken from Kaggle which consists of 50 malware and 50 benign files.

i. **Data Pre-processing:** The Data Pre-processing stage utilizes 35 features from the dataset which are used for the classification of the data as malware or benign. The proposed system will modify and customize the dataset by dropping the 'hash' and 'classification' columns. The data from these columns are utilized to identify the file as malware or benign by creating a new column 'hash' and then assigning Boolean values 0 for malware and 1 for benign.

ii. **Feature Selection:** In the feature selection stage, the proposed system will pick up the top 5 features out of all the available features. These top features will be further used for classifying the data as malware or benign by utilizing the values from the label column. The top features are identified by using an extra tree classifier and extracted by using the panda's library.

iii. **Data visualization:** Data visualization is defined as a graphical representation that contains of the information and the data. The main aim of the data visualization process is to make it easier to identify the records, trends, and outliers in large datasets. After labelling malware and finding the best feature, the bar graph is plotted based on malware and benign dataset.

iv. **Deep Learning Method for Malware detection Long Short-Term Memory (LSTM):** After importing the necessary libraries then the data is split into training and testing. For training will take 8000 records and testing will take 20 % of records and reshape the data into training and testing for building the LSTM model. The model is trained with 50 epochs. After training the model, the test records based on test records are classified. The accuracy graph is plotted and the model is stored inside the files. The dataset model stored is made use of to predict the results.

v. **Predicted Value:** Precision is useful to predict the model's performance in terms of positive example classification. In contrast, to review, however, accuracy is worried about the number of the models the model named positive were really sure. To figure this, the quantity of genuine positive models is separated by the quantity of false-positive models in addition to genuine positives. After training and validating the data predicted and the accuracy at 98.89 percent is plot. 50 epochs will classify the test records and plot the graphs of model loss and model accuracy.

V. RESULTS AND ANALYSIS

This section displays the results of the proposed method LSTM experiment. The results that are achieved by these experiments are measured in terms of accuracy. The experiments are run numerous times in the training and testing for the dataset.

Figure 5 shows the 5 best out of 35 features are shown below for predicting if the given application is malware or benign.

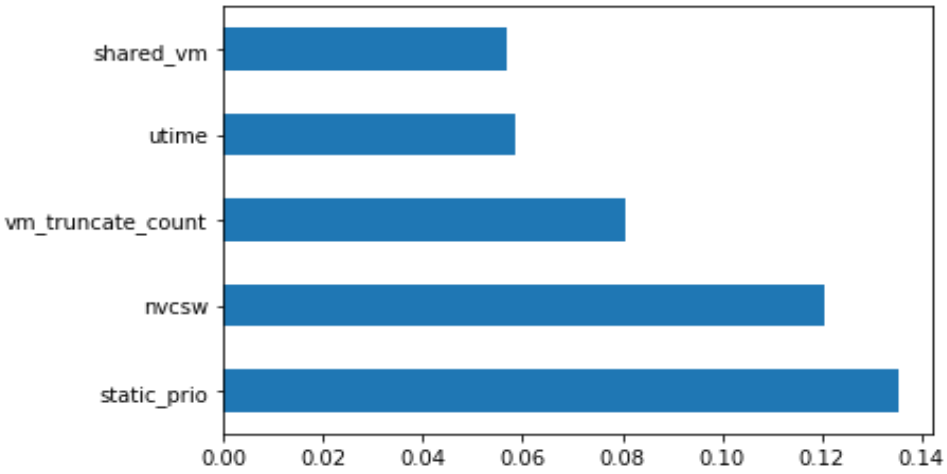

Figure5: Distribution of top five selected features from the dataset

The model loss obtained during the evaluation of dataset using the recurrent neural network machine learning algorithm is as shown in the Figure 6.

