# CHAPTER 1

# INTRODUCTION

## 1.1 DOMAIN INTRODUCTION

Android Operating System (OS) platform has turned into the quickest developing mobile OS dependent on its open source hence making it the most favored OS for many consumers and developers. The main advantages of Android OS to other mobile OS are as follow:

- it runs very powerful applications.
- it is very flexible and friendly as it allows users to make their choice of application.

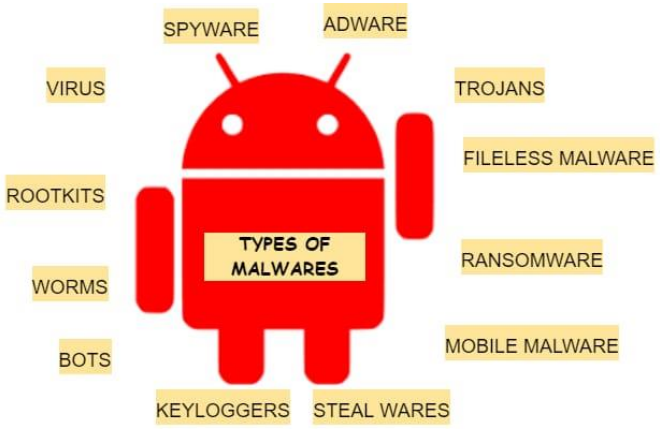


***FIGURE.1.1: TYPES OF MALWARES***

Android phones have been an attraction to several illegit mate operations because of its popularity and increasing openness. Hence, malware invading the android application is developing at a perilous rate and under the present circumstance, security of the devices and the resources these gadgets permit admittance to is very pivotal. The detection of Android malware to forestall Android devices against security breaks. Various approaches with diverse points and objectives had been broadly used in bringing out their strengths

and weaknesses. Malware is gotten from malicious software and it is often used to as software program that consciously possesses the deep attributes of malicious attackers and characterizes by its malicious aim. Different types of Malwares based on their diverse purposes and ways of penetration.

## 1.2 PROBLEM DEFINITION

The project is basically concerned about improving the accuracy of malware detection and the time needed to develop the model. The suggested approach attempts to use a Genetic Algorithm to obtain the most optimal function subset that can be used to inform machine studying algorithms in the most effective manner.

- The major objective to detect android malwares in a given dataset.
- To create a software which uses the given dataset to train & test the available algorithms and detect the Malware applications if any in the given new data.
- Comparison between the three algorithms i.e., SVM, Neural Network and Genetic Algorithm for more precise evaluation.

## 1.3 OBJECTIVES

Recently, mobile devices, particularly smartphones due to its capabilities in providing various services have been attractive. Most people use these devices for many purposes such as; online banking, web browsing, social networking, sharing data and etc. Since it is open source, it allows developers to arise their own technical advancements. As needs are, an enormous number of malignant applications might be gotten thoughtlessly. On the other hand, low information on the users in investigating the danger of these applications, make further adventures on data and protection.

These malicious applications can perform unapproved exercises for example, accessing user's delicate data, camera, SMS, calls, and so on. In this paper, a framework that assists the users with recognizing malignant or harmless introduced applications is proposed.

## 1.4 SCOPE OF THE PROJECT

- **Threat intelligence:** Strengthen security with up-to-the-minute insights.
- **Payload analysis**: Protect against zero-day attacks & complex threats with advanced malware detection engines.
- **Data loss prevention**: Block or monitor content uploads that contain sensitive data such as PII, PCI DSS, or HIPAA data.
- **Application control:** Identify and block unsanctioned applications based on risk score or limit application features.
- **Analysis and reporting**: See real-time insights into all outbound traffic, threat events, and more — on one dashboard.
- **Acceptable use policy:** Enforce your company policy. Make sure all employees comply by blocking certain domains.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 TECHNOLOGY

Technology plays important role in day-to-day life. Technology is defined as the innovative and effective methods to solve the problem using programming languages. Technology is set of programs to discover a new way to tackle the problem.

### 2.1.1 PYTHON

Python is one of the most commonly used programing language. Python works on various platforms such as Windows, Mac, Raspberry, pi, Linux and so on. It is a high-level programming language which has a wide scope. Python can be utilized to write a code In procedural way and object-oriented ways. Programs written in python has less number of lines, less syntax when compared to other programming languages like C, C+, JAVA. Programmers need not to type more lines. Python code is readable based on indentation. Python programming language uses indentation and white space to indicate the scope such as scope of iterative loops, methods and classes whereas other programming languages uses curly-brackets to indicate the scope.

### ADVANTAGES OF PYTHON

- **Extensible:** As we have observed, Python can be extended to work with different languages. A portion of your code can be developed in other languages like C++ or C. This is vital in projects.
- **Embeddable*:*** Python is additionally embeddable, which adds to its extensibility. Programmers can incorporate Python code in the source code of various programming languages like C++ or C. Python permits client to put functionality of scripting to other language's code.

- **Improved Productivity:** Python programmers can be more prolificity when compared to other programming languages due to Python programming language's simplicity and it has enormous library functions. Likewise, programmers need to write few lines and achieve more.

- **Simple and Easy:** Python is easy and simple to learn and code. In Java, programmers need to create a class to print a statement "This is Java code" whereas in python, print statement is simple and sufficient. Thus, people who learn python programming language think that it is hard to change to more verbose language like Java.

- **Readable:** Perusing Python is like perusing English because python isn't as verbose. For this reason, python is so easy to learn, understand, and code. Programmers do not require Curly-brackets to define scope whereas indentation is required.

### 2.1.2 MACHINE LEARNING

Arthur Samuel is trailblazer in the field of artificial intelligence and in the field of personal computer gaming. Machine Learning provides the capability to machines or models to learn from itself by training with the help of input, output and experiences, without need of programming.

In simple words, Machine learning can also be described as robotizing and further developing the learning system of personal computers grounded on their incidents without the need of programming specifically, which means without need of any human aid. The process of Machine learning begins with providing high standard Data with best quality and then training our machine i.e., computers, by developing ml models by applying various types of ml algorithms. Based on what kind of data we are selecting and what kind of task we need to perform, the selection of algorithms takes place.

**A SMALL EXAMPLE OF ML IS: "EXAM PREPARATION OF STUDENTS."**

In this example, machine refers to student's brain. High quality data refers to questions and answers from various textbooks or instructors' notes or various online course documents.

When examination approaches, students begin to prepare for their exams. They don't really pack the subject however They attempt to learn it with complete comprehension. During exam preparations, Students feed their machine with a high-quality data. Thus, they are training their machine using input along with the output in such a way that what sort of methods and approaches they need to apply to tackle various kinds of questions given in exam.

Every time, students tackle various practice model question paper and track down their performance (exactness/score) by contrasting their solutions with existing answer bank. Gradually, the performance continues to increment, and acquiring more confidence with the adopted approach. In this way models are assembled, train machine with information (where both the inputs and outputs bare provided to model) and then they test on data by providing only input and accomplishes our model performance by contrasting its answer with the actual output which has not been provided during training. Scientists are working with diligent endeavors to further develop the strategies and approaches by which these models perform even more better.

**ADVANTAGES OF MACHINE LEARNING**

- **Automation:** One of the advantages of Machine learning is that it reduces the work pressure and reduces time consumption to tackle the problem. Computerization is currently being used in many places because of its reliability and also assists us to think more intensively.
- **Wide Scope of uses:** ML has a wide assortment of utilizations. Ml has broad scope of application where we can apply on any significant fields. From Medical related, business related, Bank related to science and technology, Ml has its own role. Ml assists to develop more open doors i.e, more opportunities. Organizations create benefits, reduces expenses, computerize, foresee the future, investigate on

patterns and fashion from the past information, and many Other roles. Few apps like Global Positioning System Tracking for traffic, filtering the email spam messages, handwriting recognition, speech recognition, text recognition, auto corrections in documents, and so forth are utilized broadly nowadays.

- **Continuous Improvement:** Algorithms of Machine learnings are fit for gaining the information we give. As new information is given, the model's precision and effectiveness will enhance by training the model again and again. Goliaths like Amazon, Walmart, and so forth gather a gigantic volume of new information consistently. The precision of searching related products or proposal engine enhances with this immense measure of training data.

## 2.2 EXISTING SYSTEM

In modern computerized world, android application plays an imperative role. On the opposite side even, malware has turned to an unending threat to digital environment. There are many investigates and hypotheses to address the issues brought about by malware and to detect the android malware various tasks have been carried out by researchers

In existing system, researchers have used Signature based, filtering and permission-based approaches in static analysis. Honey pot, system call logs, anomaly behavior monitoring approaches in Dynamic.

**DISADVANTAGES**

- Cannot detect root exploits and scripts malware
- Can't accurately recognize the root exploits and malwares.
- Execution is low and energy is overhead
- If the dimension of features are high, then it results in incrementation of the time overhead of huge data analysis.
- Utilizing the whole program run as epoch is not practical and feasible for constantly running applications like web browsers.

## 2.2.1 BASE PAPER

Android platform because of open-source trademark also, Google backing has the biggest worldwide portion of the overall industry. Being the world's most well-known working framework, it has drawn the attractions of digital hackers working especially through the wide circulation of malevolent applications. This paper proposes an effective ML-based methodology for Android Malware Detection utilizing evolutionary Genetic algorithms for biased feature selection. Chosen features from Genetic algorithms are utilized to prepare ML classifiers and their ability in the identification of Malware. The trial-and-error outcomes approve that genetic algorithm give the most improved element subset helping in a decrease of component aspect to not exactly 50% of the first include set. Classification precision of over 94% is kept up with post feature selection for the ML-based classifiers, while dealing with much-diminished element aspects, along these lines, decidedly affecting computational complexity of learning classifiers. Android Apps are openly accessible on Google Play store, the official Android application store just as outsider application stores for users to download. Because of its open-source nature and ubiquity, malware writers progressively center around creating malicious applications for the Android working framework. Regardless of different endeavors by Google Play store to secure against malevolent applications, they actually track down their method for massing market and cause damage to users by misusing the individual's data connected with their telephone directory, mail accounts, GPS area data and others for misuse by outsiders or else assume responsibility for the telephones from a distance. Consequently, there is a need to perform malware examination or picking apart of such malignant applications which present a genuine danger to Android stages.

Since the revelation that ML can be utilized to viably distinguish Android malware and good wares, many types of research on ML-based malware detections strategies have been conducted. A few techniques dependent on features determination, especially genetic algorithms, have been proposed to expand execution and diminish costs. Nonetheless, in light of the fact that they still can't seem to be contrasted and different techniques and their many features have not been adequately confirmed, such strategies

have specific constraints. This review explores whether genetic algorithm-based feature selection helps Android malware identification. Jaehyeong Lee and et al applied nine ML algorithms with genetic algorithm-based feature selection for 1104 static elements through 5000 benign applications and 2500 malware included for the Andro-AutoPsy dataset.

Near test results show that the genetic algorithm performed better compared to the data gain-based technique, which is by and large utilized as a feature selection strategy. Also, ML utilizing the proposed genetic algorithm-based feature selection enjoys an outright benefit as far as time contrasted with ML without feature selection. The outcomes demonstrate that fusing genetic algorithms into Android malware locations is a significant methodology. Moreover, to improve malware recognition execution, it is valuable to apply genetic algorithm-based feature selection to machine learning.

In general, Malware investigation is of two types: Static Analysis and Dynamic Analysis. Static investigation fundamentally includes investigating the code structure without executing it while the dynamic investigation is an assessment of the runtime conduct of Android Apps in the compelled environment. Given in to the always-expanding variations of Android Malware presents zero-day dangers, a productive instrument for the detection of Android malware is required. In differentiation to a signature-based methodology which requires a normal update of mark data set, ML-based the approach in the mix with the static and dynamic analysis can be utilized to recognize new variations of Android Malware presenting zero-day dangers.

ML is defined as inherently a multidisciplinary field utilized in different genuine applications. ML is led by acquainting a few algorithms to tackle issues in different fields. This is the motivation behind why ML should be assessed through different algorithms. The following depictions are a few ML algorithms utilized:

**Decision Tree**: A Decision tree is an ML procedure that groups or relapses the information by making arrangement rules for the trees. A Decision tree is directed by a preparation case in which data is addressed by a tuple and the class mark of the characteristic qualities is recorded. Attributable to the tremendous space to be recovered, a tree is normally

directed via preparing information and void trees and into covetous, hierarchical, and recursive cycles. The tree is made involving processes that best segment the preparation information as the root parting characteristic. The preparation information is then divided into disjoint subsets that fulfil the upsides of the parting characteristic. Attributable to the inconvenience of effectively overfitting the trees, pruning might be applied while executing the calculation, or a few trees may be taken out to frame summed up outcomes.

**Random Forest**: A Random Forest is a classifier comprising of an assortment of uncorrelated tree structure classifiers planned by L. Bierman. Each tree has the trait of tracking down arrangements while deciding in favor of the most summed up class for input qualities. A regulated learning calculation prepared to utilize the sacking strategy assembles different choice trees and unions them to acquire a more exact and stable expectation.

**Decision Table**: A Decision table is a straightforward method for archiving various choices or activities taken under various arrangements of conditions. The choice table permits the production of a classifier, which sums up the dataset into a choice table that contains a similar number of qualities as the first dataset and applies the arrangement of new approaching information utilizing the table.

**Naive Bayes**: Naïve Bayes is a grouping model dependent on the contingent likelihood of a Bayes rule. The autonomous guileless Bayes model depends on assessing and looking at probabilities; the bigger critical likelihood brings up that the real mark is bound to be the class name worth of the bigger likelihood. As the calculation expects that the prescient characteristics are restrictively autonomous given the class, and it sets that no covered-up or dormant traits impact the forecast interaction, it is hard to apply to information reliant upon various classes through explicit properties.

**MLP**: A multi-layer perceptron (MLP) involves omnidirectional counterfeit neural organizations for learning. The neural organization structure is introduced in three sections: the info layer stowed away layer, and the result layer. The information layer gets the information, the secret layer is determined through an enactment work, and the result layer shows the consequences of grouping/relapse. Although the information/yield layers

of the model exist separately, the secret layer can be stacked as different layers. It is additionally realized that the more deeply a model is, the more summed it up can be in contrast with a shallowly stacked model.

