A novel approach for Android Malware Detection using Machine learning techniques

Mellempudi Anuradha [1], Mr. Enapakurthi Sateesh [2], Kaki Madhuri[3]
Student[1, 3], Assistant.Professor[2]
Amritha Sai Institute of Science and Technology Paritala-521180
Autonomous NAAC with A Grade,Andhra Pradesh,India

ABSTRACT

The presence of malware presents a major problem in both the operating system and software domains. The Android operating system is also included with these problems. The Android operating system is also included with these problems. Compared to other operating systems Android is one of the most used operating system in smartphones with a unique 2 billion users and has 74% market [1]. Other approaches, utilizing signature-based methods, have been employed for the detection of malware. Despite their application, these techniques were unable to detect unknown malware effectively. Even with the availability of various detection and analysis techniques, the issue of accurately detecting new malware remains a critical issue.

In a survey conducted in 2006 by Microsoft Company they have found 45000 variants of malwares like Trojan, backdoor and bots [9]. The focus of our research is to investigate and underscore the prevailing approaches employed in identifying and analysing malevolent code specifically designed for Android platforms. Alongside our study, we suggest the implementation of machine-learning algorithms for the analysis of such malware, complemented by semantic analysis methodologies. Through the use of ML, many procedures can be executed on interconnected data which involves classification, regression and clustering. ML algorithms are being used in malware detection techniques since many years [2]. In this paper we focus on a new android malware detection method which uses GUI.

INTRODUCTION

Many survey reports have shown that nearly I million malware files are being evolved every day and these malware files have become a step to many cybercrimes which would affect the world's economy. [3]. the key intention of malware is to impede, pocket or do some violations. Malware possesses the power to invade any kind of machine processing application. The detection techniques used for smartphones are trailing in comparison with the rapid rise of mobile user base. Android is the most widely used mobile operating system (OS). As of February 2023, its market share was72.26%. Based on the McAfee mobile threat report, there is a vast upsurge in backdoors, fake applications and banking Trojans for mobile devices [4]. The Google android market also offers no certainty that all the applications included are threat free. There have been reports indicating the presence of Trojan applications that, when

downloaded, secretly implant malicious code into devices.Android threats include SMS-Based Attacks, App Store Policy Violations, Phishing Attacks, bots. There are around 3.6 billion android users worldwide.

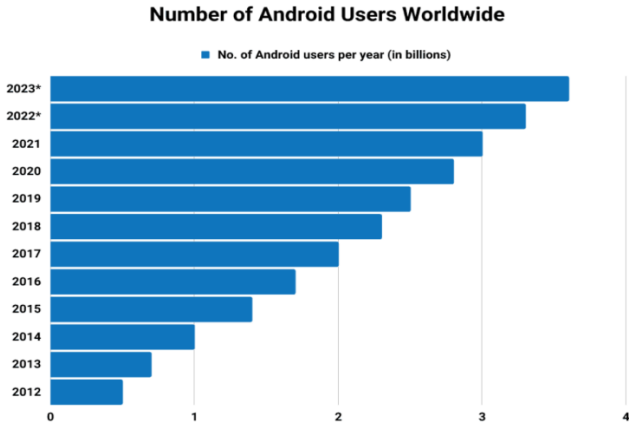


Fig: Number of android users

MALWARE ANALYSIS AND TECHNIQUES TO INSERT MALWARE

To understand how a malware works it needs to be examined. Malware analysis is the process of defining the working of malware and answers to the following questions [5,6]. How malware works, how a device can be affected, which machines and programs are affected, which data is being dented and filched, etc. There are different methods or channels through which malware applications can pass in your system. Some of the most communal practices of malware getting intervened into your system are:

A. PHISHING

Phishing is a type of social engineering attack that uses email or text messages to trick people into giving up their personal information. These attacks often involve fake websites that look like the real websites of banks, credit card companies, or other organizations. If you clack on a link in a Phishing email or text message, you could be taken to a forged website that may look like a real website. Once users enter their private information on the forged website, the invader can steal it.

