International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-09, Issue-01, Apr 2023

LSTM based Malware Detection Framework for Android Devices

[1]Prof. Kanchan Umavane, [2]Mr. Prathamesh Avhad, [3]Mr. Rutik Tavanoji, [4]Ms. Rupali Pardeshi.
[1]Asst.Professor,[2,3,4]UG Student,[1,2,3,4]Computer Engg. Dept. Shivajirao S. Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India.
[1]kanchanumavane2020@gmail.com,[2]prathamavhad2001@gmail.com,
[3]rutiktavanoji2000@gmail.com, [4]pardeshirupali888@gmail.com

Abstract- Lately, the matter of dangerous malware in devices is spreading speedily, particularly the repackaged android malware. Though understanding Android malware detection via dynamic analysis will give a comprehensive read, and need to also relate between the app's features and the features that are needed to deliver its category's functionality. The most popular freely accessible smartphone operating system is Android, yet it lacks virus detection in its permission declaration and access control systems. The matter of investigating such malware presents distinctive challenges thanks to the restricted resources accessible and restricted privileges granted to the user. But, each APK file also offers unique possibilities for the necessary information connected to each programme. By integrating permissions and API callas characteristics to describe malware, as well as using machine learning methods to automatically extract patterns to distinguish between benign and malicious Apps, and objective is to create a solution that effectively reduces the danger of Android malware. This model provide a machine learning-based method for malware detection on Android-powered devices. This model uses an intuitive user interface to efficiently identify, detect, classify, and protect Android mobile devices against harmful applications, preventing any data theft or misuse. This research aims to develop an LSTM rule-based malware detection system based on code behaviour signatures. It will be able to identify malicious code and its variations successfully in runtime and expand malware characteristics data dynamically. For static features and dynamic actions, machine learning techniques are the current approaches to Android malware patterns, and testing results show that the methodology combines a high detection rate and a low rate of false positives and false negatives. Recurrent neural networks are adjusted to create Long Short-Term Memory (LSTM) networks, which facilitate better memory retention for prior information [1].

Keywords- Android Malware, Machine Learning, Long Short-Term Memory

I. INTRODUCTION

The problem of harmful malware in smartphones is currently increasing quickly, especially the spyware that has been packaged to seem like Android. Although comprehending how to identify dynamic analysis is used by Android malware will provide a thorough read, and also need to associate between the app's features and the features required to provide the functionality of its category. The most popular freely accessible smartphone operating system is Android, yet it lacks virus detection in its permission declaration and access control systems. Due to the limited resources available and the limited rights allowed to the user, analysing this virus poses unique hurdles [7]. But, each APK file also offers unique possibilities for the necessary information connected to each programme. The objective is to create a system that effectively reduces the risk of combining permissions and API calls as features to define malware, appropriately identifies and mitigates any malicious APKs, and automatically extracts patterns to distinguish between benign and malicious Apps using machine learning techniques [8]. This model provide a machine learning-based method for the Android malware detection devices. This model uses an intuitive user interface to efficiently identify, detect, classify, and protect Android mobile devices against harmful app, preventing any data theft or misuse. This research aims to develop an LSTM rule-based malware detection system based on code behaviour signatures. It will be able to identify malicious code and its variations successfully in runtime and expand malware characteristics data dynamically. Machine learning approaches are the current ways to model patterns of static attributes and dynamic behaviours of Android malware, based on experimental findings. The approach has a low rate of false positives and false negatives and a high detection rate. Recurrent neural networks are adjusted to create Long Short-Term Memory (LSTM) networks, which facilitate better memory retention for prior information [2].

II. AIM AND OBJECTIVE

a) Aim

The Purpose of this paper is to implement malware detection for android devices. Android malware detection is to identify and prevent malicious software from infecting Android devices and potentially causing harm to users.

b) Objective

- Create an Android Application that displays all the installed Android Apps and extracts theselected APK to cloud.
- Test the desired APK via static (permissions, recievers, services) and deep dynamic (API Calls) approach.
- Create a report using the findings and results and send it to the user in a PDF form successfully displaying the conclusion of the scan and categorizing the App in an appropriate manner.

III. LITERATURE SURVEY

Paper 1: Android Malware Detection Based on Deep Learning

Neural network (NN) is a kind of mathematical model which is consisted of many neuron layers. Recurrent Neural Network (RNN) is a typical structure of NN, and it has a special memory unit which can retain the state information of previous hidden layer. RNN shows good results in various fields which use sequential data such as natural language processing and speech recognition. Several studies of RNN for malware detection have been published. While EI-Bakry suggested using neural networks with a time delay for malware classification, the study made no attempt to test the theory through experimentation. Pascanu et al. RNN has been suggested as a malware detection technique. Nonetheless, like the original capability for malware detection, requires API calls. Process behaviour is used by Tobiyama et al. to develop a deep neural network technique for malware detection. Yet because the disappearing and increasing gradient issue was identified, RNN became unpopular until the LSTM was proposed [4].