**SVM**: A Support vector machine (SVM), otherwise called a support vector network (SVN), is a learning model for double grouping that encapsulates the possibility that input vectors map non-straight to high-layered components spaces. In this element space, a straight choice surface is developed to guarantee the high speculation capacity of the learning machine attributable to the extraordinary properties of the decision surface.

**Logistic Regression**: Logistic Regression is an ML method that clarifies how factors with at least two classes are related with a bunch of ceaseless or unmitigated indicators through likelihood capacities. Dissimilar to customary direct relapse involving straight lines for arrangement, calculated relapse is reasonable for parallel grouping, involving a strategic capacity looking like ex(1+ex) when fitting the information.

**AdaBoost**: AdaBoost is a helping calculation that joins different powerless classifiers to make a vigorous classifier that expands the exhibition. Not at all like recently proposed supporting calculations, a powerless classifier is described by mistakes returned by the feeble classifier, which influences the focal point of the frail classifier on the hazardous instances of the preparation set, permitting it to more readily arrange the characteristics.

**K-NN**: As the key guideline of a K-closest neighbor (K-NN), if a large portion of the examples around the information on a point in a specific space has a place with a particular classification, the information on that point can be decided to fall into that class. The K-NN calculation works by checking the name of the suitable quantities of tests nearest to the information being ordered and along these lines marking the information as the most amassed of the samples.

**Genetic Algorithm**: Assign a genetic algorithm utilized features subsets which are paired encoded end goal that whether or not the feature is incorporated. If the feature is incorporated then it is addressed by one (1) and if the feature is not included then it is addressed by zero (0) in the chromosome. Set about the given algorithm by. Genetic algorithm is initialized by a wellness score or rate determined the characterized of the

fitness function. Parents Selection: Chromosomes have the great fitness scores or rate that are given by inclination, whereas the others are to deliver the upcoming generation of the off-springs. Transformation procedure and Perform hybrid are picked.

Detecting Android malware in a speedy and precise way is fundamental for Android OS clients. To take care of this issue, many investigations have presented ML for the identification of malicious applications, and feature selection has additionally been utilized to accelerate the cycle. Tests were directed to choose authorization and API technique data elements to apply ML dependent on existing exploration, and the outcomes show that genetic algorithm-based feature selection was additionally valuable contrasted with a generally applied data gain. A definitive objective of ML is to have the option to supply existing spending plans to ML frameworks notwithstanding precision prerequisites, with the framework observing a working point that permits such necessities to be acknowledged. Considering this, the trial results show that involving genetic algorithms for Android malware detections is additionally valuable contrasted with different methodologies. The examination additionally shows that it very well might be useful to continue with highlight choice utilizing a genetic algorithm. It is important to direct an approval utilizing a dataset comprising of as of late delivered applications. Assuming further exploration results are accessible later on, we will note genetic algorithms as an answer for malware identifications to accomplish higher exactness with lesser time costs. Further work can be upgraded utilizing bigger datasets for further developed outcomes and dissecting the impact on other ML algorithms when utilized related to the Genetic Algorithm.

### 2.2.2 RELATED REFERECE PAPERS

Mohd Faizal Ab Razak et al has been increasing use of smart phones and tablets have caused cybercrimes to change their attack tactics to mobile devices. Development of Android has drawn cybercriminals to create malicious applications that take sensitive data that affects mobile systems. A multi-criteria decision-making based (MCDM) mobile malware detection system utilizing a danger based fuzzy analytical hierarchy process

(AHP) approach. Permission-based features. The main advantage is the Risk analysis is applied to increase the awareness of the mobile user in granting any permission request.

Fan Ou, Jian Xu has proposed a hundred of features are available for machine learning-based malware detectors. Thus, the main key role of the Android security community is to continuously propose a new feature that can characterize malicious behaviors. It is a static sensitive sub graph-based feature. Examination of various methodologies; - S3Feature versus hybrid features S3 Feature is easy to be integrated with other features because of its vector-based representation.

Architecture of Recurrent Neural Network (RNN) can perform the detection process better than conventional machine learning algorithms which was proposed by Mothanna Almahmouda, is an official market of Android Google Play store which is characterized by its support for the informal stores, and it doesn't impose any restrictions on developers during the dispersing process. These features were a major reason for making it becomes the most vulnerable platform to cyber criminals, as users are experiencing the issues of exposure to malicious applications that breach their privacy or harm their devices.

Mohd Faizal Ab Razak has proposed the unprecedented growth of a mobile technology which has been generated by an increase of malware and raised concerns over malware threats. Different methodologies have been embraced to overcome the malware attacks yet this spread is still increasing. An Android malware detection system is dependent on permission features.

Qing-Fei Wang, et al has proposed an Android system is widely favored by users because of its open platform, rich software content and services. Simultaneously, Android malware is also emerging for the purpose of obtaining improper benefits, which brings serious security problems to the Android stages. The main advantage is it combines static analysis with dynamic analysis can improve detection accuracy and efficiency at the same time

Zhixing Xu Sayak Ray Past proposition for malware detection has been basically focused on a software-based detector which is defenseless against malware exposed. Malware and Memory Access Patterns attainability of our methodology with tests both kernel-level

and user-level malware. This provides for increasing in an automation and coverage through decreasing the user input on a specific malware signature.

## 2.3 PROPOSED SYSTEM

One of the most important concerns that need to be addressed is cyber criminals posing the threat to digital environment. In this paper, we provide a scene which detects the malware in android application.

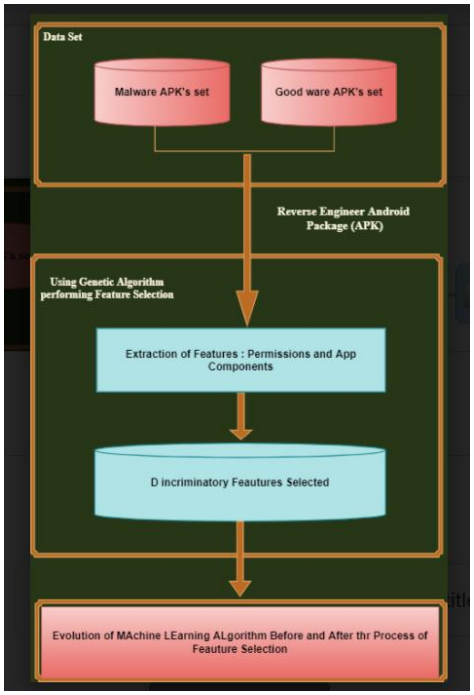


***FIGURE 2.1: PROPOSED FRAMEWORK***

- The model is trained using feature dataset. Dataset consists of malware APKs set and good ware APKs set.
- These databases will undergo reverse engineered process to extract all the features such as Permissions and app components- Activity, Services, Intents etc.
- The model is developed using python. Genetic algorithm is used to choose the important features from the dataset to train the model.
- Later the application is detected weather it is malware or good ware based on feature data set.

**ADVANTAGES:**

- Machine learning helps to detect the anti-virus software's and anti-virus apps which imposes threats to day-to-day life and user's data without relying on signatures.
- Using Genetic Algorithm in this project supports multi-objective programming and it perform search from overall population of points rather than a single point.
- Reverse engineering is used which reduces the expenses and provides effectiveness in detecting malware function.

# CHAPTER 3

## ANALYSIS OF REVIEWED PAPERS

In Today's world, cell phones are telephones as well as go about as convenient PCs that offer assorted types of assistance. Nonetheless, cell phones are running Android are additionally focused on by malware identification. The motivation behind the review proposed by SONGYANG WN and YONG HANG is to decide the method for relating Android malware and their malevolent touchy information transmissions more successfully than the current strategies. As of late, numerous methods and plans have been favorable to presented to determine the developing issue of Android malware. One more examination heading of related investigations is to recognize conceivable malevolent practices by applying AI. Songyang Wn, Yong Zhang [1] has utilized three standard measurements that are utilized to assess the exhibition of the created grouping models. The grouping models are classified as true positive, false positive rate and accuracy. Algorithm used by authors is: K-Nearest Neighbors (KNN), Logistic Regression (LR). The activity of dataflow-related APIs is an admirable sentiment for relating Android malware. The proposed plot gives a successful way to deal with identifying Android malware and researching security infringement in vindictive applications.

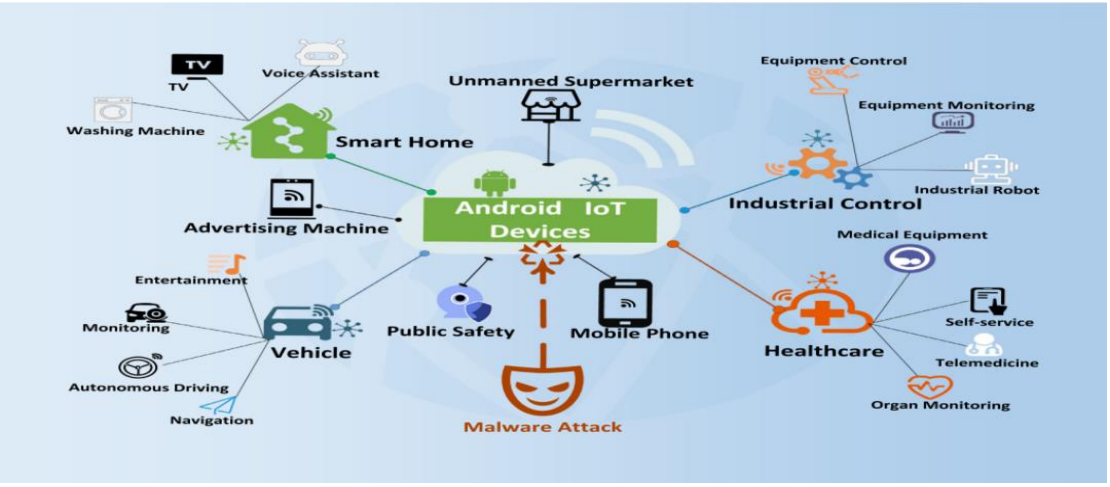


***FIGURE 3.1: RAPID SPREAD OF MALWARES***

DARGOS et al proposed a mutable system where one can utilize distinctive AI calculations to effectively recognize malware lines and clean lines. MIHAI CIMPOES and et al presents the idea behind the architecture by working initially with course uneven perceptron and also with course kernelized uneven perceptron. In the course of been efficiently tried on datasets which are medium in size of malware and error free documents, the idea behind this system were collected to an increasing interaction that empower to work with genuine enormous datasets of malware and error free records. A basic part of malware comprises of record virus and independent malware. Malware location through norm, hand grounded styles is getting further increasingly more sensitive since all current malware activities will quite often have numerous polymorphic layers to keep away from recognition. In this paper GAVRILUT [5] and others present a system for malware recognition intending to get as scarcely any bogus cons as could be expected, by utilizing a straightforward and a basic multi-stage mix (course) of various exhibitions of the perceptron calculation. Few algorithms used by MIHAI CIMPOES and 2 others are Perceptron algorithm, One-sided perceptron, Simple feature generation algorithm, kernelized one-sided perceptron Algorithm and One-sided Perceptron Algorithm. Their principal target was thought of an AI structure that conventionally recognizes malware tests, with the extreme limitation of having a zero bogus positive rate. Malware identification by means of machine education won't supplant the standard discovery styles utilized by hostile to infection merchandisers, yet will come as an expansion to them.

Android malware recognition has drawn in significant consideration as of late. Being styles significantly test on extricating static or dynamic highlights from versatile applications and make portable malware discovery model by AI calculations. The quantity of evacuated static or dynamic elements maybe much high. Therefore, the information moves from high dimensionality. What's more, to try not to be recognized, malware information is differed and difficult to acquire in any case. ZHEN LIU and JIE ZHAO [3], proposed a solo point learning calculation called Subspace grounded Restricted Boltzmann Machines (SRBM) for lessening information dimensionality in malware

identification. Various subspaces in the first information are initially looked. And furthermore, a RBM is raised on every subspace. All results of the resigned layers of the prepared RBMs are consolidated to address the information in lower aspect. Versatile malware advancement is malignant programming that is explicitly intended to target cell phones, like PDAs and tablets. An assortment of AI based ways has been dove to descry Android malware. In this paper, we significantly complete the plan from the accompanying angles: examination on the presentation on account of zero-day malware location, examination on the presentation assessed by bunching assessment rules, examination on the presentation assessed by characterization assessment rules, conversation on the boundaries of SRBM and conversation on the time utilization of element decrease strategies. Author has used Stacked Auto Encoder (SAE), Principal Components Analysis (PCA), K-step contrastive divergence. This paper proposes a solo element learning technique for versatile malware discovery. It depends on RBM and it very well may be utilized for diminishing the information dimensionality on account of solo portable malware recognition. To drop asset utilization and to improve the presentation of RBM, it advances RBM by presenting the subspaces idea.

Large scale malware in Microsoft Office lines has since quite a while ago endured as a network protection inconvenience. However, it retrograded after its unique frenzies when the new century rolled over, it experiences reappeared as difficulty. Bushwhackers are adopting a convincing strategy and utilizing record designing, upheld by bettered information mining styles, to make MS Office document malware seem real. Ongoing assaults have designated explicit pots with noxious records containing shockingly material data. This advancement sabotages the capacity of clients to recognize vindictive and real MS Office lines and strengthens the requirement for robotizing full scale malware discovery. RUTH BEARDEN and DAN CHAI TIEN LO [4] used k-nearest algorithm. This review proposes a procedure for grouping MS Office records containing macros as awful or harmless utilizing 3 the K-Nearest Neighbors AI calculation. Malware identification with AI is an arising answer for programmed malware discovery in a wide range of noxious records; however more work has been done in its application to executable documents.

In the identification of noxious non-executables. The excellence of the technique for mechanized large scale malware location this review proposed is its dependence on data recovered from records without opening them. The magnificence of the technique for mechanized full scale malware identification this review proposed is its dependence on data recovered from documents without opening them. This would empower assortment of a precise and huge example sets on which to prepare AI classifiers.

