Software Proposal Document for Android Malware Detection

Amr Ehab, Mostafa Ashraf, Omar Shereef, Seif ElDein Mohamed

October 25, 2020

Proposal Version	Date	Reason for Change
1.0	18-October-2020	Proposal First version's specifications are defined

Table 1: Document version history

GitHub: https://github.com/OmarShereef/Graduation-Project.git

Abstract

Nowadays, the mobile industry is in rapid evolution making smartphones available with affordable rates for all segments of society. Smartphones' purposes are not limited to making phone calls or sending messaging, users can also take photos, store personal data, do online banking and trace their daily activities. The more applications appear, the more security becomes a concern to mobile users. This concern arises from the fear of being subjected to a security breach that jeopardizes confidential personal data such as emails, passwords, location, credentials etc. Malware applications which are developed for the sake of compromising users' personal data are also increasing rapidly day after day. In our work, we aim to design an intelligent detection framework for android malware applications. The framework uses different analysis-based approaches along with different machine learning algorithms to distinguish between benign and malicious applications.

1 Introduction

Using smartphones has become an indispensable part of our daily lives. By December 2018, Android occupies 75.16% of the market share of mainstream portable working framework [1].

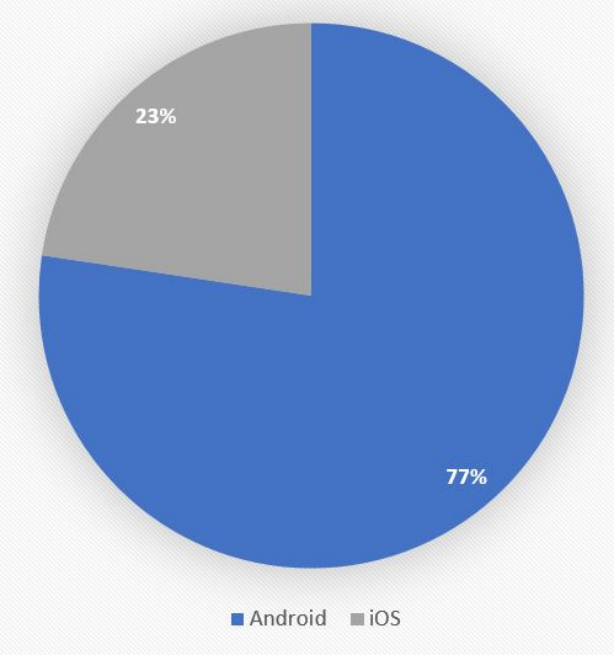

Figure 1: Most popular mobile operating system

More than one million Android applications such WeChat, TikTok, and mobile banking apps are used in our daily lives and keep on playing a progressively significant role found in Android markets such as Google Play. The vast majority of these applications have breached the users' private data, for example, their location, credit cards, and contacts data. Practically all applications can access the users' private information, in spite of the fact that this gives users with better-customized administrations. It might likewise result in data spillage of private information and financial misfortune. Moreover, android malware applications continue rising perpetually and this security problem has widely expanding consideration in the industry and academic fields. A huge group of researches against malware has been conducted. As of now, static analysis and dynamic analysis are the two primary kinds of identification techniques. Each approach has its benefits and deficiencies. Static analysis techniques, for example, PApriori [2], and DREBIN [3] examine applications without perform the program requiring low overheads. Be that as it may, the strategies cannot protect against anti-decompiling and obfuscation. On the other hand, analysis techniques, for example, VetDroid [4] run the application in realtime to detect malicious one, yet it is hard to secure all the execution ways. With malware being quickly developed, the machine learning technique is utilized to perform android malicious detection. Therefore, gathering features that better represent malicious behavior as the features of machine learning is helpful to improve the presentation of malware location.

1.1 Background

Malware operations are extremely reference to getting to users' private data by stealing, spying and showing of the regrettable ad. Malware is included within malicious software and it is frequently indicate to as software program that consciously own the deep attributes of malware aggressors and describes its malicious aim. Various kinds of malware are depicted in Figure 2 based on their different purposes,and methods of infiltration.