Paper 2: Enhanced Android Malware Detection: An SVM-Based Machine Learning Approach

A linear classifier that builds a fundamental model underlies the SVM, a two-class model. The interval maximization in the eigenspace. Meanwhile, it can also solve the nonlinear problem employing kernel trick. The convex quadratic programming problem also known as the maximum edge algorithm that the SVM learns to solve tries to maximize the interval, whose advantage lies in strong generalization ability, which can solve the issues of nonlinear, small samples, high dimension, etc. Taking the linear separable SVM as an example, the principle of SVM is to search for a separable hyperplane in given eigenspace and then divide the sample space into two categories, one is a positive class and the other is a negative class, corresponding to two different categories of samples. The equation where is the normal vector and is the intercept may be used to describe the hyperplane in a support vector machine [3].

Paper 3: Android Malware Detection using LSTM

Long short-term memory (LSTM) is a modified network of RNN proposed to learn long-rangedependencies across time-varying patterns. In general, LSTM is a second order recurrent neural

network that addresses the problem of disappearing and exploding gradients by substituting memory blocks in the recurrent hidden layer for RNN simple units. In the LSTM, a memory block is a multi-unit complicated processing unit. It is made up of adaptive multiplicative gating units and one or more memory cells. (input, output and forget) and a self-recurrent connection with a fixed weight 1.0. It serves as a short-term memory with a control from adaptive multiplicativegating units. The input and output flow of a cell activation of a memory cell is controlled by input and output gate respectively [1].

IV. EXISTING SYSTEM

SVM was applied previously to detect android malware. SVM is a combination of decision tree and support vector machine algorithms. SVM relies on feature extraction, which involves extracting specific characteristics of the Android application, such as the permissions requested, API calls made, and network behaviour. However, it may not capture all of the relevant information, particularly in cases where the behaviour of the malware is time-dependent [3].

VI. PROBLEM STATEMENT

To create an efficient system which curbs the danger posed by android malware by correctly detectingand mitigating any malicious APKs via combining API calls and permissions are features to characterize malware, and apply techniques for machine learning automatically extract patterns to distinguish between good and bad apps.

VII. PROPOSED SYSTEM

Import data in the form of CSV file using pandas framework. Using train test split divide entiredataset in a ratio of 1:3 (75% of data is for training and 25% for testing). Design a model whilekeeping the inputs in mind. Select ‘sigmoid activation function’ as input/output is binary. Analyze accuracy using confusion matrix. This was the methodology is applied. This model was developed using a very effective dataset, comprehensive requirements, and a range of apps with several APKs [5]. SVM is known for classification so it was first algorithm to go towards but during the testing and found out that the novel SVM approach had low accuracy on real time applications. Long Short Term Memory Network is an advanced RNN, a sequential network that enables persistent information. LSTM are specifically made to prevent long-term dependency problems. LSTM can analyse whole data streams, including data sequences, such as audio or video, in addition to single data points like pictures. Kyunghyun Cho et al. presented gated recurrent units (GRUs) as a gating technique for recurrent neural networks in 2014. The GRU has fewer parameters than an LSTM since it doesn't have an output gate, but it is similar to an LSTM with a forget gate. GRU was shown to perform similarly to LSTM on several tasks including polyphonic music modelling, speech signal modelling, and natural language processing. This mainly compared these two algorithms GRU and LSTM where found LSTM to have an edge with respect to accuracy and hence preferred it as prime algorithm for model [1].

V. COMPARATIVE STUDY

Table 1: Comparative Analysis

Sr No.	Author	Project Title	Publication	Techniques	Purpose
1.	Jianming Zhang, Futai Zou, Junru Zhu	Android Malware Detection Based on Deep Learning	IEEE 2019	Recurrent Neural Network	The paper proposes a new architecture of Recurrent Neural Network (RNN) that can perform the detection process better than traditional machine learning algorithms.
2.	Hyoil Han, SeugnJin Lim, Kyoungwon Suh	Enhanced Android Malware Detection: An SVM-Based Machine Learning Approach	IEEE 2020	Support Vector Machine	SVM aims tocreate a decision tree based on the sample set and then update the decision node from the bottom up. Decision treewith SVM nodes.
3.	Satheesh Kumar Sasidharan,Ciza Thomas	Android Malware Detectionusing LSTM	IEEE2022	Long-Short Term Memory Framework	LSTM with 32 memory blocks containing one cell each has performed well on detection of all individual behaviors of malicious application in comparison to other traditional static machine learning classifier.