B. SOFTWARE UPDATES

Software updates can contain malware occasionally so it is important to download software updates only from reliable sources.

C. DRIVE-BY DOWNLOADS

These are the type of attacks takes advantage of potential vulnerabilities in operating system, apps and applications. It refers to unintentional download of virus or malware onto your computer or mobile devices.

D. FILE SHARING

File sharing is a way to share files among the users. If you download file from any file sharing site, make sure you trust the source of file.

E. FAKE APPS

These applications usually pretend to be a real thing and tries to dupe the users to download these fake applications onto the targeted devices and thereby compromising the security of the devices. They try to act as legitimate apps and tries to trick users to install them.

F. AD-WARE

Some websites are peppered with different types of ads, which when clicked will redirect to certain webpages. While one goal of these ads is to make

revenue for these websites, but some of them are composed of different malware.

When you click on those ads you may involuntarily downloading them onto your device which can compromise the security of the devices.

G. BOTNETS

A bot-net is a network of compromised Android devices which is generally controlled by attackers.

OBJECTIVES

The objective of an android malware A detection system using machine learning aims to provide a fast, effective, and reliable way to protect Android devices from malware. This system utilizes the latest advances in machine learning and artificial intelligence.

- To provide co-operating dynamic user interface for users to find the correctness or maliciousness of the application.
- To protect users from installing malicious apps onto their devices.

CLASSIFICATION ALGORITHMS

Support vector machine

Support vector machine is one of the supervised machine learning algorithms, which is used to analyse the data which is used for classification analysis. It is used in linear, non-linear classification, regression, image classification tasks.

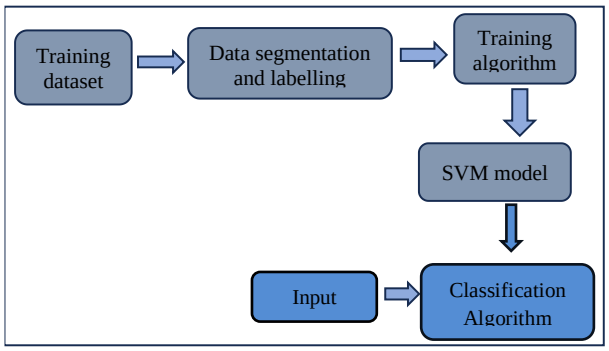


Fig: SVM Algorithm block diagram

Neural Networks

Neural networks are very efficient for classification and clustering tasks. They are composite models which tries to impersonate the way the brain of the human who develops different classification rules. A neural network consists of many diverse layers of neurons, where each layer receives inputs from preceding layers, and passes the result to successive layers.

I have used two algorithms SVM and Neural networks each of them gave an accuracy of 89 and 92 percentages.

FLASK

It is a web framework in python in order to create a dynamic user interface to users. It is used in developing web applications implemented on Werkzeug and Jinja2.

Table of Accuracy predictions

ML Algorithm	Accuracy (%)
SVM	89
Neural Networks	92

DATASET

In our study we have choosen (CICInves AndMal2019) to perform our experiment. This dataset contains more information about malware variants and malware classification [10].

EXISTING SYSTEM

Many solutions that use machine learning to identify Android malware already exist. Here are a few illustrations:

Droid-Detector: Droid-Detector is an Android malware detection tool that analyzes and categorizes Android apps using machine learning techniques. To find malware, the system combines static and dynamic analysis.

Drebin: Drebin is an Android malware detection technology based on machine learning that uses static analysis methods to spot risky code in Android applications. The system analyzes the application's code and employs a set of characteristics to determine if it is malicious or safe to use.

AndroGuard: A machine learning-based malware detection module is part of the open-source toolbox for Android malware analysis known as AndroGuard. In order to detect malware, the system combines static and dynamic analytic approaches.

Deep Droid: Deep Droid is a machine learning-based malware detection solution for Android that examines Android apps using deep neural networks. To determine if an application is malicious or not, the system pulls characteristics from its source code and trains a deep learning model on them.