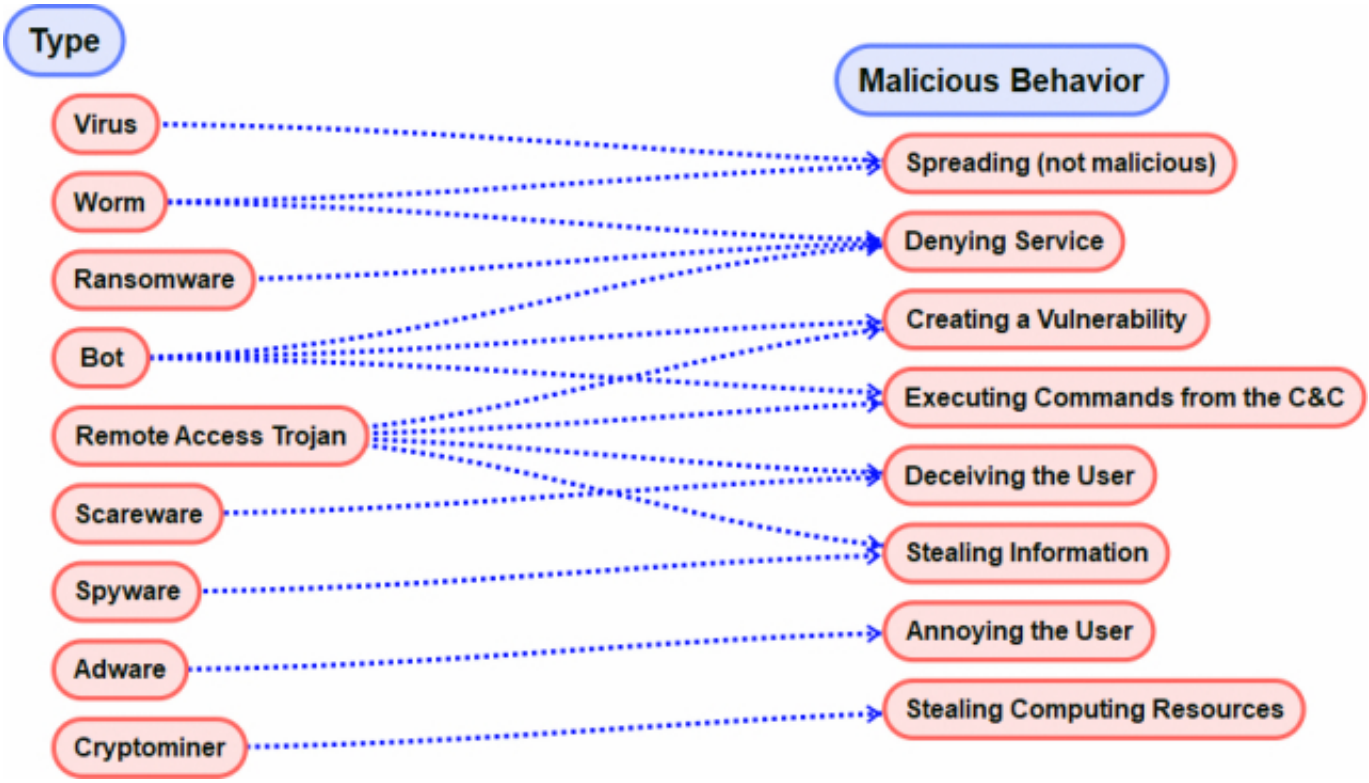

Figure 2: Types of malware

Number	Type	Definition
1	Worms	Worms are spread via software vulnerabilities or phishing attacks.
2	Trojan Horses	Just as it sounds, a Trojan Horse is a malicious program that disguises itself as a legitimate file.
3	Virus	Unlike worms, viruses need an already-infected active operating system or program to work.
4	Scareware	Scareware is a malware tactic that manipulates users into believing they need to download or buy malicious, sometimes useless, software. Most often initiated using a pop-up ad, scareware uses social engineering to take advantage of a user's fear.
5	Spyware	Spyware is unwanted software that infiltrates your computing device, stealing your internet usage data and sensitive information.
6	Ransomware	Ransomware denies or restricts access to your own files.
7	Bots	A bot is a computer that's been infected with malware so it can be controlled remotely by a hacker.
8	Adware	Adware is unwanted software designed to throw advertisements up on your screen,
9	cryptominer	Cryptojacking is the unauthorized use of someone else's computer to mine cryptocurrency.