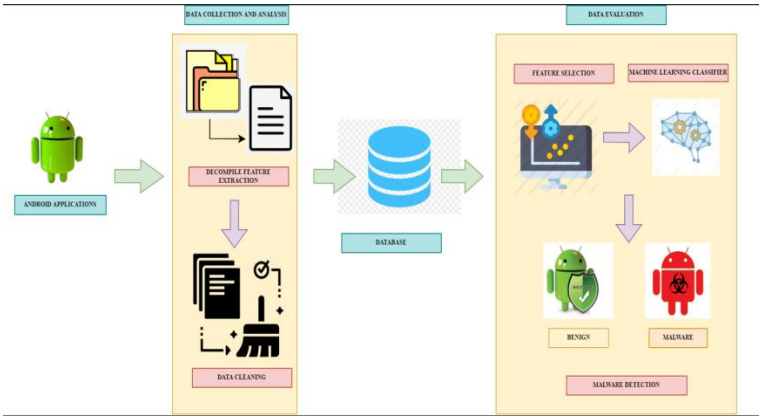


***FIGURE 3.2: MALWARE DETECTION SYSTEM FRAMEWORK***

Android working framework to upgrade its heartiness, sheer assurance stays an open issue: awful activities by and large observe ways of bypassing the security processes, while clients are not worried of an earlier whether an activity can work as malware. To bar this issue, a few methodologies impact AI for recognizing malware utilizing static examination information. In this cycle, we can concentrate on the viability of managed AI calculations utilizing static investigation informational index. Right now, cell phones are broadly utilized for arranging various undertakings comparable as installments, booking and purchasing tickets, taking prints and so on utilizing the relating versatile application. Toward this path as of late, various methodologies use AI (ML) and static examination to isolate malevolent from great product applications in the Android OS. In the long run we close this work with reflections and pointers for future work. Authors used Support Vector Machine (SVM), Naive Bayes (NB), Chi-square Test, K Nearest Neighbor (K-NN), Decision Tree (DT). VASILEIOS SYRRIS and DIMITRIS GENEIATAKIS [6], completely examine the viability of notable ML classifiers for distinguishing vindictive applications in Android OS, we examine elective ways of measuring the significance of the first elements and we report about other material works

and approaches. Along these lines, we give a top to bottom check in the space of ML-based malware discovery in Android OS.

Shrewd portable predisposition is progressively well known and normal in ultramodern life. NGOC C. LE and TRANG TRUONG [7], proposed an AI strategy to recognize malevolent procedure on Android gadgets. The elements utilized for AI are grounded on the normal gets of malignant applications, the required warrants, and different elements taken from the applications documents. There by making a system that can run continuous on ordinary inclination like a typical PC. Lately, the utilization of AI or profound learning techniques has become famous, applied in various fields, and accomplished positive output. AI strategies are appropriate to the qualities gotten from dynamic and static investigation. The pernicious code identification issue on Android is additionally a model. The utilization of AI or profound learning strategies is relied upon to make malignant indicators that can distinguish new malwares all the more precisely without earlier examination of the noxious code. These methodologies for Android malware location prompted another exploration pattern with many great outcomes. Different extraction techniques explicit to the AI preparing process have been proposed. Author used Decision Tree, Naive Bayes, Random Forest, Boosting methos. The result of our model is exceptionally sure. Simultaneously, we can exhibit the included extraction speed from APK document with our extractor. It is feasible to assemble a continuous android malware analyzer and can likewise be utilized as a local library to construct an AV application on Android gadgets.

The open-source Android stage permits engineers to exploit the portable activity framework, yet in addition raises huge issues identified with noxious applications. The expanded quantities of uses, then again, ready a reasonable inclined for certain clients to foster various types of malwares. Creator has proposed to join application consent and API calls and use AI calculations to distinguish malignant applications. In this plan, the consent is extricated from every Application s profile subtlety and the APIs are separated from the stuffed Application record. By utilizing authorizations and API calls as highlights to describe every Application dependent on noxious level. Author has proposed an element-based learning structure for Android malware ID. The structure incorporates the application elements, consents and API calls to distinguish Android applications practices. Every Application addressed as a solitary example with paired consent and API call highlights, and a class name demonstrates whether the Application is an easy to understand or a malware. The system has changed over Application into nonexclusive case highlight design; one can basically utilize any learning calculation to get grouping models from the

preparation information. NASER PEIRAVIAN and XINGQUAN ZHU [8] used Support Vector Machines, Decision tree, bagging as their base algorithm to identify the malwares. Author utilizes consents and API calls of utilizations to distinguish malware and vindictive codes in portable stage. The commitment of the structure, contrasted with the current arrangements, is tri overlap: Combining consents and API calls is successful for malware discovery; Framework doesn't include dynamical following of the applications; Framework can be summed up to all applications for malware recognizable proof.

In today's technological world android application plays a vital role. On the other side even, malware has become an endless threat to digital ecosystem. There are many researches and theories to solve the issues caused by malware and it is been found that machine learning is one of the effective and efficient way to deal with malware detection. In this paper author has introduced few backgrounds of android application such as android system architecture, security mechanism, and lastly classification of malwares. GUOAI XU [9] has included static dynamic, hybrid detection techniques for sorting the threats faced by android devices due to malwares. Static detection is based on review of suspect code without running it in the android system in contrast dynamic detection is done while running the code in the application. Hybrid 4 is the combination of both static and dynamic detection to get a balance among detection effectiveness and efficiency. GUOAI XU used supervised, unsupervised, semi-supervised, reinforcement learning algorithms. The involvement of artificial intelligence methods, such as machine learning, has significantly improved the expectation level for the identification of android malware. In this article author has provided a detailed review of present approaches for detecting Android malware using machine learning.

In this paper, MINGHUI CAI et al momentarily presents conduct level highlights of an application from work calls. The difficulties of this issue are double overlay. In the first place, the shortfall of capacity ascribes mishaps the comprehension of use practices. Second, the graphical portrayal of capacity calls can't be straightforwardly handled by AI calculations. Android is the significant prey for versatile malware. Till now, an assortment of AI based ID techniques have been to create to determine the Android malware issues. In light of the removed elements codes, these capacities utilize a separator to choose whether an application is pernicious. Practically speaking, the presentation of malware location relies intensely upon the highlights. Significant highlights of Android malware discovery fall under two segments: dynamic and static. Dynamic elements mirror the runtime practices of utilization, henceforth gives significant clues to malware ID.

Dynamic elements require perception of execution of utilizations, which causes overheads and difficulties. Interestingly, static highlights don't need the execution of utilization. In static elements of malware identification authorization required, goal activities and capacity calls are utilized. Be that as it may, static highlights can't unequivocally address the runtime practices of use. YUAN JIANG, CUIYING GAO [10] and two others used Function embedding, Graph Convolution Networks for malware detection. They also used LR, DT, SVM, KNN, RF, MLP, CNN classifiers. Reviewing the runtime practices of Android applications assumes indispensable part in malware recognition. Noticing the execution of use is expensive, thusly taking in conduct level component from the data of capacity calls helps in location of malware. Creator has utilized new idea of E-FCGs to definitively portray the runtime practices of use. Creator has fostered a GCN based calculation to execute conduct level elements from E-FCGs. Tests has shown highlights which outperforms the downsides of customary static elements.

Android application because of open-source highlight has the biggest worldwide portion of the overall industry. Being the universes most renowned working framework, it has drawn the consideration and interest of digital cheats working especially through wide conveyance of malevolent applications and malware. ANAM FATIMA [11] and others proposes a successful AI based calculations for Android Malware ID utilizing Genetic calculation for oppressive component determination. Recognized elements from Genetic calculation are utilized to prepare AI classifiers and their ability in discovery of Malware prior and then afterward include choice is looked at. Two arrangements of Android Application: Malware/Good product are utilized to remove highlights, for example, consents and count of Application Components like Activity, Services, Content Providers, and so on These highlights are utilized as component vector with class marks addressed by 0 and 1 separately in CSV design. The upgraded set of highlights got is utilized for preparing double AI classifiers: Support Vector Machine and Neural Network. At long last, the pick highlights are taken care of as contribution to AI calculations for assessment reason. RITESH MAURYA and other authors Support Vector Machine, Neural Network, Genetic Algorithm. Here is a pinnacle expansion in dangers to computerized frameworks primarily through malevolent applications or malwares, in this way plan a system which can recognize such malwares with exact outcomes. Where AI based calculations are being utilized. The proposed procedure utilizes Genetic Algorithm to get most enhanced element subset which can be utilized to prepare AI calculations in most proficient manner.

AI has the ability to distinguish assortment of malware records that can sidestep signature-based ID. The utilization of Feature hashing is to change highlights into a fixed-length vector. Vector size is utilized for include hashing for a tremendous datasets of malware documents. Malware records have a few highlights that can be utilized for preparing models of AI. Static highlights can be taken out from malware records without running the documents. Scarcely any static elements are - File header data, byte n-gram, strings, and dismantled codes. Dynamic highlights are removed while the documents are executing. Creator centers around static highlights produced from compact executable (PE) documents that make up the significant piece of malware records. DAMIN MOON, JAEKOO LEE, MYUNGKEUN YOON [12] center around regulated learning alongside a marked datasets of easy to understand and malware documents, like irregular woods, support vector machines, profound learning calculations. After highlights are removed from both easy to understand and malware documents, it is addressed as a fixed-length vector that is placed into AI calculations as an information vector. Not many elements are easy to change into fixed length vectors, for example, record size and the quantity of areas and addressed as a vector involving a proper number of components. Authors utilize vector size of element hashing when preparing and testing AI models to distinguish malware records. Default vector size of the cutting edge conspire is excessively enormous, and thusly the space isn't productively utilized. Through datasets of genuine malware and easy to use documents and vector size can successfully save memory space.

Android gadgets are powerless against cybercrimes utilizing malware applications. AI calculation focuses on security. ALI AL ZAABI, DJEDJIGA MOUHEB [13] proposed AI for android malware distinguishing proof where the fundamental center is to utilize static elements of Android Application Package. Highlights like authorizations, API calls, administrations, opcodes, and exercises to prepare distinctive AI models to isolate an APK document as malware or easy to understand. In Android Malware Identification directed learning is utilized to check an application as either vindictive or easy to use. AI models utilize highlights, which 5 are the quantifiable factors utilized in the model. Android highlights can be created by three capacities, static and dynamic examination and cross analysis. Static investigation is filtering the decompiled Android Package's documents, consents, source code, and API calls. Dynamic Analysis is executing the application in a disconnected climate where the System calls, peruse and compose activities, and network information is utilized. Crossover Analysis is the idea of utilizing both static and dynamic investigation in the malware distinguishing proof system however it’s exorbitant. Authors focus on static component-based AI approach like opcodes, authorizations, API calls, exercises, and

administrations for Malware Identification. They have wanted to add a second layer of examination utilizing dynamic elements produced from APK documents to additional eliminate out malware that has been delegated easy to use. They had investigated AI models like Deep Neural Networks, utilize better elements determination calculations to eliminate any excess or pointless elements for showing the model hence decreasing the time taken to prepare and test the models.

The increase in malware becomes a serious threat to digital world. Therefore, automated behavior-based malware identification using machine learning techniques is considered a perfect solution. One of the answers for tackling the issues of malware is by utilizing programmed dynamic malware examination joining along with information mining undertakings, similar to AI methods to accomplish adequacy and productivity in identifying malware. IVAN FIRDAUSI [14] et al Information Acquisition and Storage, Programmed Behavior Monitoring and Report Generation, Information Pre-handling, Learning and Classification. Author focuses on Feature selection utilizing Best First search methodology to recognize the malware. In couple of cases, the execution can likewise increment marginally. The exhibition examination of 5 unique classifiers was additionally detailed. The complete best execution was accomplished by J48 utilizing the term recurrence weight without highlights choice informational index. The review of the tests and results was viewed as productive in recognizing malware.

Android application has eased the path for the digital revolution. This is accompanied by steep growth in the number of crimes giving steady rise to variety of advanced malicious files. Old malware identification approaches have revolved around pattern based detection. Author presents a novel feature-engineering technique for android malware identification using Machine Learning. Android applications were generally inclined to malware contaminations. Old Static Analysis process will more often than not fall flat in the distinguishing proof of malwares. Because of an exceptionally minuscule distinction in the consents needed by easy to understand and vindictive records it's not plausible. AI arrangements assume crucial part investigated to take care of the issue of Android Malware ID. ARINDAAM ROYA [15] et al investigated the Android API calls to create includes and further total them to work out the aggregate recurrence of each element and make them in a solitary tuple per Android Package Kit record. They utilized Non-negative Matrix Factorization, an amazing AI strategy for diminishing the absolute number of elements to make model lightweight furthermore versatile. Author investigated API calls from the produced

small codes for the advancement of a recurrence-based component vector for application. The adequacy of list of capabilities is determined with the help of different AI classifiers.

Late rise of computerized stages prepared to do running progressively complex programming and the ascending of delicate applications, there is an ascent of caution related malware designated on android application. JUSTIN SAHS and LATIFUR KHAN [16] present an AI based framework for the ID of malware in android portable. The brilliant gadgets have in short order become a very predominant figuring stage. The issue of utilizing an AI based classifier to recognize malware needs to confront two primary difficulties: Firstly, given an application, some kind of element portrayal of the application ought to be created; Secondly, informational collection that is only easy to use ought to be available, so we should pick a classifier that can be prepared on just one class. To resolve the main issue, a heterogeneous list of capabilities is created and processes each element autonomously utilizing various bits. To resolve the subsequent issue, a One-Class Support Vector Machine is utilized, which we train utilizing just easy to use applications. They have tried their framework against an assortment of 2081 easy to understand and 91 vindictive Android applications. For each informational collection, they have chosen an irregular subset of the preparation applications and performed overlap cross approval. Notwithstanding the full part, they likewise prepared against every individual bit independently. At long last, they utilized the One-Class Support Vector Machine.