DroidSVM: DroidSVM is a machine learning-based Android malware detection solution that employs support vector machines (SVMs) to categorize apps as harmful or benign. To extract features from the application's code and train the SVM model, the system combines static and dynamic analysis methods.

All these existing models are mostly subscription based and some of them are free. Also, these models are not proven to be efficient by most researchers. From those available models we have tested 200 apps and websites that are available over the internet and came to a conclusion that most of the available malware detection software’s are used to find the existence of malware in android OS and in-an app by extracting a string, which is a package name, this package names will be in the android manifest file of every app. Then those extracted strings are compared with the existing dangerous package names [7]. In the existing system the applications were analysed to detect the malware using command line prompt. An appropriate GUI (Graphical User Interface) was not there to perform tasks. It was problematic for unspecialised users to use the system. The user would execute all the commands using command prompt. GUI is easy to use.

PROPOSED SYSTEM

In our framework we have enforced a panel that have contact to upload the apk files and its

particulars accompanied with its classification. In the panel user can upload an apk file and an output will be displayed containing the type of file whether it is benign or a malware along with accuracy of the algorithm selected for test.

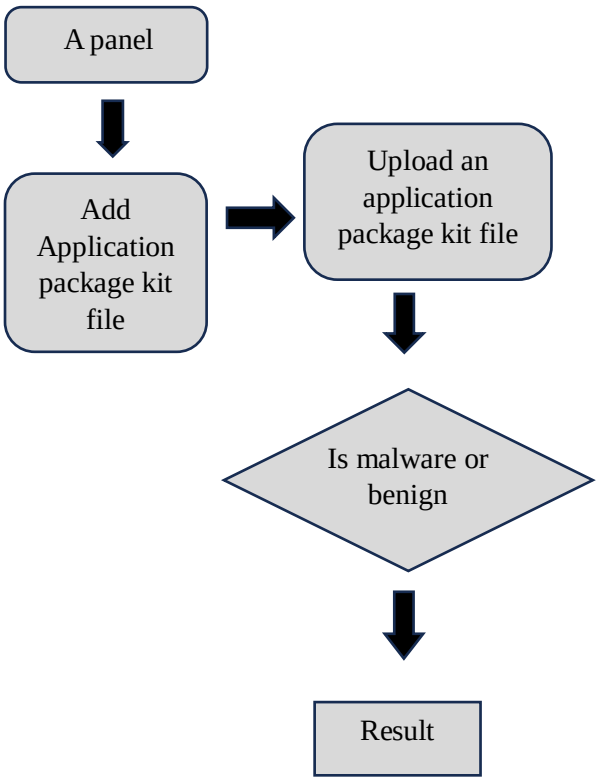
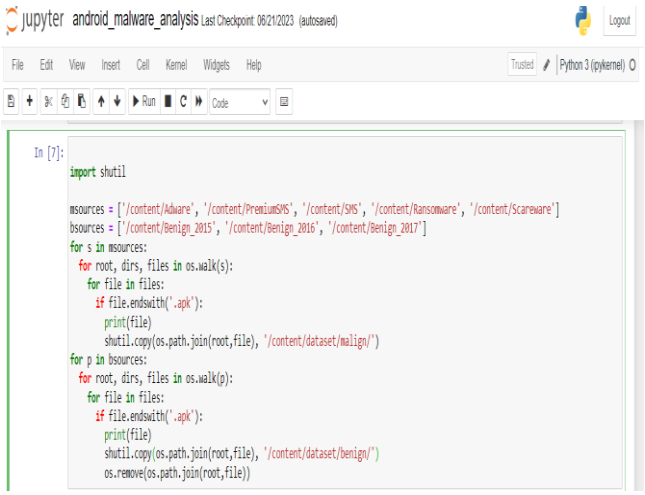