VIII. ALGORITHM

Step 1: Start

Step 2: Data preprocessing

Collect a dataset of Android apps that includes both malicious and benign apps. Extract a set of features from each app, such as API calls, permissions, and intent filters. Convert the feature sequences into numerical vectors using techniques like one-hot encoding, embedding, or other feature representation methods.

apkfile = flask.request.files[file_id]

filename = werkzeug.utils.secure_filename(apkfile.filename)

Step 3: Model training

Create an LSTM model with an output gate, an input gate, a forget gate, candidate cell states, updated cell states, and a fully linked layer with a sigmoid activation function. Train the model on the training set using the extracted feature vectors as input and the binary labels (0 or 1) as the output.

Permissions

a = APK(appUndertest)

perms=processPermission(a.get_permissions())

predictionList = [0] * len(setPerms)

permissionPrediction= predictionModel.predict(np.array(predictionList, ndmin=3)

Recievers

a = APK(appUndertest)

recievers= recProcess(a.get_receivers())

recieversPrediction= predictionModel.predict(np.array(predictionList, ndmin=3))

Services

a = APK(appUndertest)

services= serviceProcess(a.get_services())

servicesPrediction= predictionModel.predict(np.array(predictionList, ndmin=3))

Step 4: Evaluation

Evaluate the trained model on the testing set by feeding the feature vectors through the LSTM model and comparing the predicted labels to the true labels. Calculate performance metrics like accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) curve to assess the model's effectiveness in detecting Android malware.

_callee = callee.get_class_name() + '->' +callee.get_name()+ callee.get_descriptor()

CG.add_node(_callee)

Step 5: End

IX. MATHEMATICAL MODEL

The mathematical model of Long Short-Term Memory (LSTM) for Android malware detection involves training a recurrent neural network on a dataset of Android apps to classify them as either malicious or benign. The input to the LSTM model is a sequence of features extracted from the Android apps, such as API calls, permissions, and intent filters. These features are represented as a vector at each time step, where each element of the vector corresponds to a specific feature. The LSTM model processes this sequence of feature vectors using the following equations:

Input Gate (i_t):
$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i)$
Forget Gate (f_t):
$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f)$
Candidate Cell State ($\tilde{c_t}$):
$\tilde{c_t} = \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c)$
Updated Cell State (c_t):
$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$
Output Gate (o_t):
$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o)$
Hidden State(h_t):
$h_t = o_t * \tanh(c_t)$
where x_t is the feature vector at time t, h_{t-1} is the hidden state of the previous timestep, c_{t-1} is the cell state of the previous timestep,

and W_{xi}, W_{xf}, W_{xc}, W_{xo}, W_{hi}, W_{hf}, W_{hc}, W_{ho}, W_{ci}, W_{cf}, W_{co}, b_i, b_f, b_c, and b_o are the learnable parameters of the LSTM cell.

X. SYSTEM ARCHITECTURE

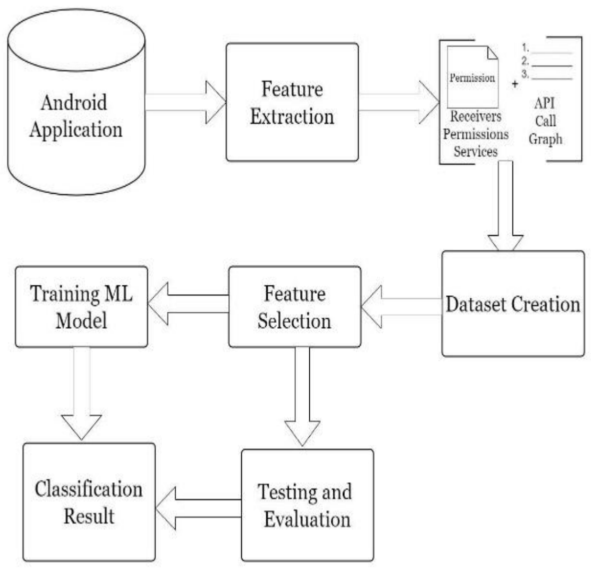


Fig. 1: System Architecture

Description:

Feature Extraction: Implement feature extraction by enumerating the API calls, permissions and activity of the APKs to get a fair idea of the behaviour of the particular apps.

Dataset Creation: Then, for further use and reference, generate a dataset based on the extracted characteristics by combining them into CSV files.