1.2 Motivation

1.2.1 Academic

Unfortunately, the popularity of android and it is facilities gives to develop and transfer applications with harmful side. Moreover, the variety of android markets favors the presence of rebel markets really track their effectiveness and have appeased even more the improvement of an enormous malware environment. In this light, Android has gotten one of the most important focuses for malware developers, our work is motivated by minimum 49 malware families have been identified [5]. Therefore, detection of malware with ordinary techniques becomes unwieldy, which represents the need to build up a novel and proficient methodology for detecting malicious applications.

1.2.2 Business

Due the using of android framework in industry, company's like OnePlus target to build security system to avoid malware attacks. This research aims to improve the applications that detect malicious ones. Such as DroidKungFu, this piece of malware is unique in that it is able to avoid detection by anti-malware software, according to the Wall Street Journal [4]. It installs a backdoor in the android OS that allows hackers to gain full control over a user's mobile device. And a lately malware attack "CovidLock" ransomware is an example. This type of ransomware infects victims via malicious files promising to offer more information about the disease. The problem is that, once installed, CovidLock encrypts data from android devices and denies data access to victims. To be conceded access, you must pay a ransom of USD 100 per device.

1.3 Problem Statement

Malicious attacks are increasingly appearing. Which intentionally developed to infiltrate or damage the system without the consent of the owner. And smartphones' purposes are not limited to making phone calls or sending messaging, users can also take photos, store personal data, do online banking and trace their daily activities.

2 Project Description

2.1 Objectives

- To use different machine learning algorithms to detect malicious Android applications based on their runtime behaviour.
- To secure the personal data information.
- To provide firewall from breaching android users' critical data.

2.2 Scope

The proposed system identifies malware with efficiency and viability. Utilizing machine learning to naturally identify malware in all file types.