Figure No -1 Block diagram of the panel



```
In [7]:
import shutil

msources = ['/content/Adware', '/content/PremiumSMS', '/content/SMS', '/content/Ransomware', '/content/Scareware']
bsources = ['/content/Benign_2015', '/content/Benign_2016', '/content/Benign_2017']
for s in msources:
  for root, dirs, files in os.walk(s):
    for file in files:
      if file.endswith('.apk'):
        print(file)
        shutil.copy(os.path.join(root,file), '/content/dataset/malign/')
for p in bsources:
  for root, dirs, files in os.walk(p):
    for file in files:
      if file.endswith('.apk'):
        print(file)
        shutil.copy(os.path.join(root,file), '/content/dataset/benign/')
        os.remove(os.path.join(root,file))
```

Sample code that separates the android package kit files based on whether they are malware or benign.

```
In [0]: from keras.models import Sequential, Model
        from keras.layers import Dense, Dropout, Activation, Embedding, LSTM
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import classification_report, confusion_matrix

        import numpy as np
        from sklearn.preprocessing import LabelEncoder

In [0]: Y = dataset_df['class']
        X = dataset_df.drop(['class'], axis=1)
        encoder = LabelEncoder().fit(Y)
        Y = encoder.transform(Y)
        #print(encoder.transform(['malign']))
        X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)

In [0]: AN = Sequential()
        AN.add(Dense(256, activation='relu', input_dim=428))
        AN.add(Dropout(0.2))
        AN.add(Dense(128, activation='relu'))
        AN.add(Dropout(0.2))
        AN.add(Dense(128, activation='relu'))
        AN.add(Dropout(0.2))
        AN.add(Dense(32, activation='relu'))
        AN.add(Dropout(0.2))
        AN.add(Dense(1, activation='sigmoid'))
        AN.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])

        AN.fit(X_train, y_train, epochs=100, batch_size=32)

        scores = AN.evaluate(X_test, y_test)
        for i in range(len(scores)):
          print("\n%s: %.2f%%" % (AN.metrics_names[i], scores[i]*100))
```

A sample code for Multilayer Perceptron (Simple Artificial Neural Network)

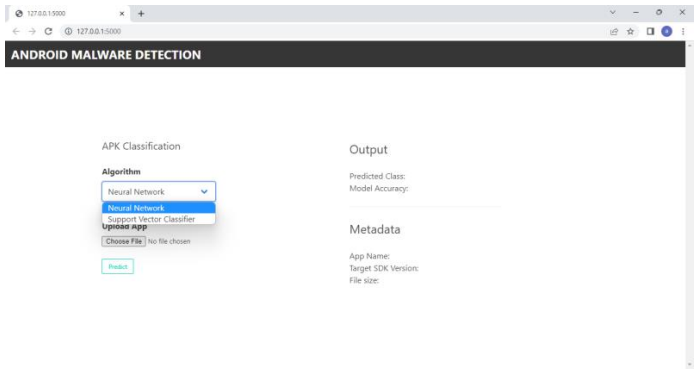


Figure No -2 GUI of panel (Add apk)

In the above figure we have shown the Graphical user interface (GUI) of our panel where the user can upload files and view the result. Once the user uploads apk file and selects any particular algorithm the result will be displayed along with the accuracy to the user.