The Android working framework has become generally recognizable for cell phones and android gadgets. This notoriety has prompted a fire raise of Android malware. Android malware obscurity and recognizable proof aversion techniques have altogether evolved, making numerous old malware ID processes out of date. Deep learning (DL) has acquired expanding consideration in the AI segment and is reappearing as a popular strategy for AI being applied. DL classifiers have extraordinary number of powerful methodologies. As of late, Android malware hypotheses have likewise been investigating DL classifiers for malware examination to expand ID exactness. MOHAMMED K. ALZAYLAEEA [17] et al utilized DL-Droid, a profound learning framework to distinguish vindictive Android applications through powerful examination. The result features the significance of upgraded input age for dynamic examination as DL-Droid with the state-based info age is displayed to outflank the current best in class draws near. Utilizing DL-Droid, the exhibition of the stateful info age approach is examined. Author has introduced DL-Droid, a robotized dynamic investigation structure for Android malware recognizable proof. DL-Droid utilizes profound learning with a state-based information age approach as the default interaction. The

introduced results plainly infer that accomplished high precision execution arriving at preferable figures over those introduced in existing deep learning-based Android malware ID structures. Android portable is exceptionally difficult on the grounds that it is an open-source OS which is likewise powerless against assaults. Past investigations have shown various versatile malware discovery techniques to beat this issue, yet at the same time, we can improvement it more. Versatile clients generally overlook extensive arrangements of authorizations as these are oppressive to peruse and comprehend. Along these lines, to regard harmless or malware applications and the likelihood of every authorization demand is perceived, it is important to assess Android portable applications.

MOHR FAIZAL AB RAZAK [18] proposed a multi-model independent direction based (MCDM) android portable malware discovery framework utilizing hazard based fluffy logical chain of command process (AHP) way to deal with assess the Android versatile applications. This examination focuses on static investigation which utilizes authorization-based highlights to assess the versatile malware identification framework approach. In these 6 papers, Risk investigation is applied to build the familiarity with the portable client in allowing any consent solicitation to contain a high hazard level. In this review, a versatile malware discovery framework is created dependent on hazard assessment utilizing the fluffy AHP strategy. Consent based elements were chosen to measure the portable malware discovery framework execution. Thus, the pair-wise examination featured those properties authorization gatherings gotten the most noteworthy weightage. Later on, Comparative concentrate between fluffy AHP and other MCDM approaches can be led to approve the huge techniques to further develop portable malware discovery frameworks.

As the most well-known portable stage, Android has become the significant objective of malware, and hence there is a crisis need to viably upset them. Of late, the AI based procedure has been a key for malware recognition, which exceptionally relies upon recognizing elements to isolate the malware applications from the harmless applications. Despite the fact that few elements are accessible for AI based malware indicators, foes can likewise use highlight related information to foster variations of malware to sidestep identification. Consequently, a vital job of the Android security local area is to continually propose new highlights that can describe malignant practices. In this paper, we propose an original static delicate subgraph-based element (S3 Feature) for Android malware recognition. The broad reception of Android gadgets and the developing fame of portable applications has made the Android stage and android framework to cause the

significant assault focus of malware. A new report passes on that the quantity of malignant versatile applications and assaults saw in the wild is accounted for to have increment dramatically, which forces a colossal danger on portable application markets and versatile clients. Accordingly, there is a crisis need to successfully obstruct them. FAN OU and JIAN XU [19] proposed SSG as a component from the openness of malware practices and change an APK into a component vector addressed by S3Feature for Android malware location. This paper exhibit that those neighborhood practices caught by SSGs are ascribed to progress inside the presentation of malware location by directing complete investigations, which additionally shows that S3Feature can give a new perspective for include designing in the area of malware location. In future, we can overhaul S3Feature by consolidating with dynamic data, for example, information stream-based chart, in compatibility of foil encryption and reflection-base avoidance.

The startling development of android innovation has made an increment in malware and elevated worries over the malware dangers. A large portion of the various methodologies have been taken on to defeat the malware assaults yet at the same time malware assaults are advancing. To conquer this issue, this review research is proposed as an Android malware recognition framework dependent on authorization or solicitation highlights utilizing Bayesian characterization by MOHD FAIZAL AB RAZAK [20]. The authorization highlights were drawn out by means of the static examination strategies. The improvement of cell phones and versatile applications has essentially changed our methods of taking part in our everyday lives Such as Web perusing, E-banking, online 10 shopping, long range interpersonal communication and web-based learning are a portion of the instances of administrations furnished by cell phones by means of association with the Internet. In this manner, cell phones have assumed a significant part and have turned into an essential piece of human existence. This review has proposed an investigation of authorization-based highlights utilizing static examination. To improve the malware discovery, these highlights were streamlined utilizing data gain calculation and chi-square calculations. In this paper study, the chose authorization highlights were isolated into different gatherings to recognize the best presentation of precision.

Due to flexible distributing approaches of android, many organizations and associations are making a few applications to serve and fulfill client needs. The authority market of Android Google Play store is distinguished by its help for the informal stores and organizations, and Android Google Play Store won't force numerous limitations on android engineers during their distributing interaction. In this manner, these are the significant justification for making. Due to previously

mentioned reasons, malware become the weakest stage to digital hoodlums, cyberbullies and so on, hence clients experience the ill effects of the issues of openness to malignant applications which abuse their protection or harm their gadgets. MOTHANNA ALMAHMOUDA et al [21] proposed a clever model is drawn up dependent on an association of four Static Features which are Permissions, API calls, Monitoring System Events, and Permission Rate. The proposed model has acquired exactness and furthermore has a promising outcome gotten for android malware identification. In this exploration, a deep learning order model was created for Android malware recognition dependent on static highlights. The last type of the dataset was contained 46 preparing characteristics and one result trait which is paired classes i.e., malware and harmless. Five AI (ML) classifiers and one profound learning (DL) classifiers were evaluated by utilizing 10-folds cross approval strategy, and the result uncovered that RF was the most effective AI classifier after SVM for identifying Android malware.

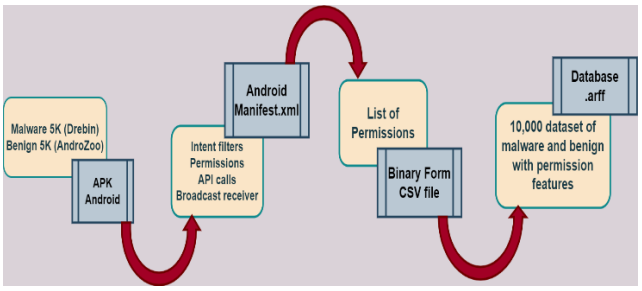


***FIGURE 3.3: DATA COLLECTION AND DATA ANALYSIS PHASE***

Out of a few versatile terminals, Android framework is widely preferred by clients, the justification behind this is, android is an open stage and has rich programming content and administrations accessible. Simultaneously, Android malware is additionally progressing to gain ill-advised benefits, which carries true security issues to the Android stage. Thusly, this exploration is forced to give a powerful security instrument by recognizing pernicious programming. Research depends on Android working framework and AI (ML) innovation. The vital job of this paper is to distinguish Android malware from two parts of static examination and dynamic investigation. In the beyond couple of years, with the advancement of portable Internet, versatile terminals have turned into the significant device for individuals to accumulate data. Among them, the brilliant telephones conveying the Android framework are by and by the quickest developing cell phones with a high portion of the overall industry. Nonetheless, with the fast advancement of the Android market, the security issues are turning out to be increasingly genuine, and malware and pernicious assaults are progressively showing up. QING-FEI WANG and XIANG FANG [22] comprehend that by

applying AI to the location of Android malware, it is conceivable to viably identify the malware. 7 Consolidating static investigations with dynamic investigation can upgrade recognition precision and productivity at a similar time. Malevolent program is a malware which is continually developing refinement. Past research proposition which identifies the malware essentially focused on programming-based locators that are powerless against being compromised. Of late work has been proposed equipment helped malware recognition. Consequently, in this research proposition, ZHIXING XU SAYAK RAY et al [23] starts another structure for equipment helped malware identification grounded on observing and characterizing memory access designs utilizing Machine Learning (ML). Malevolent program characterized as malware which is truly developing security danger and the recognition of malware stays as a critical space of examination. Examination is the underlying advance in identification of malware. This includes either static examination or dynamic investigation of known malware and is generally acted in disconnected with master human information. This paper has presented a system for recognizing malware dependent on internet-based investigation of virtual memory access designs by utilizing AI (ML). This system was applied to the application-explicit malware location situation which means to recognize malware tainted runs of known applications. In this exploration, we tended to the test of online memory information assortment by utilizing a framework or capacity call age-based memory access synopsis.

With the appearance of PDAs/android mobiles, the notoriety of free Android applications has risen rapidly which prompted vindictive Android applications being unequivocally introduced in an android portable which caused to abuse of the client's protection or lead assault on client's information. This worry prompted foster Malware recognition on Android stages in view of the unsuitable comparability between noxious conduct and harmless conduct. Consequently, XINNING WANG and CHONG LI [24] proposed an article which deals with various dimensional, portion highlight-based structures and element Weight-Based Detection (WBD) intended to classify and grasp the qualities of Android malware and harmless applications. The advancing piece of the pie of Android stages has been joined by the uncommon increment of malignant dangers, which incorporates electronic and application-based dangers. As differentiation to online vindictive dangers, which exploit weak sites to infuse malware into clients' telephones, though, application-put together dangers concentrate with respect to taking on the appearance of authentic applications to hoodwink clients for introducing and executing them. Android is a versatile working framework (OS) and created by numerous designers from Open Handset

Alliance. The author dissected the exhibition issues for picking applicable elements that are successful for distinguishing vindictive applications on the Android stages. In view of these, we planned a different dimensional bit include based malware recognition foundation and furthermore executed a numerous dimensional part element's assortment specialist to progressively gather the information, move the information, and store our 112-aspect information. The paper results exhibit that the portrayal of bit highlights is straightforwardly the material in foreseeing precisely varying sorts of Android malware.

Currently, Malicious is one of the major cyberhate hazards as there is an enhancement in digital applications such as android application and web applications. Thus, the significant part of security is to identify the malware in android frameworks and PC frameworks. Therefore, Sunitha Choudhary et al have proposed a research paper on detecting the malicious and classifying the malicious by utilizing Machine Learning Concepts. By utilizing the antimalware tools for identifying the malwares will assists us to understand the capability of Algorithms of Machine Learning over the tradition methods. The method used to identify the malware is done by behavioral-based strategies and pattern which assist with static and dynamic strategies. Static and Dynamic methods, behavioral-based method, Ripper Algorithm, Naïve Bayes and Multi-naive Bayes are the algorithm used by this author. on each and every dataset, few Machine Learning algorithms are applied which are Support Vector Machine (SVM), KNN, Decision Tree and also multi-layer perceptron. According to this review, the author Sunitha Chowdary has comprehended the capability of Machine Learning can be incredible assistance in detecting the malicious.

# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENTS

According to software engineers, Functional requirement is defined as a function of a software or components. Function contains a set of codes. When function is called, it performs a specific task. Function takes parameters as input and return an output.

### 4.1.1 USER CLASSES AND CHARACTERISTICS OF USER CLASSES:

**User class 1:** Admin

**User class 2:** client

- Admin is one type of users of android application where these users will train the model with different types of malwares and with new types.
- Client is another type of users of android application who will use the application with malware-free.

### 4.1.2 ASSUMPTION AND DEPENDENCIES

We gather the data regarding malware as well as good ware and clubbed as dataset. These datasets are trained and passed to the system. System will assume that passed data is correctly trained and it has been correctly maintained to identify the malware.

## 4.2 NON-FUNCTIONAL REQUIREMENTS

According to system engineers and requirement engineers, A non-functional requirements are defined as a requirements certain norm which can be utilized to determine the operations / tasks of the system on behalf of actions.

### 4.2.1 ACCESSIBILITY

Accessibility is the act of making your sites usable by whatever number of individuals as could reasonably be expected. We generally consider this to be about individuals with incapacities, however, the act of making locales available likewise helps different gatherings like those utilizing cell phones, or those with slow organization associations.

### 4.2.2 MAINTAINABILITY

In computer programming, maintainability is the simplicity with which a product item can be adjusted to:

- Right deformities
- Meet new prerequisites

New functionalities can be included in the venture dependent on the user necessities just by adding the suitable records to an existing project.

Maintainability is characterized as the likelihood of playing out an effective fix activity inside a given time. All in all, practicality estimates the straightforwardness and speed with which a framework can be re-established to functional status after a disappointment happens. This is like framework dependability examination with the exception of that the arbitrary variable of interest in practicality investigation is an ideal opportunity to fix instead of timing to disappointment.

### 4.2.3 SCALABILITY

Adaptability is the proportion of a framework's capacity to increment or abatement in execution and cost because of changes in application and framework handling requests.

### 4.2.4 PORTABILITY

Programming portability is the likelihood to involve similar programming in various conditions. It applies to the product that is accessible for at least two unique stages or can be recompiled for them. Portability is one of the vital ideas of undeniable-level programming. Portability is the product codebase component to have the option to reuse the current code as opposed to making new code while moving programming from a climate to another. The task can be executed under various activity conditions given it meets its base designs. Just framework documents and dependent congregations would need to be designed in such a case.

## 4.3 DOMAIN AND UI REQUIREMENTS

IDE stands for "Integrated Development Environment". IDE is a program which is devoted for software programming and development of software. Name “IDE “itself indicates that it incorporates with few apparatuses and tools explicitly which are intended for software development. These apparatuses normal includes:

- Auto correction editor which displays the syntax while typing and highlights the keywords while typing.
- Tools to build or construct the code, execution tools like run execution, add breakpoints, toggles etc., troubleshooting devices.

One of the major properties of IDE is that it supports a wide range of programming languages. But IDE is large because it consists of many new features and takes lot of time to download and install ide.

### PREREQUISITES FOR A GOOD PYTHON CODING ENVIRONMENT

- IDE or editor should be able to save the created or edited documents. When the user reopens the document, editor should have ability to reload the saved contents.
- IDE should be able to execute the program without exiting from the editor.
- Troubleshooting support is one of the major prerequisites of IDE. Having the option to venture through your code as it runs is the major feature of IDEs.
- Having the option to rapidly detect the keywords, variables, syntax in editor makes easier to code for programmers. Syntax featuring in IDE is important.