2.3 Project Overview

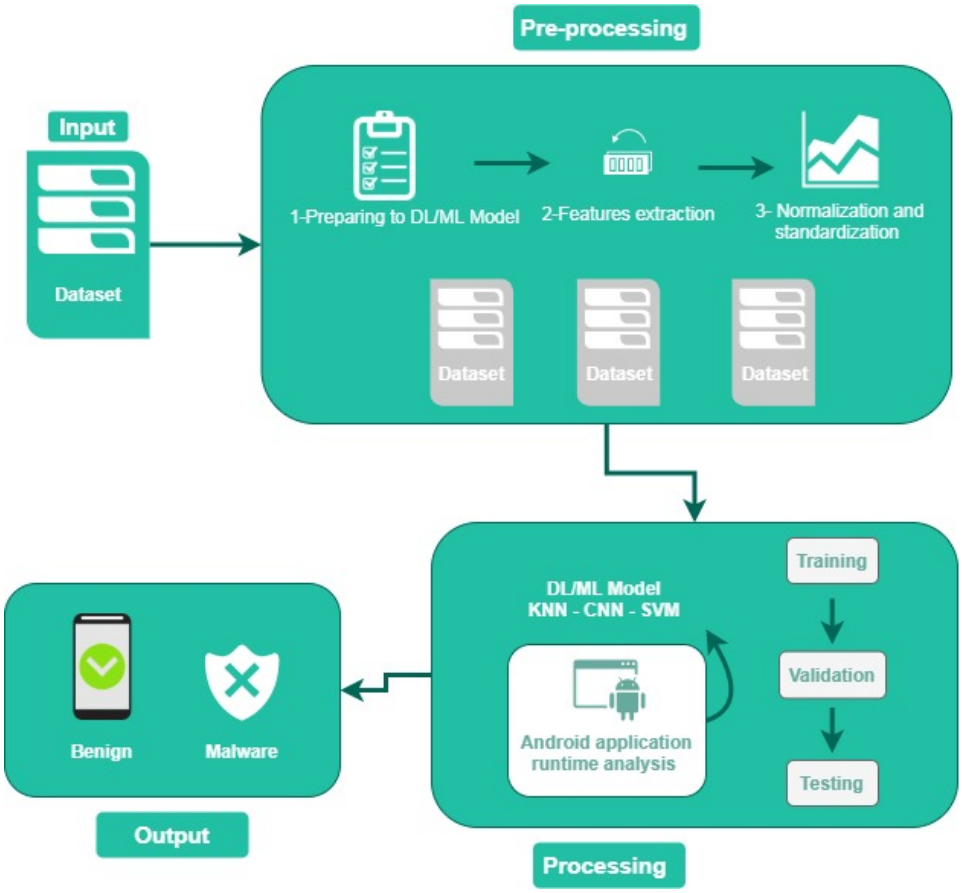

Figure 3: Project Overview

First we prepare our dataset to fit the different DL/ML model next we extract the interested featured only, then we perform normalization technique, standardization if needed. Then splitting our dataset to three subset , training, validation and testing respectively. In the processing stage we focus on training the different models using training set, next adjust the hyper-parameters of the model with the aid of validation set, then testing the trained model using testing set. Also we are performing behavior analysis of Android application in runtime. In the output stage we distinguish between benign and malicious application.

2.4 Stakeholder

2.4.1 Internal

1. Seif ElDein Mohamed is the **Team Leader**
2. Amr Ehab
3. Mostafa Ashraf
4. Omar Shereef

We split into two groups Mostafa and Seif work on the proposal, while Amr and Omar implement a demo for the proof of concept.

2.4.2 External

Android users and Android based operating systems vendors.

3 Similar System

3.1 Academic

Faiz[6], proposed a system that detects android malware applications by hybrid classification with K-means clustering algorithm and support vector machine (SVM). The researchers used two datasets. From the first dataset, they generate two datasets Data1 and Data2. The Data1 consists of 13,176 applications to train the model and a test set of 1860 applications. The Data2 consists of 12,028 applications to train the model and a test set of 3008 applications. The second dataset consists of 230 colluding app-pairs. They assume that colluding app-pairs can execute each one of those dangers that are executed by android malware. Then they use the parameter vector and a simple decision function to detect application collusion.

Parnika Bhat[7], proposed a system that detects malicious android applications based on naive bayes model. They got two datasets DREBIN and PRAGuard datasets that contain 2870 applications. From 2870 applications they got 1472 malicious applications and 1398 benign applications. The researchers solve the problem by malware detection technique called MaplDroid uses a static analysis approach with naive bayes model analysis. Also, they applied another classification technique such as random forest to bring out comparative analysis. MaplDroid achieved a score of 99.12%.

Umme Sumaya Jannat[8], proposed a system that do analysis and detection of malware in android using machine learning. The researchers solve the problem in two ways by dynamic analysis and static analysis. They get the best result in the dynamic analysis by Random Forest (RF) algorithm that is an extended version from Decision Tree (DT) algorithm. The result of dynamic analysis has exceeded the static analysis accuracy scores over 93% accuracy. Also, the researchers have used different datasets for static analysis and dynamic analysis. They used MalGenome dataset for static analysis and it consists of around 360 applications assembled by their malware families and another dataset from Kaggle that contains 4000 malicious applications in JSON format. And in dynamic analysis, they used a MalGenome dataset than contain 1260 malware applications belonging and classified to 49 different malware families.

Zhuo Ma[9], proposed a system that makes a detection for malware android applications. The researchers used control flow graphs and machine learning algorithms. They build chronological datasets and train it by making a Long short-term memory algorithm which is a recurrent neural network. The researchers de-compile and set up 3 kinds of systems: API usage datasets, API frequency datasets, and API sequence datasets. Then, they achieve 98.98% detection precision.

Muhammad Murtaz[10], proposed a system that scope to detect android malware applications. They solve the problem by using 6 algorithms K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (J48), Neural Networks (NN), Naïve Bayes (NB), and Random Forest (RF) on the dataset they get and test it on waikato environment for knowledge analysis. They used a dataset called CICAndMal2017 that have 10854 data (6500 generous and 4354 malware) and the dataset grouped into four noteworthy classifications (Adware, Ransomware, Scareware, SMS Malware). They show in this research that they detect malware location in 9 movements to accelerate the productivity of activity classifier. Also, the model uses gathering methodologies including stream-based, bundle-based, and time-based highlights to describe malware families. The assessment exhibits the proposed incorporate set has more than 94 significant for real

malware acknowledgment frameworks.

Ridho Alif Utama[11], proposed a system that makes analysis and classification of danger level in android applications. The researchers used Naive Bayes (NB) algorithm to detect if the android application is dangerous or not. The dataset that they used contains 188,389 data. They analyzed this research on permissions and vulnerabilities. Furthermore, from this technique, they can distinguish that the android application is safe or not. Lastly, they get this research accuracy is 97.2%.