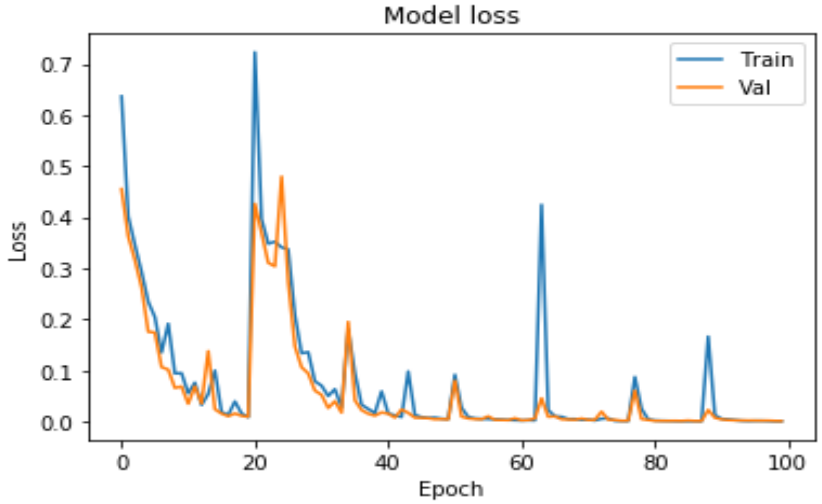

Figure6: Interpretation of Model loss for RNN algorithm

An accuracy of 98.89% was successfully achieved. Figure 7 shows the statistical graph of the model loss based on the training and validation processes.

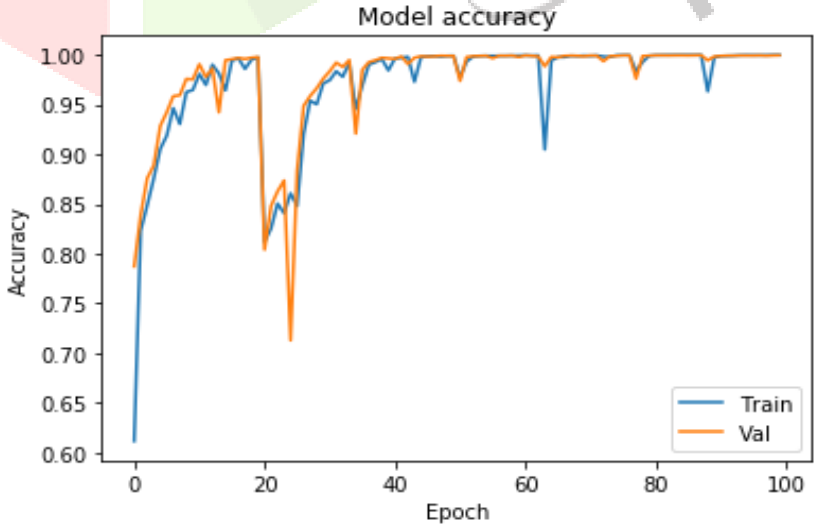

Figure7: Model accuracy for RNN algorithm

Table 1: F1-score is the measure of test results accuracy and it is a harmonic average of precision and recall

Table 8.1 Evaluation results of Recurrent neural network Algorithm

Parameters	precision	recall	f1-score	support
malware	0.98	1.00	0.99	9921
benign	1.00	1.00	0.99	10079
accuracy			0.99	20000
macro avg	0.99	0.99	0.99	20000
weighted avg	0.99	0.99	0.99	20000

VI. CONCLUSION AND FUTUREWORK

This study shows evaluation with the usage of deep-learning classifiers to accurately detect mobile malware by choosing the relevant malware features for classifier inspections as well as to identify the ideal classifier based on TRP values. The findings and accomplishments of the classifier were enormous. Deep learning classification system is evaluated in this research to enhance the malware detection solution of large collection of samples and to acquire the finest classification capable of detecting mobile malware. Recurrent neural network based long short-term memory are the classifiers opted. 98.89% accuracy of the detection rate with current classifier for the malware dataset is observed in the experimental study. The processing time is directly proportional to the dataset. In order to build the accurate detection model, longer processing time is utilized with larger dataset. This strategy shows that deep learning is productive and efficient in true malware operations.

In future,

i. Larger dataset can be utilized to improve the accuracy.
ii. Accuracy can also be achieved with the usage of more hidden dense layers.

REFERENCES

[1] Hemalatha, Selvabrunda, "Mobile Malware Detection using Anomaly Based Machine Learning Classifier Techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* ISSN: 2278-3075, Volume-8, Issue-11S2, September 2019.

[2] Mohammed K. Alzaylaee, Suleiman Y. Yerima, SakirSezer, "DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices", *Computers & Security* (2019),

[3] Hoch Reiter and Schmid, "Evaluation of recurrent neural networks for crop recognition from multitemporal remote sensing images, "*Fotogrametria e Sensoriamento Remoto, Rio de Janeiro,* Nov,2017.

[4] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network neural computation", *Massachusetts Institute of Technology, 1235–1270, 2019.*

[5] Zheng, M. Sun, M.& Lui, "Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malware", *In 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (2013, July). (pp. 163-171). IEEE.*