### PYTHON SUPPORT

Python supports general editors and IDE ‘s such as Eclipse +Pydev software, Sublime Text, Visual Studio, Visual Studio Code Software etc. Along with these software’s there are few pythons’ specific editors and IDE such as Pycharm, Spyder etc.

In this project, Pycharm IDE is used for coding. This editor provides a lot of library supports, and it also have a supportive community. This editor allows the programmers to create the code, edit the code, run the code and troubleshoot the code without exiting the editor.

## 4.4 HARDWARE AND SOFTWARE REQUIREMENTS

### HARDWARE REQUIREMENTS

- Processor : Any Processor above 500 MHz
- RAM : 512Mb
- Hard Disk : 10 GB
- Input device : Standard Keyboard and Mouse
- Output device : VGA and High-resolution Monitor

### SOFTWARE REQUIREMENTS

- Operating system : Windows XP
- Language : Python
- Domain : Machine Learning

# CHAPTER 5

# SYSTEM ARCHITECTURE

## 5.1 DESIGN GOALS

The fundamental objective of the proposed framework is identifying and examining important data that leaving the framework. The proposed Methodology includes two units: highlight extraction utilizing the Androguard apparatus and element determination utilizing a Genetic Algorithm. At long last, the chose features are taken as input to ML functions for assessment purposes. To empower secure re-appropriating of record under the previously mentioned model, our project configuration ought to accomplish the accompanying security and performances ensures:

### 5.1.1 INPUT PRIVACY

Input privacy is an assurance that at least one individual can take part in the execution so that neither one of the gatherings picks up anything about the other party's contributions to the execution.

### 5.1.2 OUTPUT PRIVACY

Output privacy is tied in with guaranteeing that specific subsets of data don't endure the data stream.

### 5.1.3 ROBUSTNESS

The term robustness portrays the code's capacity to adapt to unforeseen or flawed information. A genuine illustration of such defective information is the program database document design.

### 5.1.4 EFFICIENCY

Effectiveness is estimated by task-consummation rates and other test measurements. Data design and semiotics play an enormous roll in the adequacy of a site. Other significant regions that influence viability are; page format, picture choice, and content.

### 5.1.5 SIMPLICITY

The beneficial thing about effortlessness is that it is smart for both programming tasks and security. Everybody concurs that keeping it straightforward is a word of advice.

### 5.1.6 OPENNESS

Since we can't give all usefulness and we additionally don't have any desire to re-carry out existing calculations, it was likewise a significant expect to give an adequate transparency, which means similarity with other class libraries yet in addition similarity with the Standard Template Library. Moreover, transparency infers extendibility and measured quality, for example it ought to be easy to add new usefulness and information structures without changing the current code.

### 5.1.7 CORRECTNESS

Accuracy from a programming viewpoint can be characterized as the adherence to the determinations that decide how clients can connect with the product and how the product ought to act when it is utilized accurately. On the off chance that the product acts mistakenly, it may invest in some opportunity to accomplish the undertaking or here and there it is difficult to accomplish it.

### 5.1.8 FLEXIBILITY

Adaptability is utilized as a characteristic of different sorts of frameworks. In the field of designing frameworks plan, it alludes to plans that can adjust when outside changes happen.

### 5.1.9 SCALABILITY

Adaptability is the proportion of a framework's capacity to increment or abatement in execution and cost because of changes in application and framework handling requests.

### 5.1.10 SECURITY

Programming security is a thought executed to ensure programming against pernicious assault and other programmer chances with the goal that the product keeps on working accurately under such possible dangers. Security is important to give trustworthiness, validation and accessibility.

## 5.2 SYSTEM ARCHITECTURE

Two categories of Android Apps or APKs: Malware/Good ware are figured out to extricate highlights, for example, consents and count of App Components like Activity, Services, Content Providers, and so on These elements are utilized as an element vector with class names as Malware and good ware addressed by 0 and 1 separately in CSV format. To lessen the dimensionality of the list of capabilities, the CSV is taken care of to the Genetic Algorithm to choose the most advanced arrangement of highlights. The advanced arrangement of elements acquired is utilized for preparing two ML classifiers: Support Vector Machine and Neural Network.

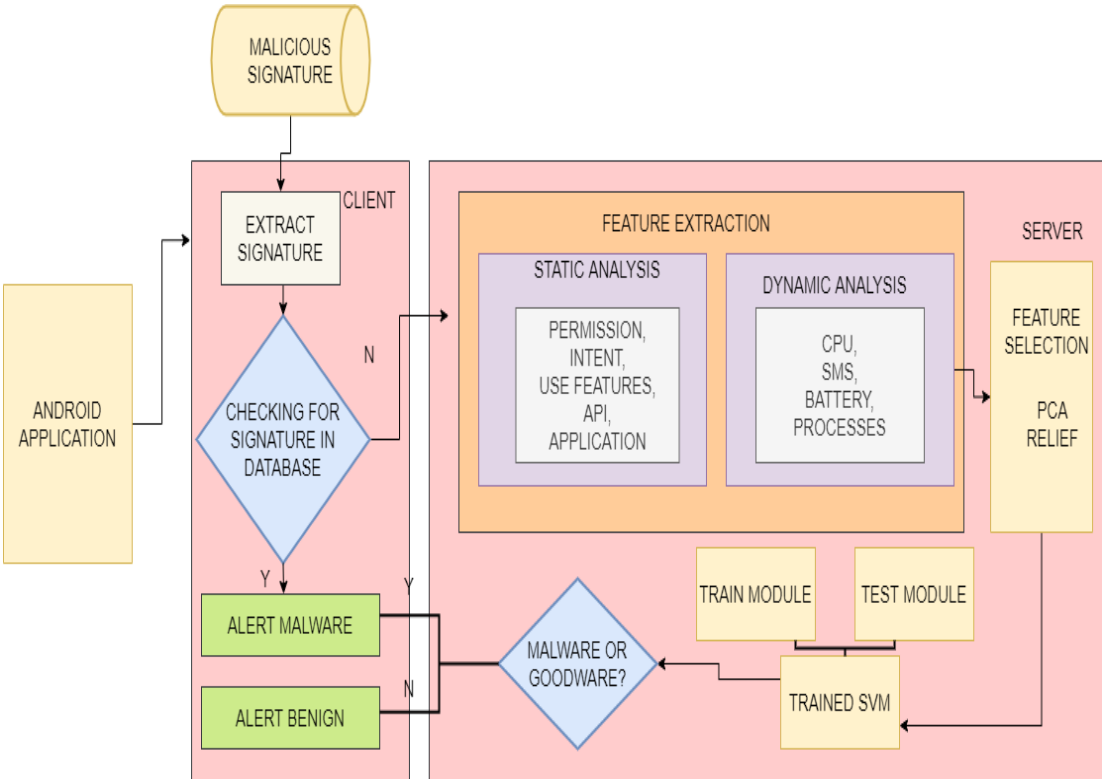


***FIGURE 5.1 SYSTEM FRAMEWORK***

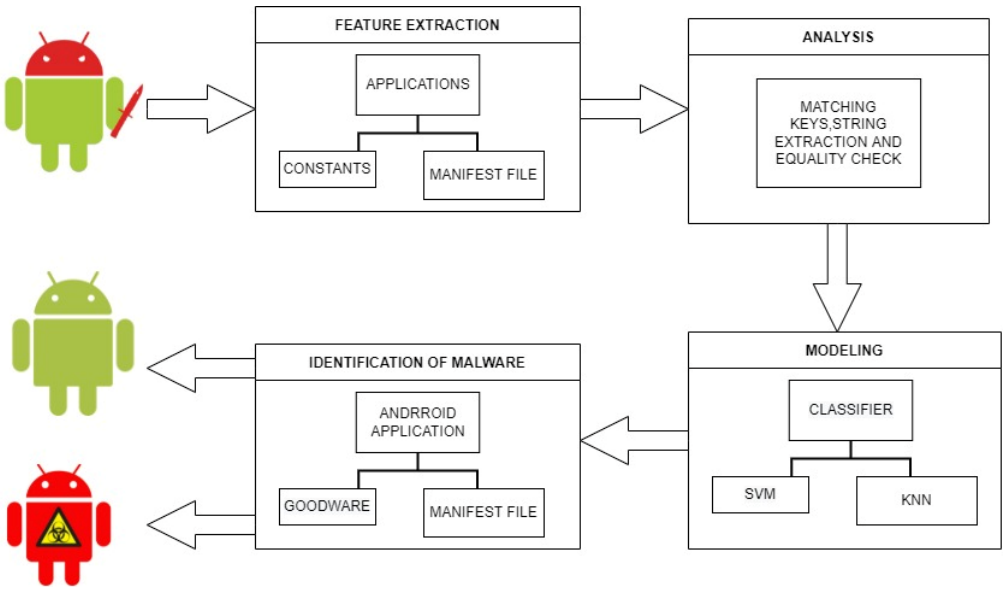


*FIGURE 5.2: SYSTEM ARCHITECTURE*

## 5.3 DATA FLOW DIAGRAM

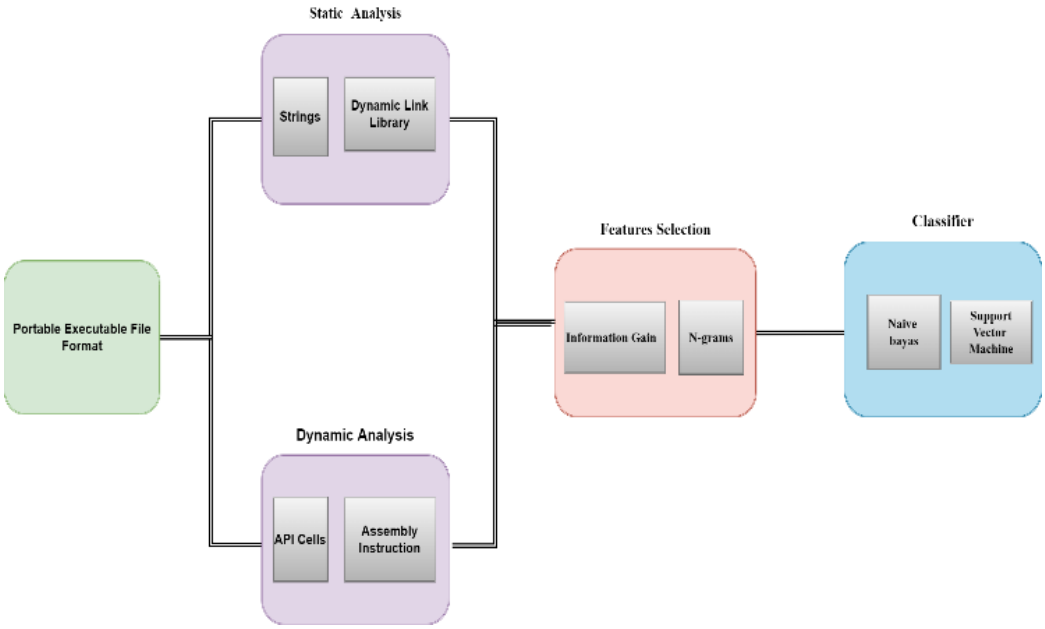


*FIGURE 5.3: DATA FLOW DIAGRAM*

## 5.4 SEQUENCE DIAGRAM

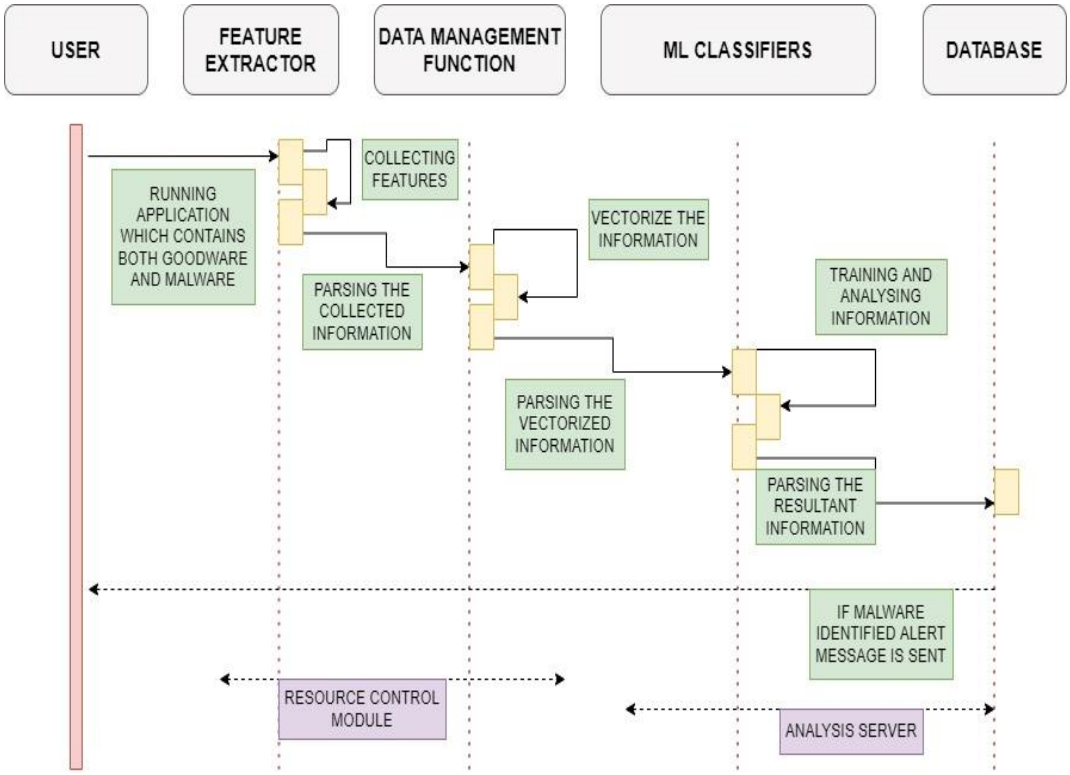


***FIGURE 5.4: SEQUENCE DIAGRAM***

## 5.5 FUTURE WORK

- ML showed a superior methodology with higher identification precision than different methodologies most particularly the hybrid-based. Analysts need to investigate the advancement of a further developed component in the space of ML by investigating a greater amount of the deep learning procedures in the identification of Android malware and preparing the model with enormous datasets for full use of the model.
- Despite the fact that there are many surveys that have been led to feature the works accomplished in the Android malware dissecting area, there is no complete scientific categorization for all examination patterns in this space. Moreover, none of the current survey papers contains a schematic model that makes it simple for the peruse to know the techniques and approaches utilized in a specific field of exploration absent a lot of exertion. This paper targets proposing a far-reaching

scientific categorization and recommending a point-by-point schematic audit approach.

- A novel detailed schematic model called Schematic Review has been created. It has been seen that the vast majority of the concentrated works utilize the static investigation strategy. Likewise, it has been noticed that the majority of the recently led works don't address the examination avoidances strategies, for example, Native code utilization, Dynamic code stacking, repackaging, or code encryption.
- Furthermore, we have seen that the greater part of the works that looked thorough somewhat deal with issues, for example, expanding intricacy and computational time or it is un-computerized systems.
- We likewise noticed that the malware-representation based investigation strategy has been utilized in a tiny number of the concentrated on works despite the fact that of its accomplishment in the work area malware recognition space. Also, practically all static works have been done dependent on the bytecode level, just in two investigations, the application was broken down dependent on the Native code level, and no review has examined the two-level of code. Along these lines, the normal flimsy part of all planned static investigation-based methodologies is the examination of the Native code.
- It has been noticed that semantic elements were not utilized widely in concentrated works. Moreover, the elements' designing and determination techniques were utilized in a couple of concentrates in particular.
- As far as the utilized dataset, we noticed that the vast majority of the created structures have been assessed utilizing a harmless dataset that was downloaded from the authority market, and notable noxious datasets like Darebin, Malgenom as malware datasets.
- In a little piece of the works, a blend of applications downloaded from the authority market and the outsider business sectors have been utilized as a harmless dataset. Also, a blend of the notable vindictive datasets and a few

examples gathered from the web have been utilized as malevolent datasets in a few works.

- Likewise, the powerful investigation downsides have not been tended to in a large portion of the works which utilize this examining strategy (regardless of whether dynamic or half and half examination systems). Additionally, the majority of the proposed dynamic structures experience the ill effects of expanded overhead and computational intricacy.
- Besides, a large portion of the powerful examination systems utilize irregular-based occasions age apparatuses for associating with the tried program, so some application execution ways can be missed. Consequently, there is an earnest requirement for more profound and more thorough investigation strategies to such an extent that all malware designers' disguise advancements like jumbling, dynamic stacking, local code… and so forth can be tended to.
- Also, the proposed instruments ought to keep up with the exhibition at a satisfactory level and the required client intercession ought to be pretty much as low as could really be expected. In this manner, we propose developing the application markets' future security instruments dependent on multi-level investigation systems. All in all, the applications ought to be sifted by their seriousness level so few applications arrive at the phases of investigation that need an incredible examining cost.
- Consequently, a mark-based or heuristic-based technique can be utilized in the principal level, to this end, a lightweight marks data set ought to be built, and the applications are coordinated with it. In the event that the application has not matched any signature, it will be moved to the second level wherein a lightweight static examination-based strategy can be utilized. Assuming that the application can be decided to be harmless or malignant with no question the examination will be finished.
- Then again, if the application has dubious conduct or in the event that that the application utilizes obscurity, dynamic substance stacking procedures… and so forth it will be moved to the third investigating level, which we propose to be a

unique examination strategy. In the third level, the application will be executed in breaking down climate and its conduct will be concentrated by extricating however many as could be allowed dynamic highlights. On the off chance that a choice can't be taken the removed static and dynamic highlights can be dissected in a hybridized way.

- Also, it is feasible to add one more level to this model, so that assuming the framework can't judge the application's conduct plainly, the reports can be considered by the investigator physically to give an ultimate conclusion.
- Additionally, the mark data set ought to be refreshed by the choices that are taken in the breaking down levels. Utilizing this technique, the applications are sifted with the goal that a couple of uses will arrive at the last stage.
- Therefore, only a couple of utilizations will take a great deal of examination time, hence it will essentially diminish the general overhead contrasted and performing cross breed or dynamic investigation on all applications which will be dissected.
- All the more significantly, the App Store will have an undeniable degree of assurance. Through the broad review done in this paper, a few focuses that ought to be centered around in later works were distinguished.

# CHAPTER 6

# IMPLEMENTATION

## 6.1 STRATEGY

- Two set of Android Apps or APKs: Malware/Good ware are reverse engineered to extract features such as permissions and count of App Components such as Activity, Services, Content Providers, etc. These features are used as feature vector with class labels as Malware and Good ware represented by 0 and 1 respectively in CSV format.
- To reduce dimensionality of feature-set, the CSV is fed to Genetic Algorithm to select the most optimized set of features. The optimized set of features obtained is used for training two machine learning classifiers: Support Vector Machine and Neural Network.
- In the proposed methodology, static features are obtained from AndroidManifest.xml which contains all the important information needed by any Android platform about the Apps.

## 6.2 OUTPUT SNAPSHOTS

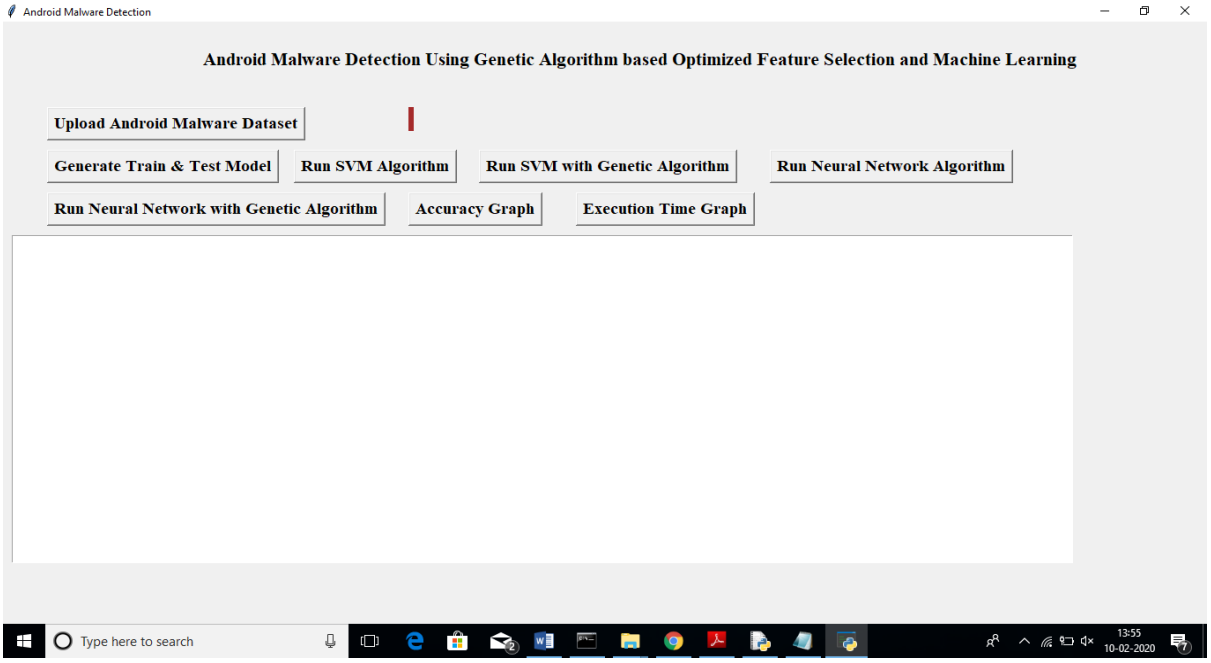


*FIGURE 5.6.1: HOME SCREEN*

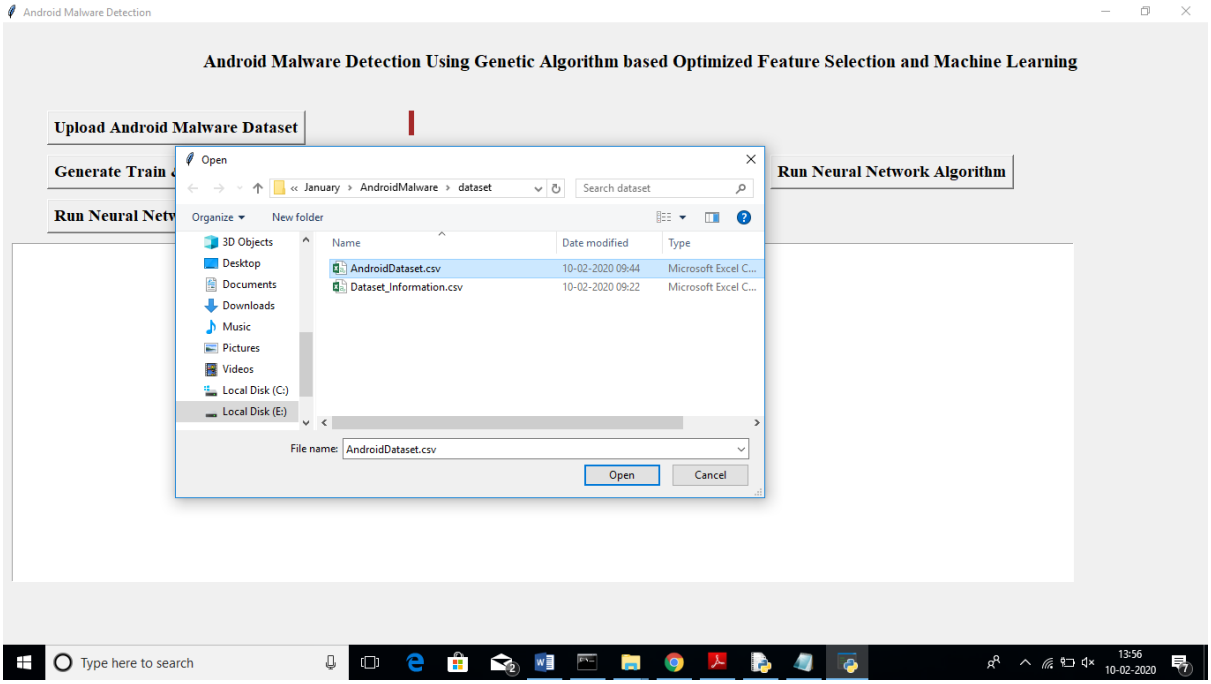


*FIGURE 5.6.2: UPLOADING DATASET*

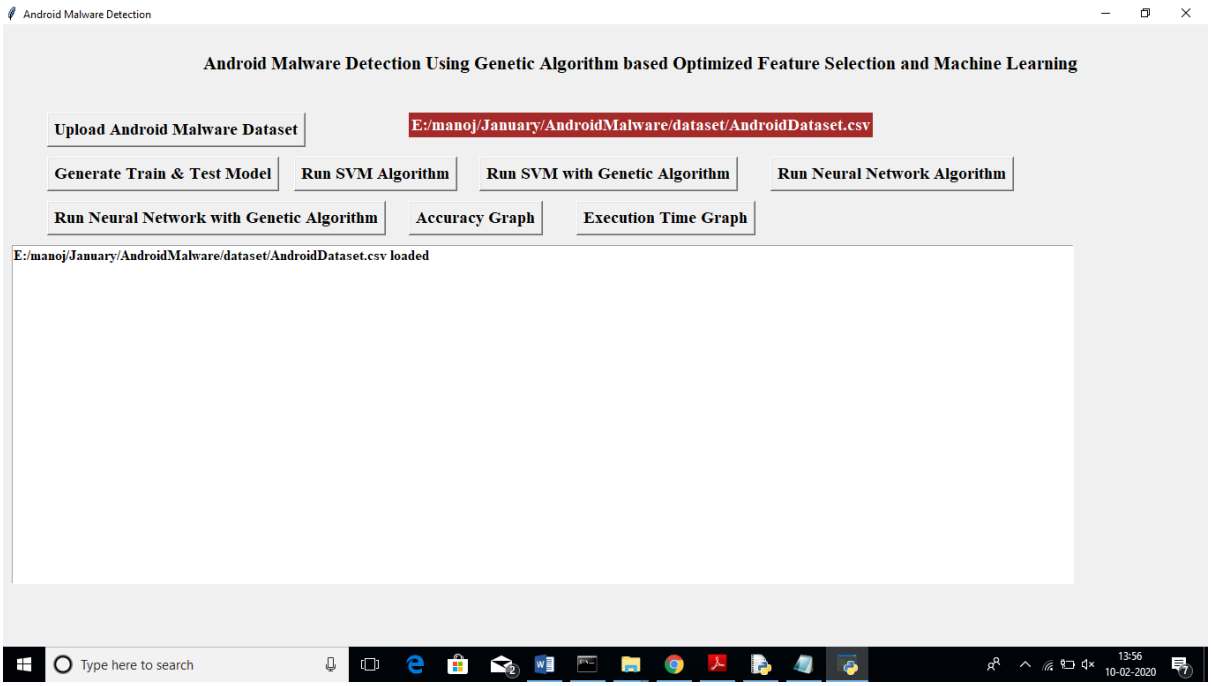


***FIGURE 5.6.3: DATASET UPLOADED***

All machine learning algorithms will take 80% dataset for training and 20% dataset to test accuracy of trained model. After clicking that button will get train and test model

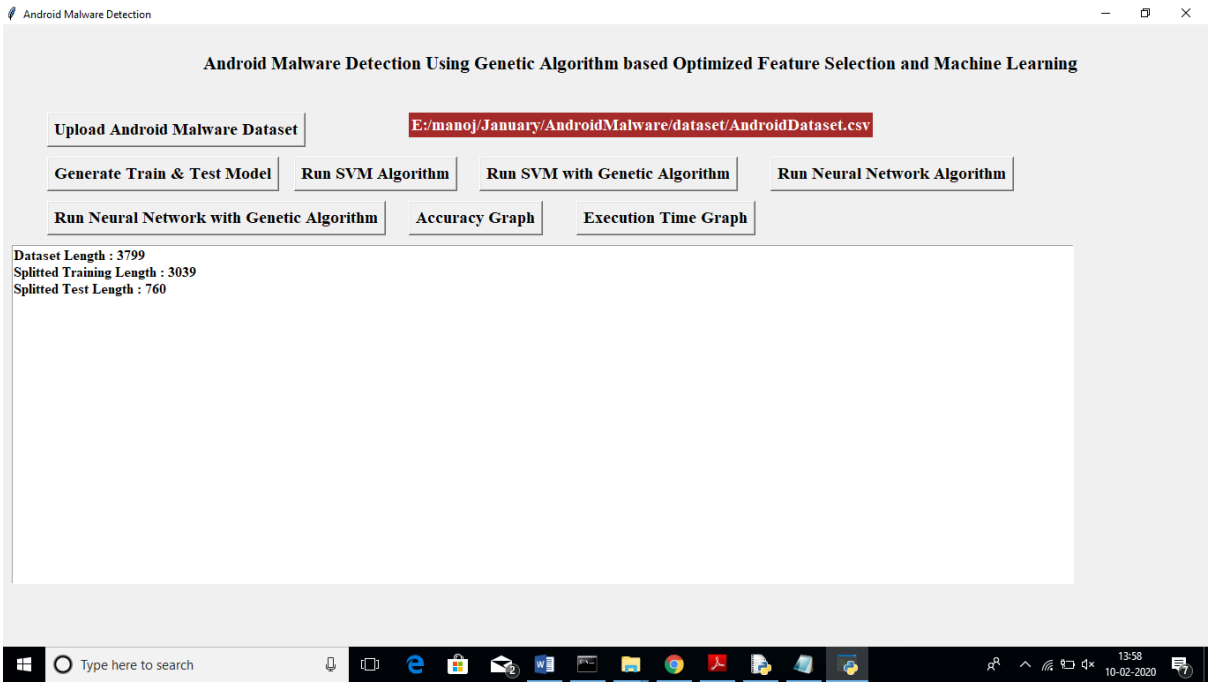


***FIGURE 5.6.4: GENERATE TRAIN AND TEST MODEL***

In above screen we can see there are total 3799 android app records are there and application using 3039 records for training and 760 records for testing. Now we have both train and test model .

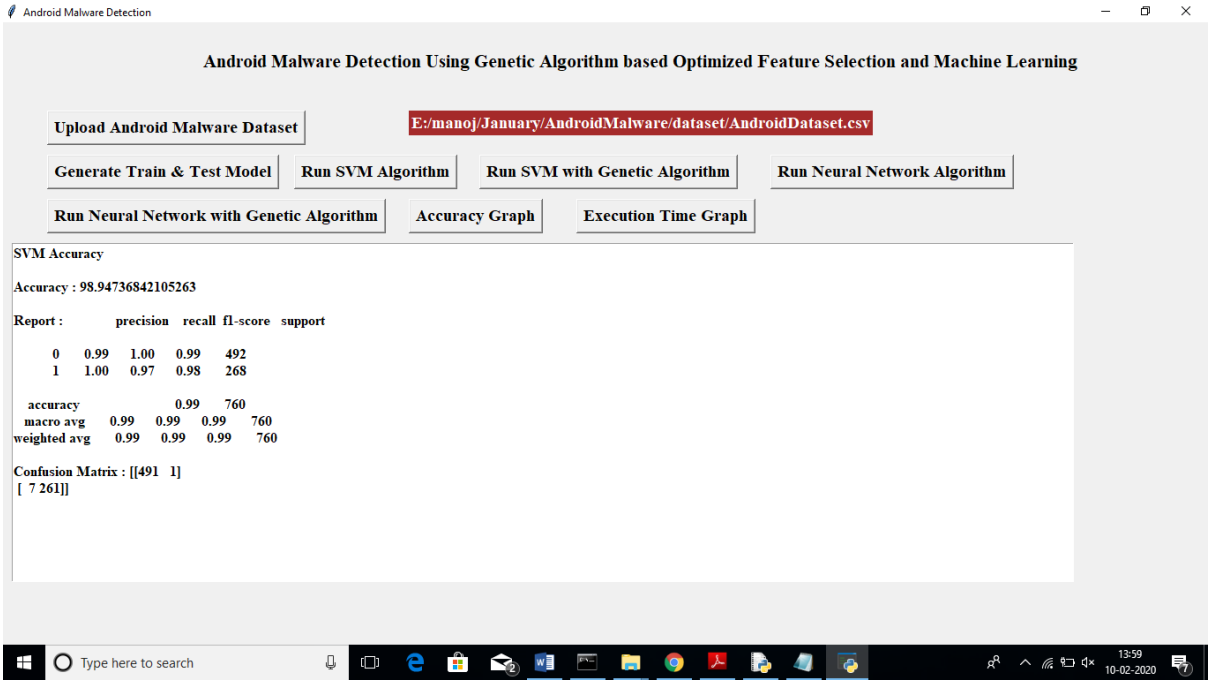


***FIGURE 5.6.5: SVM ALGORITHM***

In above screen we got 98% accuracy for SVM

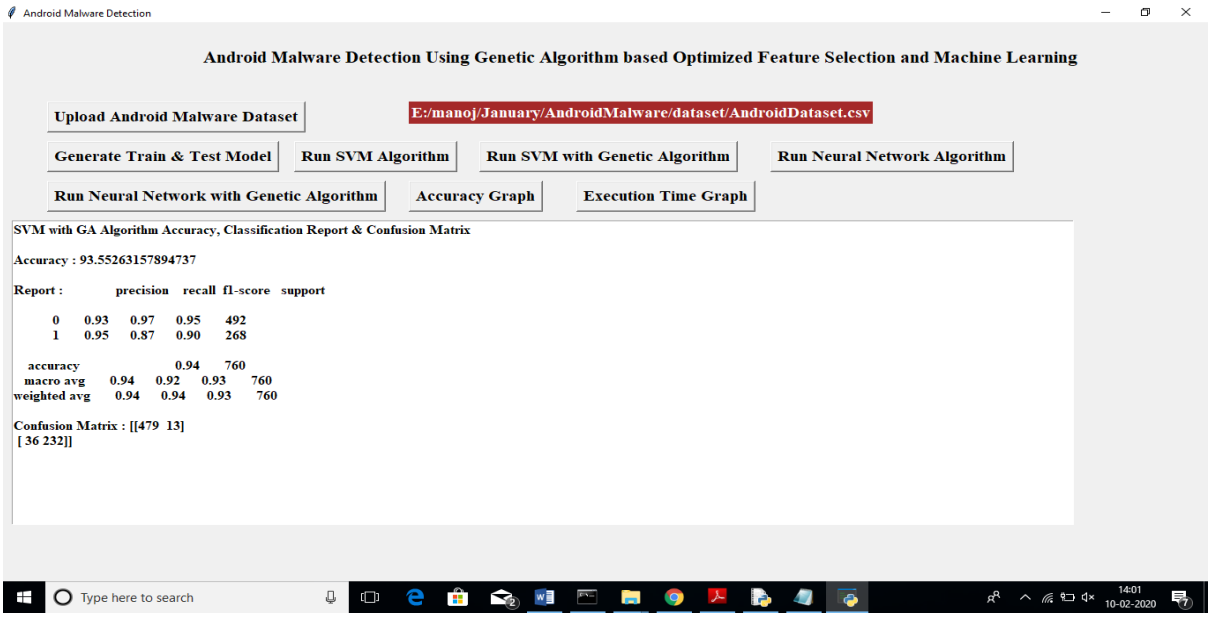


***FIGURE 5.6.6: SVM WITH GENETIC ALGORITHM***

In above screen SVM with Genetic algorithm got 93% accuracy. Genetic with SVM accuracy is less but its execution time will be less which we can see at the time of comparison graph.

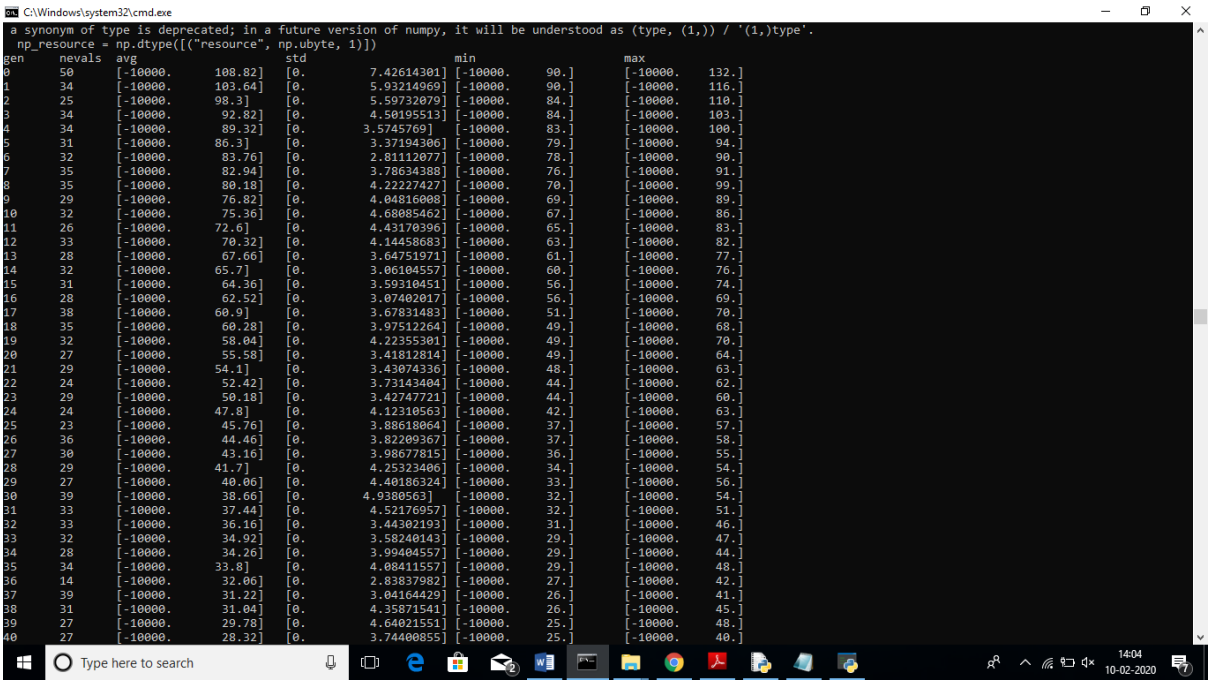


***FIGURE 5.6.7: FEATURES SELECTED***

In above console we can see genetic algorithm chooses 40 features from all dataset features.

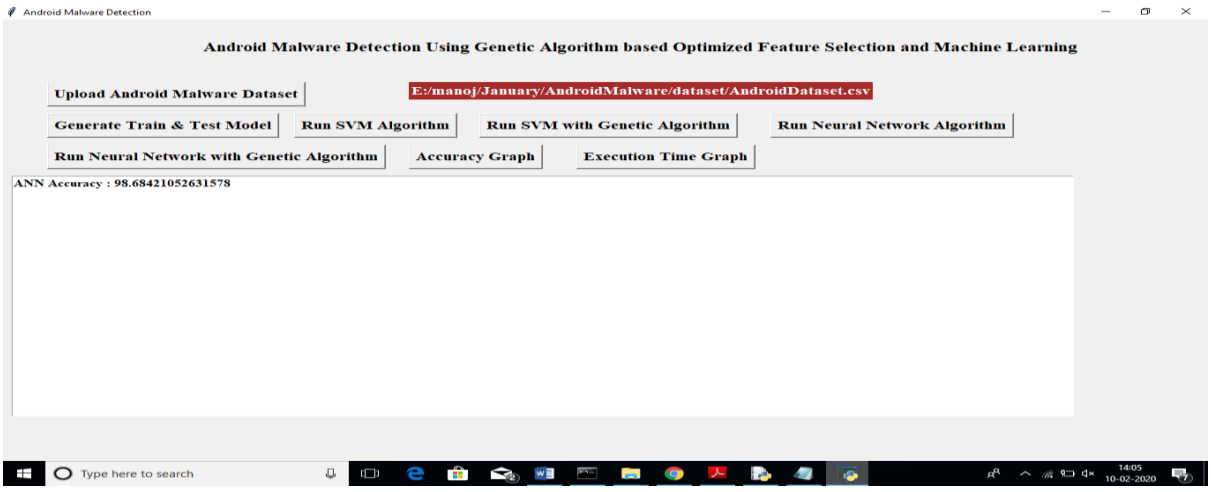


***FIGURE 5.6.8: NEURAL NETWORK ALGORITHM***

In above screen neural network also gave 98.64% accuracy.

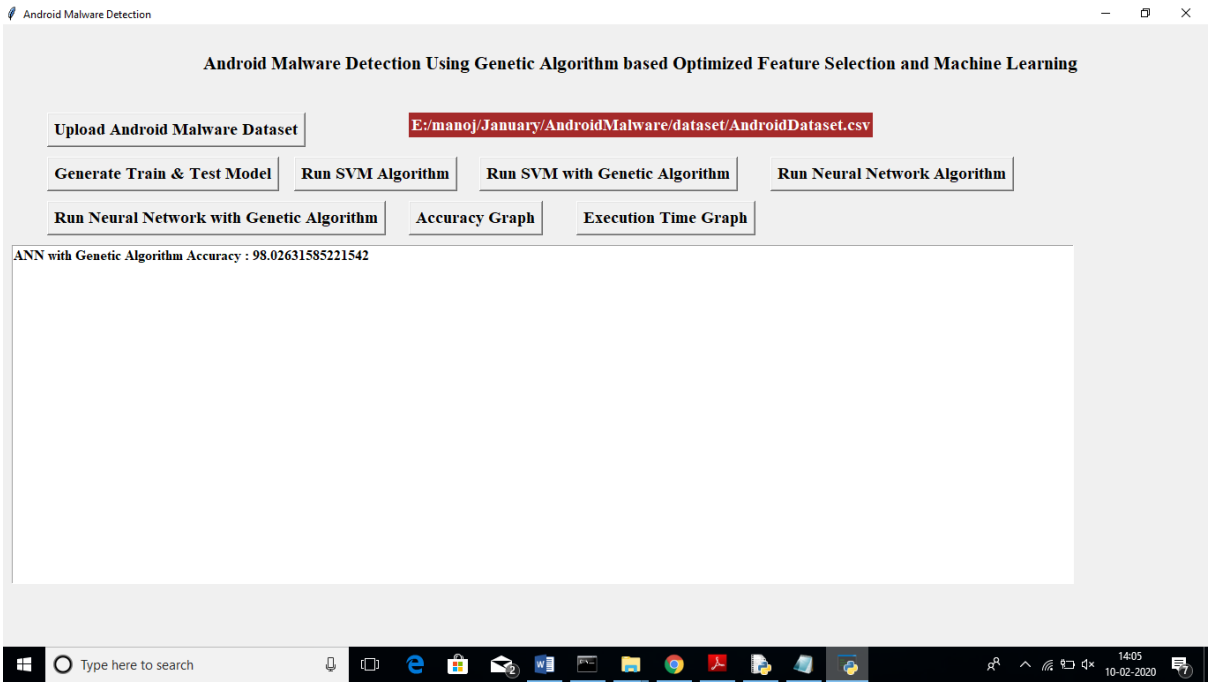


***FIGURE 5.6.9: NEURAL NETWORK WITH GENETIC ALGORITHM***

In above screen NN with genetic got 98.02% accuracy.


***FIGURE 5.6.10: ACCURACY GRAPH***

In above graph x-axis represents algorithm name and y-axis represents accuracy and in all SVM got high accuracy.

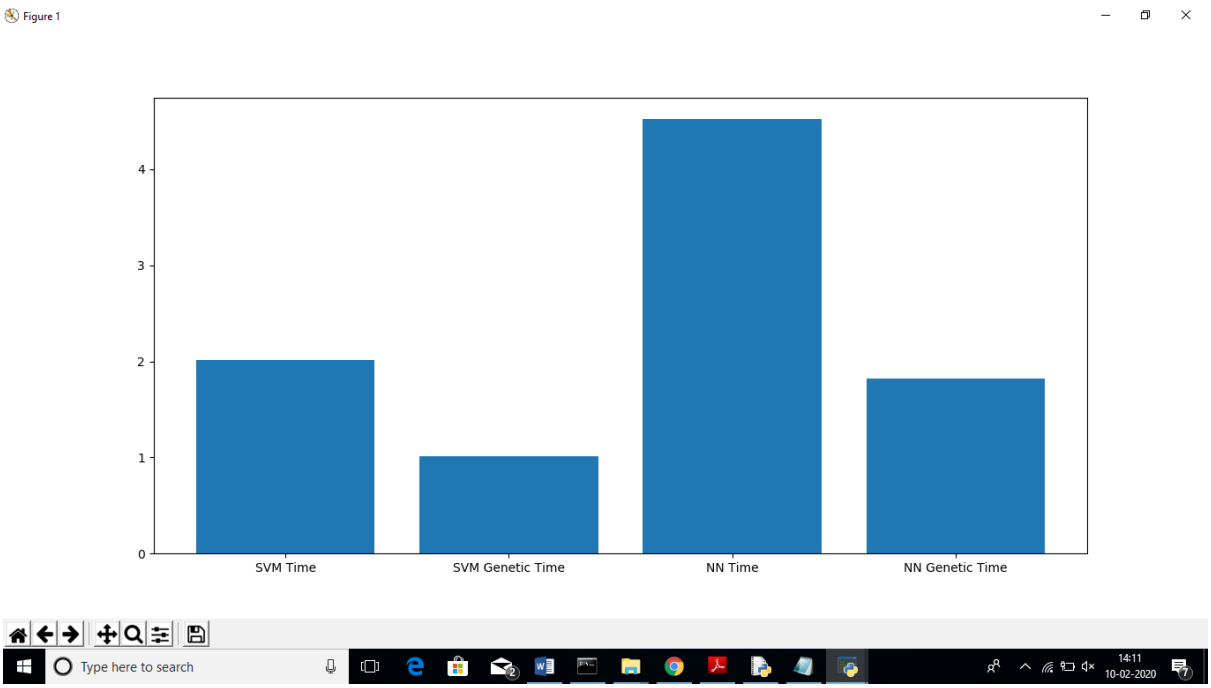


***FIGURE 5.6.11: EXECUTION TIME GRAPH***

In above graph x-axis represents algorithm name and y-axis represents execution time. From above graph we can conclude that with genetic algorithm machine learning algorithms taking less time to build model.

# CHAPTER 7

# CONCLUSION

The quantity of dangers presented to Android stages are extending each day, creeping basically through toxic malwares, in this way a significant job is to design and foster a system that can recognize applications with accurate outcomes. Where signature-based methodology neglects to identify the new variations of malware presenting day-zero threats, ML based procedures are utilized. This proposed system utilizes transformative Genetic Algorithm and to get the most streamlined elements of of subset which can be used to get ready the ML calculations in most capable manner. From the surveys, it tends to be seen that a nice order precision more than 94% is stayed aware of using the Support Vector Machine (SVM) and Neural Network classifiers machine we work on the lower aspect highlight set, thusly, diminishing the preparation intricacy of the classifiers. Further work can be redesigned using greater datasets for additional created results and dissecting the effect on other ML calculations when used related to Genetic Algorithm

# REFERENCES

*[1]Wu, Songyang; Wang, Pan; Li, Xun; Zhang, Yong (2016). Effective Detection of Android Malware Based on the Usage of Data Flow APIs and Machine Learning. Information and Software Technology, (), S0950584916300386– . doi:10.1016/j.infsof.2016.03.004 Zhao,*

*[2]Jingling; Zhang, Suoxing; Liu, Bohan; Cui, Baojiang (2018). [IEEE 2018 27th International Conference on Computer Communication and Networks (ICCCN) - Hangzhou, China (2018.7.30-2018.8.2)] 2018 27th International Conference on Computer Communication and Networks (ICCCN) - Malware Detection Using Machine Learning Based on the Combination of Dynamic and Static Features. , (), 1– 6. doi:10.1109/ICCCN.2018.8487459*

*[3]Liu, Z., Wang, R., Japkowicz, N., Tang, D., Zhang, W., & Zhao, J. (2021). Research on unsupervised feature learning for Android malware detection based on Restricted Boltzmann Machines. Future Generation Computer Systems, 120, 91– 108. doi:10.1016/j.future.2021.02.015*

*[4]Bearden, Ruth; Lo, Dan Chai-Tien (2017). [IEEE 2017 IEEE International Conference on Big Data (Big Data) - Boston, MA, USA (2017.12.11-2017.12.14)] 2017 IEEE International Conference on Big Data (Big Data) - Automated microsoft office macro malware detection using machine learning. , (), 4448– 4452. doi:10.1109/BigData.2017.8258483*

*[5]Gavrilut, Dragos; Cimpoesu, Mihai; Anton, Dan; Ciortuz, Liviu (2009). [IEEE 2009 International Multiconference on Computer Science and Information Technology (IMCSIT) - Mragowo, Poland (2009.10.12-2009.10.14)] 2009 International Multiconference on Computer Science and Information Technology - Malware detection using machine learning. , (), 735– 741. doi:10.1109/imcsit.2009.5352759*

*[6]Syrris, V., & Geneiatakis, D. (2021). On machine learning effectiveness for malware detection in Android OS using static analysis data. Journal of Information Security and Applications, 59, 102794. doi:10.1016/j.jisa.2021.102794*

*[7]Le, Ngoc C.; Nguyen, Tien-Manh; Truong, Trang; Nguyen, Ngoc-Dam; Ngo, Tra (2020). [IEEE 2020 RIVF International Conference on Computing and Communication Technologies (RIVF) - Ho Chi Minh, Vietnam (2020.10.14-2020.10.15)] 2020 RIVF International Conference on Computing and Communication Technologies (RIVF) - A Machine Learning Approach for Real Time Android Malware Detection. , (), 1– 6. doi:10.1109/RIVF48685.2020.9140771*

*[8]Peiravian, Naser; Zhu, Xingquan (2013). [IEEE 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI) - Herndon, VA, USA (2013.11.4-2013.11.6)] 2013 IEEE 25th International Conference on Tools with Artificial Intelligence - Machine Learning for Android Malware Detection Using Permission and API Calls. , (), 300– 305. doi:10.1109/ICTAI.2013.53*

*[9]Liu, Kaijun; Xu, Shengwei; Xu, Guoai; Zhang, Miao; Sun, Dawei; Liu, Haifeng (2020). A Review of Android Malware Detection Approaches based on Machine Learning. IEEE Access, (), 1–1. doi:10.1109/ACCESS.2020.3006143*

*[10]Cai, Minghui; Jiang, Yuan; Gao, Cuiying; Li, Heng; Yuan, Wei (2021). Learning features from enhanced function call graphs for Android malware detection. Neurocomputing, 423(), 301– 307. doi:10.1016/j.neucom.2020.10.054*

*[11]Fatima, Anam; Maurya, Ritesh; Dutta, Malay Kishore; Burget, Radim; Masek, Jan (2019). [IEEE 2019 42nd International Conference on Telecommunications and Signal Processing (TSP) - Budapest, Hungary (2019.7.1-2019.7.3)] 2019 42nd International Conference on Telecommunications and Signal Processing (TSP) - Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning. , (), 220– 223. doi:10.1109/TSP.2019.8769039 9*

*[12]Moon, D., Lee, J., & Yoon, M. (2021). Compact feature hashing for machine learning based malware detection. ICT Express. doi:10.1016/j.icte.2021.08.005*

*[13]Zaabi, A. A., & Mouheb, D. (2020). Android Malware Detection Using Static Features and Machine Learning. 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI). doi:10.1109/ccci49893.2020.925645*

*[14]Firdausi, Ivan; lim, Charles; Erwin, Alva; Nugroho, Anto Satriyo (2010). [IEEE 2010 Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT) - Jakarta, Indonesia (2010.12.2-2010.12.3)] 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies - Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection. , (), 201–203. doi:10.1109/ACT.2010.33*

*[15]Roy, Arindaam; Jas, Divjeet Singh; Jaggi, Gitanjali; Sharma, Kapil (2020). Android Malware Detection based on Vulnerable Feature Aggregation. Procedia Computer Science, 173(), 345– 353. doi:10.1016/j.procs.2020.06.040*

*[16]Sahs, Justin; Khan, Latifur (2012). [IEEE 2012 European Intelligence and Security Informatics Conference (EISIC) - Odense, Denmark (2012.08.22-2012.08.24)] 2012 European Intelligence and Security Informatics Conference - A Machine Learning Approach to Android Malware Detection. , (), 141– 147. doi:10.1109/EISIC.2012.34*

*[17]Alzaylaee, Mohammed K.; Yerima, Suleiman Y.; Sezer, Sakir (2019). DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices. Computers & Security, (), 101663– . doi:10.1016/j.cose.2019.101663*

*[18]Mohamad Arif, J., Ab Razak, M. F., Tuan Mat, S. R., Awang, S., Ismail, N. S. N., & Firdaus, A. (2021). Android mobile malware detection using fuzzy AHP. Journal of Information Security and Applications, 61, 102929. doi:10.1016/j.jisa.2021.102929*
*[19]https://www.sciencedirect.com/science/article/pii/S016 7404821003370*

*[20]https://doi.org/10.1016/j.icte.2021.09.003*

*[21]Almahmoud, M., Alzu'bi, D., & Yaseen, Q. (2021). ReDroidDet: Android Malware Detection Based on Recurrent Neural Network. Procedia Computer Science, 184, 841–846. doi:10.1016/j.procs.2021.03.105*

*[22]Qing-Fei, Wang; Xiang, Fang (2018). [IEEE 2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC) - Wuhan, China (2018.4.19-*

*2018.4.21)] 2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC) - Android Malware Detection Based on Machine Learning. , (), 434– 436. doi:10.1109/ICNISC.2018.00094*

*[23]Wang, X., & Li, C. (2021). Android malware detection through machine learning on kernel task structures. Neurocomputing, 435, 126– 150. doi:10.1016/j.neucom.2020.12.088*

*[24]Xu, Zhixing; Ray, Sayak; Subramanyan, Pramod; Malik, Sharad (2017). [IEEE 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE) - Lausanne, Switzerland (2017.3.27-2017.3.31)] Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017 - Malware detection using machine learning based analysis of virtual memory access patterns. , (), 169–174. doi:10.23919/DATE.2017.7926977*

*[25]Udayakumar, N; Saglani, Vatsal J.; Cupta, Aayush V.; Subbulakshmi, T (2018). [IEEE 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) - Tirunelveli, India (2018.5.11-2018.5.12)] 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) - Malware Classification Using Machine Learning Algorithms. , (), 1– 9. doi:10.1109/ICOEI.2018.8553780*

*[26]Gopal M.K., Amirthavalli M, "Applying machine learning techniques to predict the maintainability of open source software" 2019 International Journal of Engineering and Advanced Technology.*

*[27]Vandana C.P., Chikkamannur A.A. "Feature selection: An empirical study" 2021 International Journal of Engineering Trends and Technology.*

*[28]Gautam K.S., Kaliappan V.K., Akila M, Strategies for Boosted Learning Using VGG 3 and Deep Neural Network as Baseline Models, 2021 Lecture Notes*

*[29] Karthikayini T., Srinath N.K., Comparative Polarity Analysis on Amazon Product Reviews Using Existing Machine Learning Algorithms.*

*[30] Nithya B., Ilango V., Predictive analytics in health care using machine learning tools and techniques.*

*[31] BASKAR P., PADMANABHAN S., ALI M.S., Finite-time H∞ control for a class of Markovian jumping neural networks with distributed time varying delays-LMI approach.*

*[32]Muthuswamy J., Extraction and classification of liver abnormality based on neutrosophic and SVM classifier.*

*[33] Ramkumar M., Ganesh Babu C., Vinoth Kumar K., Hepsiba D., Manjunathan A., Sarath Kumar R., ECG Cardiac arrhythmias Classification using DWT, ICA and MLP Neural Networks*