Hongbing Yan[12], proposed a system that detects malicious android applications using machine learning. The researchers got two datasets which are Malgenome and virus share. Also, they downloaded over 1000 applications from Google Play to test these applications. They solve the problem by collecting runtime logs from each application. By utilizing the data extricated from the logs. The researchers got a result with a false-positive rate can be below 8% and a true positive rate over 90%.

Bahman Rashidi[13], proposed a system detection for android malicious applications by support vector machine and active learning. The researchers took a log of apps by their own developed instrumentation tool called DroidCat after capturing the logs a filtering and parsing mechanism is applied. They used a dataset called Drebin Project. The dataset has more than 5000 applications from 179 malware families they select 500 applications from malicious applications and 500 from benign applications to make training on it using RBF (Radial basis function) Kernel and 200 malicious applications and 200 benign applications to make tests on them through the researchers' model. The running time per application to be 2-5 minutes that took 79 hours for all applications. Their exploratory outcomes exhibit that their proposed model accomplishes fulfilling precision regarding true positive and false-positive rates and adjusts the recognition model for new malware patterns. Also, they used Optimal Query Strategy (QQS) that gets a high impact on the model performance and accuracy.

Yuxia Sun[14], proposed a system that detecting malware android applications based on extreme learning machine. The researchers collected a dataset from Tencent YingYongBao store. They used 524 benign applications and 525 malicious applications to test on them. To distinguish the android malicious application, they get permissions and API calls of application and by using extreme learning machine. The results show their work excels with the current ones with minimal human intervention, better detection efficiency, and less detection time.

Nathaniel Lageman[15], proposed a system that detects malicious android applications from runtime behavior. The researchers used logcat and strace outputs to produce runtime datasets. They get the dataset from North Carolina State University's android malware Genome project. They test the dataset using both Random Forest (RF) classifier and support vector machine (SVM). In results, they view that Random Forest classifier performs better than support vector machine with a true positive rate that exceeding 90% and a false-positive rate less than 6%.

4 What is new in the Proposed Project?

we are proposing a novel approach to detect malware android applications based on their behaviour in run-time.

5 Proof of concept

Our experiments are conducted on **Malgenome** dataset, we implemented two different machine learning models, KNN and Naive Bayes Classifier in Python language. Dataset is split into two subsets, training set and testing set respectively. First training set is fed to the models to adjust their parameters, then the pre-trained model is tested using testing set achieving accuracy listed in the following table.

KNN			Naive Bayes
K = 3	K = 6	K = 9	
0.99	0.98	0.98	0.99

The figure below depicts the common permission in the malicious plotted in red and the benign plotted in blue.

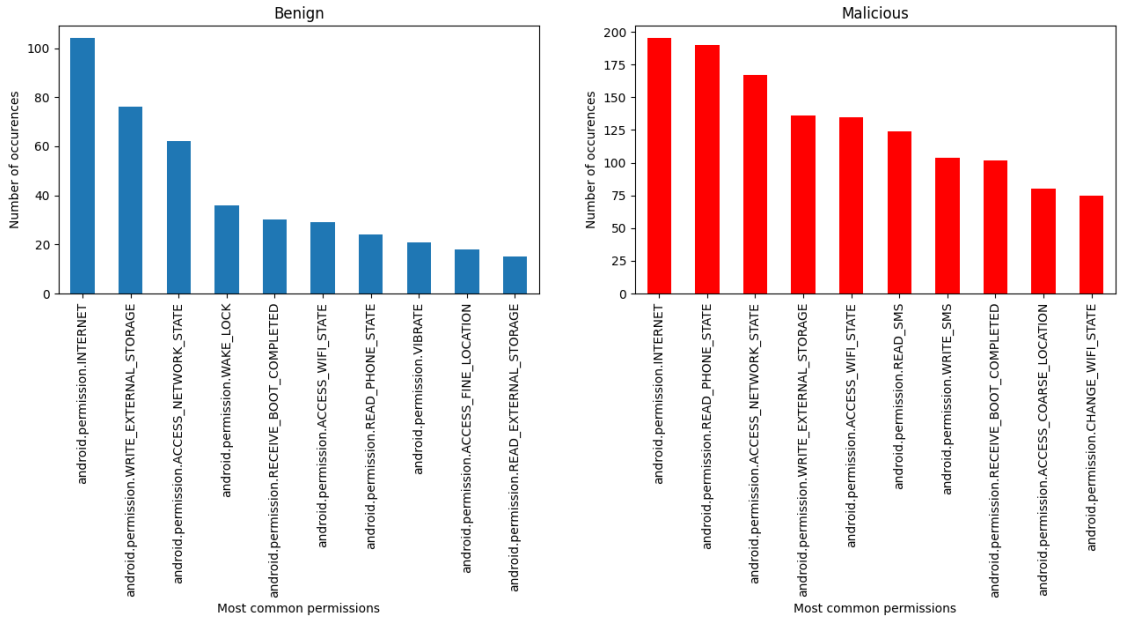

Figure 4: Permission analysis

6 Project Management and Deliverables

6.1 Deliverables

- Android framework that detect malware applications.
- Software Requirement Specification document.
- Software Design document.
- Thesis document and conference papers.

6.2 Tasks and Time Plan

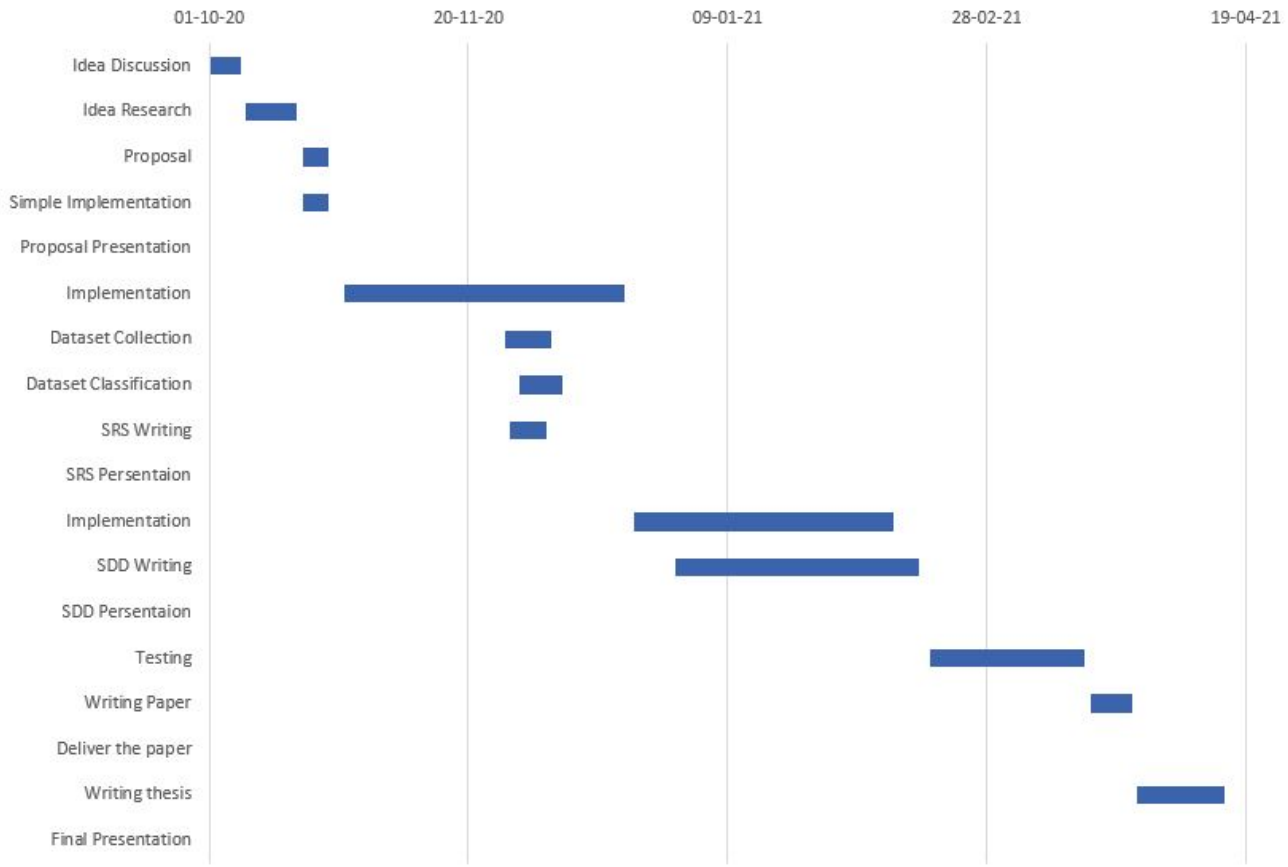

Figure 5: Project Timeline

6.3 Budget and Resource Costs

Using an Android smartphone.

7 Supportive Documents

We collect our dataset from a related work and gathered a large dataset to train and test our project on [16]. And another dataset which contain 300 column and row, column contains permission of api's ,and rows have zero's and one's in which 0 indicate to malware and 1 for the benign one.

References

[1] Dimitropoulou. https://ceoworld.biz/2019/01/18/worlds-most-popular-mobile-operating-systems-android-vs-ios-market-share-2012-2018/, 2019, January 18.

[2] Oscar Somarriba, Urko Zurutuza, Roberto Uribeetxeberria, Laurent Delosières, and Simin Nadjm-Tehrani. Detection and visualization of android malware behavior. *Journal of Electrical and Computer Engineering*, 2016, 2016.

[3] Xiang Li, Jianyi Liu, Yanyu Huo, Ru Zhang, and Yuangang Yao. An android malware detection method based on androidmanifest file. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 239–243. IEEE, 2016.

[4] Yuan Zhang, Min Yang, Bingquan Xu, Zhemin Yang, Guofei Gu, Peng Ning, X Sean Wang, and Binyu Zang. Vetting undesirable behaviors in android apps with permission use analysis. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 611–622, 2013.

[5] Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE symposium on security and privacy*, pages 95–109. IEEE, 2012.

[6] Md Faiz Iqbal Faiz and Md Anwar Hussain. Hybrid classification model to detect android application-collusion. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 492–495. IEEE, 2020.

[7] Parnika Bhat, Kamlesh Dutta, and Sukhbir Singh. Mapldroid: Malicious android application detection based on naive bayes using multiple. In *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pages 49–54. IEEE, 2019.

[8] Umme Sumaya Jannat, Syed Md Hasnayeen, Mirza Kamrul Bashar Shuhan, and Md Sadek Ferdous. Analysis and detection of malware in android applications using machine learning. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–7. IEEE, 2019.

[9] Zhuo Ma, Haoran Ge, Yang Liu, Meng Zhao, and Jianfeng Ma. A combination method for android malware detection based on control flow graphs and machine learning algorithms. *IEEE access*, 7:21235–21245, 2019.

[10] Muhammad Murtaz, Hassan Azwar, Syed Baqir Ali, and Saad Rehman. A framework for android malware detection and classification. In *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–5. IEEE, 2018.

[11] Ridho Alif Utama, Parman Sukarno, and Erwid Musthofa Jadied. Analysis and classification of danger level in android applications using naive bayes algorithm. In *2018 6th International Conference on Information and Communication Technology (ICoICT)*, pages 281–285. IEEE, 2018.

[12] Hongbing Yan, Yan Xiong, Wenchao Huang, Jianmeng Huang, and Zhaoyi Meng. Automatically detecting malicious sensitive data usage in android applications. In *2018 4th International Conference on Big Data Computing and Communications (BIGCOM)*, pages 102–107. IEEE, 2018.

[13] Bahman Rashidi, Carol Fung, and Elisa Bertino. Android malicious application detection using support vector machine and active learning. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE, 2017.

[14] Yuxia Sun, Yunlong Xie, Zhi Qiu, Yuchang Pan, Jian Weng, and Song Guo. Detecting android malware based on extreme learning machine. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 47–53. IEEE, 2017.

[15] Nathaniel Lageman, Mark Lindsey, and William Glodek. Detecting malicious android applications from runtime behavior. In *MILCOM 2015-2015 IEEE Military Communications Conference*, pages 324–329. IEEE, 2015.

[16] ElMouatez Billah Karbab, Mourad Debbabi, Abdelouahid Derhab, and Djedjiga Mouheb. Maldozer: Automatic framework for android malware detection using deep learning. *Digital Investigation*, 24:S48–S59, 2018.