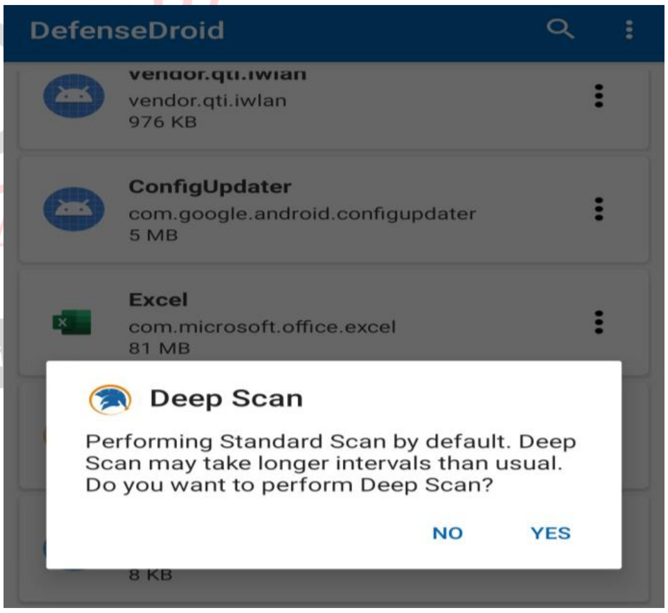
Feature Selection: After that select specific features which seem like an anomaly or a red flag, collect and store them in new CSV files and train Machine Learning model according to them.

Testing and Evaluation: Finally, last phase is to test model with the data and evaluate findings according to the results.

XI. ADVANTAGES

1. Effectively detect unknown malware.
2. Reducing the storage space usage Reduce the complex processing.
3. Detect malware accurately using less computational resources.
4. Less complex to analyse.
5. Faster feedback to the user.

XII. DESIGN DETAILS



DefenseDroid

vendor.qti.iwlan
vendor.qti.iwlan
976 KB

ConfigUpdater
com.google.android.configupdater
5 MB

Excel
com.microsoft.office.excel
81 MB

Deep Scan
Performing Standard Scan by default. Deep Scan may take longer intervals than usual. Do you want to perform Deep Scan?
NO YES

8 KB

Fig. 2: Scan Dialog Box

Description: This project share different application on different platforms. When users select Scan option, it pops up a dialogue box asking for Deep Scan. By default, this application is doing a Standard Scan which contain analysis of Permission, Receivers and Services while Deep Scan acts on API Calls.

XIII. CONCLUSION

Thus we have tried to implement the paper “Satheesh Kumar

Sasidharan and Ciza Thomas", "Android Malware Detection using LSTM", IEEE 2022. according to implementation, Conclusion as follows to use permissions, receivers and services of Android applications to detect malware and malicious codes in Android based mobile platform. This is a novel approach to distinguish and detect Android malware with different intentions. It is effective, that is, it is able to distinguish variant of Android malware betouren distinct purposes of them. The suggested framework takes Android application permissions and extracts and further combines the API calls to characterize each application as a high dimension feature vector. By applying learning methods to the collected datasets, this model can derive classification models to classify Apps as benign or malware. Experiments on real world data demonstrate the good performance of the framework for malware detection.

REFERENCES

[1] Satheesh Kumar Sasidharan, Ciza Thomas, "MemDroid: Android Malware Detection using LSTM", IEEE, 2022

https://ieeexplore.ieee.org/document/9686531

[2] Shakhnaz Amenova, Cemil Turan, Dinara Zharkynbek, "Android Malware Classification by CNN-LSTM", IEEE 2022

https://ieeexplore.ieee.org/document/9945816

[3] Hyoil Han, SeugnJin Lim, Kyoungwon Suh, "Enhanced Android Malware Detection: An SVM-Based Machine Learning Approach", IEEE 2020 https://ieeexplore.ieee.org/document/9070608

[4] Jianming Zhang, Futai Zou, Junru Zhu, "Android Malware Detection Based on Deep Learning", IEEE 2019 https://ieeexplore.ieee.org/abstract/document/8781037

[5] Haoyu Wang, Junjun Si, Hao Li, "RmvDroid: Towards A Reliable Android Malware Dataset with App Metadata" IEEE 2019 https://ieeexplore.ieee.org/document/8816783

[6] Mohammed K. Alzaylaee, Suleiman Y. Yerima, Sakir Sezer, DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices, Computers & Security (2019), doi: https://doi.org/10.1016/j.cose.2019.101663

[7] Gayatri Naik "Reliability and Security of Large amount of Data Storage in Cloud Computing", International Journal of Emerging Technology and Advanced Engineering, ISO 9001:2008 Certified Journal, ISSN 2250-2459, Volume3 Issue 2

[8] Prof. Vishal R. Shinde "Search rank fraud and malware detection in google play" in IJREAM, ISSN: 2454-9150, Volume 08, Issue 01, APR 2022 Special Issue https://sdbindex.com/documents/00000217/00001-75894.pdf