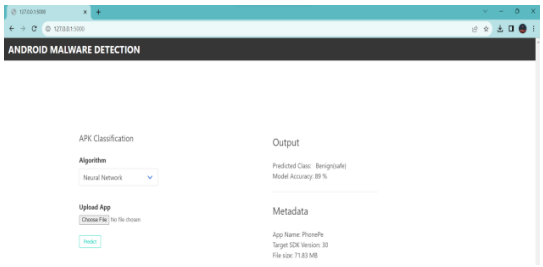


Figure No -2.1 Detecting Safe App using Neural Networks

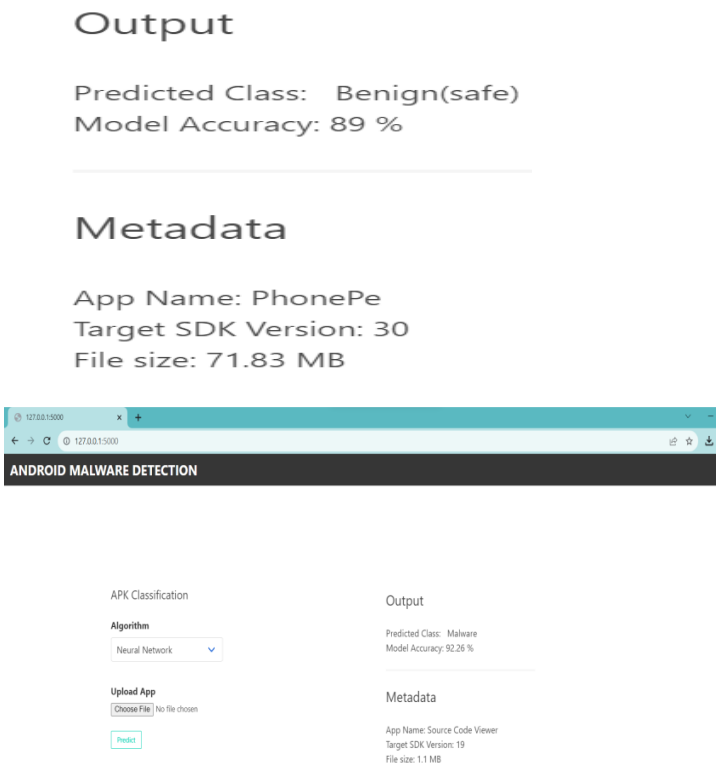


Figure No -2.2 Detecting Malware App using Neural Networks

CONCLUSION

In our supposition, we have built a framework for effectively detecting Android-related malware that combats the ever-growing malware variants that are being generated. Also, this type of framework is easy to use and increases user experience. It helps users to detect whether an application is malware or safe before installing them on their devices. By sharing this information among cyber security experts and data scientists it will be helpful in identifying new malware threats and potentially unsecured apps.

FUTURE WORKS

In the field of Android malware detection using machine learning, there is still much room for improvement and development. Future research could explore the use of other machine learning techniques in combination with GUI for Android malware detection. Additionally, the proposed approach could be tested on a larger dataset to evaluate its scalability and robustness.

There are limitations in our system, such as it does not provide precise details about the malware detected, and the user can upload only a single app to check the result. This is called static analysis.

Future work will focus on enhancing these limitations by implementing a dynamic analysis model. In this model, the system will have access to every app on the device instead of relying on the user to provide input.

REFERENCES

[1] An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms Ahmed S. Shatnawia, QussaiYassenb, Abdulrahman Yateem

[2] Darus, Fauzi Mohd, Salleh Noor Azurati Ahmad, and AswamiFadillah Mohd Ariffin.” Android Malware Detection Using Machine Learning on Image Patterns”2018 Cyber Resilience Conference (CRC). IEEE,2018.

[3] S. Morgan, “2019 Cybersecurity Almanac:100 Facts, Figures, Predictions and Statistics”, Cisco and Cybersecurity Ventures, [Online]. Available: https://cybersecurityventures.com/cybersecurityalmanac-2019.

[4] R. Samani and G. Davis, “McAfee Mobile Threat Report Q1”, 2019 [Online]. Available: https://www.mcafee.com/enterprise/enus/assets/reports/rp-mobile-threat-report-2019.pdf.

[5] Y. Alosefer, "Analysing web-based malware behaviour through client honeypots," Diss.

Cardiff University, 2012.

[6] M. Sikorski and A. Honig, "Practical malware analysis: the hands-on guide to dissecting malicious software," no starch press, 2012.

[7] International Journal of Research in Engineering, Science and Management Volume 5, Issue 1, January 2022.https://www.ijresm.com/

[8] Microsoft, “Microsoft security intelligence report”, http://www.microsoft.com/technet/security/default.mspx, July December 2006.

[9] Taheri, L., Kadir, A.F.A., Lashkari, A.H., 2019. Extensible android malware detection and family classification using network-flows and Api-calls, in: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE.