MALWARE DETECTION USING DYNAMIC MACHINE LEARNING METHODOLOGY

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEM

Submitted By

Amit Singh

2K16/ISY/01

2016-2018

Under the Supervision of:

Dr. Kapil Sharma
(Head Of Department)

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)

2016 - 2018

CANDIDATE'S DECLARATION

I, **Amit Singh** (**2K16/ISY/01)** student of M.Tech **(Information System),** hereby declare that The project Dissertation titled **"MALWARE DETECTION USING DYNAMIC MACHINE LEARNING METHODOLOGY"** which is submitted by me to the **Department of Information Technology**, Delhi Technological University, in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship or some other title or recognition.

Place: Delhi **AMIT SINGH**

Date: **2K16/ISY/01**

CERTIFICATE

This is to certify that project dissertation entitled "**MALWARE DETECTION USING DYNAMIC MACHINE LEARNING METHODOLOGY**" submitted by **Amit Singh (roll no. 2k16/ISY/01)** Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master in Technology (Information System), is a record of a project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full time for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Dr. Kapil Sharma

Professor

Date:

(Head Of Department)

(Department of Information Technology)

Delhi Technological University

ACKNOWLEDGEMENT

I am very thankful to **Dr. Kapil Sharma** (Professor, Information Technology Dept.) and all the faculty members of the Information Technology Dept. of DTU. They all provided us with immense support and guidance for the project.

I would also like to express my gratitude to the university for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

AMIT SINGH

2K16/ISY/01

M.Tech (Information System)

Department of Information Technology

Delhi Technological University, Delhi

ABSTRACT

Malicious programming is bounteous in a universe of endless PC clients, who are always looked with these dangers from different sources like the web, nearby systems and versatile drives. Malware is possibly low to high hazard and can make frameworks work erroneously, take information and even crash.

Malware might be executable or framework library records as infections, worms, Trojans, all went for rupturing the security of the framework and bargaining client protection. Commonly, hostile to infection programming depends on a mark definition framework which continues refreshing from the web and in this manner monitoring known infections. While this might be adequate for home-clients, a security hazard from another infection could undermine a whole undertaking system.

we propose using dynamic machine learning algorithms for higher accuracy in detection with minimum false positive ratio.

KEYWORDS: ANDROID, MALWARE DETECTION, MACHINE LEARNING

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This section vividly explains the domain and purpose of the project. It embarks on by laying emphasis on the popularity of Android Operating System, simultaneously highlighting the loopholes and security issues related to it. Thereafter, this section explores the traditional techniques of identifying malware and safeguarding user data against them. If the further course of the discussion, the need for a robust and behavior-based malware detection system is introduced and all the associated aspects are precisely illustrated.

1.1 ANDROID OS - POPULARITY AND VULNERABILITY

Google's golem package owns nearly ninetieth of the smartphone market, however its quality comes at a price: higher vulnerability. in keeping with this chart by Statista, supported knowledge from CVE Details, Android's package saw the foremost variety of vulnerabilities in 2017.

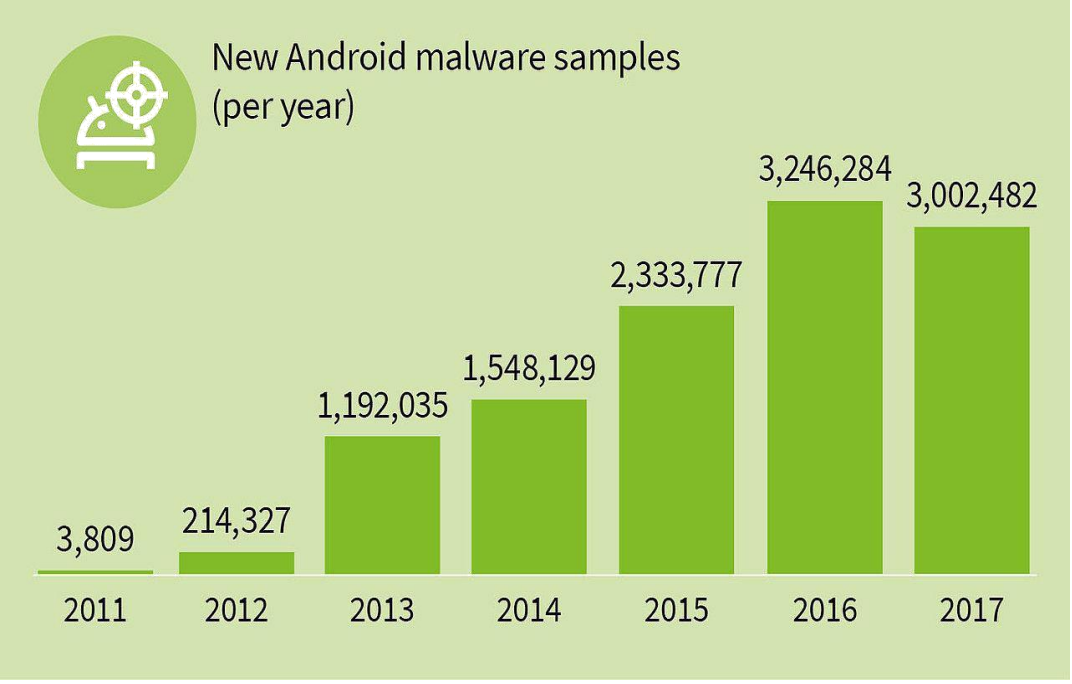

Fig. 1.1 A sample study on trend in malware proliferation [2]

Android is the clear forerunner among experts when it comes to security holes. Developers and researchers alone discovered 841 vulnerabilities among the various versions of the Google operating system in 2017.

A recent report indicates that a new malicious app for Android is introduced every 10 seconds.

Customary security item utilizes the infection scanner to identify malevolent code, these scanner utilizes signature which made by figuring out a malware. In any case, with malware that ended up polymorphic or changeable, the standard mark based discovery strategy used by hostile to infection isn't any more drawn out viable against this issue of malware. Thus, the standard technique that finds the malware upheld the mark can't identify obscure applications. In current hostile to malware item, there's 2 principle undertaking to be circulated from the malware investigation strategy, that region unit malware location and malware characterization.

we present a machine learning-based system for the detection of malware on Android devices.

Machine learning could be a technique that permits computers to find out and improve from their past experiences while not being expressly programmed. In alternative words, we will imagine Machine Learning as a student UN agency learns from their previous mistakes.

1.2 MALWARE TYPES

Virus - This is the most straightforward type of programming. it's simply any bit of code that is stacked and propelled while not client's consent though repeating itself or tainting (adjusting) elective code.

Worm - This malware compose is fundamentally the same as the infection. The distinction is that worm can spread over the system and repeat to different machines.

Trojan - This malware class is used to portray the malware creates that hope to appear as veritable programming. Thusly, the general spreading vector utilized as a part of this class is social outlining, i.e. affecting people to envision that they are downloading the true blue programming.

Adware - The main motivation behind this malware compose is showing notices on the PC.

Spyware -the malware that performs secret activities can be named as spyware. Run of the mill activities of spyware incorporate following inquiry history to send customized promotions, following exercises to pitch them to the outsiders in this manner.

1.3 ANDROID MALWARE DETECTION

In view of the highlights used to group an application, we can sort the examination as Static, Dynamic and Hybrid.

- Static investigation is managed without running an application.
- Dynamic investigation manages highlights that were extricated from the application while running.
- The mixture examination joins the highlights from static and dynamic methods.

1.4 NEED FOR MACHINE LEARNING

With the development of innovation, the quantity of malware is additionally expanding step by step. Malware currently is composed with transformation trademark which causes a colossal development in some of the variety of malware. Not just that, with the assistance of mechanized malware created apparatuses, beginner malware creator is currently ready to effectively produce another variety of malware. With these advancements in new malware, standard stamp based malware area are wound up being unable against the massive variety of malware.

Despite the fact that not broadly actualized, the idea of machine learning strategies for malware identification isn't new. A few kinds of studies were completed in this field, planning to make sense of the exactness of various strategies.

1.5 MACHINE LEARNING METHODS

Hypothetical foundation in machine learning techniques is required for understanding the down to earth usage.These methodologies consolidate k-Nearest Neighbors, Decision Trees, Random Forests, Support Vector Machines and Naive Bayes.

1.5.1 Machine Learning Basics

The quick change of data mining frameworks and techniques realized Machine Learning molding an alternate field of Computer Science.

To build up a more profound comprehension, it merits experiencing the general work process of the machine learning process represented in Fig. 1.2

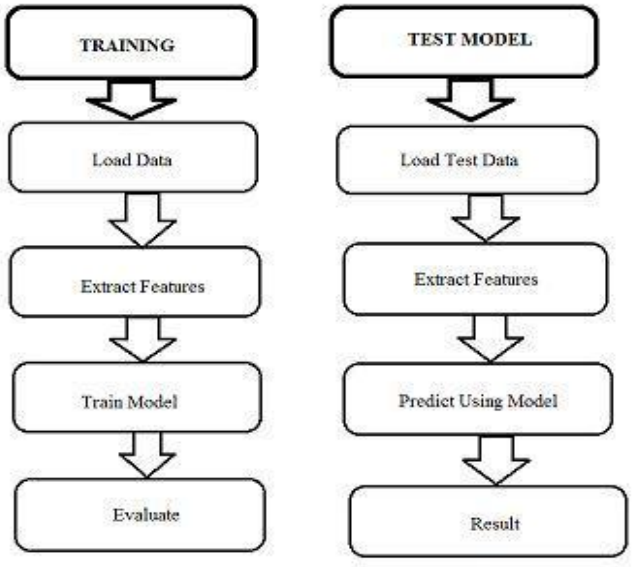

Fig. 1.2 General Workflow of Machine Learning Process

Machine learning work process, by and large, includes two arrangements of tasks: one including preparing the model and the other testing the prepared model with more current datasets.

1.6 Performance Metrics In Machine Learning

1.6.1 Confusion matrix:- A confusion matrix is a technique for summarizing the performance of a classification algorithm.

Confusion Matrix

	P(Predicted)	**N(Predicted)**
P(Actual)	True Positive	False Negative
N(Actual)	False Positive	True Negative

1.6.2 Accuracy:- Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset.

1.6.3 Sensitivity:- Sensitivity (SN) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall (REC) or true positive rate (TPR). The best sensitivity is 1.0, whereas the worst is 0.0

1.6.4 Precision :- Precision (PREC) is calculated as the number of correct positive predictions divided by the total number of positive predictions. It is also called positive predictive value (PPV). The best precision is 1.0, whereas the worst is 0.0.

1.6.5 False positive rate:- False positive rate (FPR) is calculated as the number of incorrect positive predictions divided by the total number of negatives. The best false positive rate is 0.0 whereas the worst is 1.0. It can also be calculated as 1 – specificity.

CHAPTER 2

LITERATURE SURVEY

This part expounds the area of this task alongside correctly determining the requirement for headway in existing philosophies. It reviews all the related work and pinpoints required adjustment along these lines uniting the purpose of this undertaking.

2.1 MACHINE LEARNING

The expression "Machine Learning" was authored by an American pioneer in the field of PC gaming and man-made reasoning named Arthur Samuel in the year 1959. Machine learning investigates the examination and development of calculations that can gain from and make expectations on the given information. Such calculations fabricate a model from the example information sources and utilize them to settle on information driven forecasts or choices. This causes them in conquer following the static program directions.

2.2 MACHINE LEARNING AND CYBER SECURITY

The steady progression in the field of data innovation has prompted a fast improvement of the general public. It has in the end helped in interfacing individuals sitting in two distinct parts of the world and that too inside a couple of moments. All of the administrations can be gotten to while sitting at home. This without a doubt has helped in sparing bunches of time and cash. These administrations frequently require client's delicate information which is should have been ensured keeping in mind the end goal to keep any wrongdoing. Aggressors now and again attempt to get an unapproved access to the client's framework with a specific end goal to either take a critical information or play out any untrustworthy activity. Malware is utilized for such activities.

Keeping in mind the end goal to anchor the framework from different kinds of assaults, a few systems have been formulated inside past years. Each and every strategy gives security in its own special way. However, because of the fast

progression of innovation, assortments of assaults have additionally expanded which are prepared to do going through the customary countermeasure frameworks. Customary malware location frameworks fundamentally incorporate heuristic based and mark based frameworks. They additionally incorporate interruption location frameworks, firewall, and antivirus. These protection frameworks perform activities against known dangers yet are regularly demonstrated unsuccessful against new malware variations. Mark based interruption identification frameworks are a sort of frameworks created for malware discovery. Mark based ID frameworks recognize interruptions by distinguishing occasions and example which coordinate the mark of the known assaults and the request in which they have a tendency to perform. The working of mark based ID is like that of antivirus programming. These kind of frameworks are exceptionally successful towards understood assaults. In any case, Signature-based ID neglects to recognize Zero-Day assaults and different Advanced malware assaults. It can without much of a stretch be avoided by changing the mark of assault. Additionally, the mark database of IDS should be refreshed frequently. This builds the CPU stack as the framework is accused of examining every signature. Heuristic-based interruption identification frameworks likewise called as peculiarity based IDS.The fundamental favorable position of abnormality based IDS over the mark based IDS is that it can distinguish possibly an extensive variety of novel assaults. Anyway numerous a times peculiarity based IDS neglects to identify already known kinds of assaults. Peculiarity based IDS stores particular watched measurements of different novel assaults. On the off chance that these assaults don't stand out to the watched measurements, IDS may neglect to recognize them. Additionally, peculiarity based IDS has high false positive rates.

An IPS gives an unmistakable perspective of the system. It watches out for the suspicious documents. An IPS can help in following the engendering of infection when the infection hits the system. The real disadvantage of IPS is that it has high false positive rates. Likewise some of the time IPS neglects to get a malware. Antivirus (AV) is a PC programming which was initially created to distinguish and expel malware from the PC framework. Be that as it may, with progression in assortments of assaults, antivirus programming began to give insurance from different dangers as well. Present day antivirus can shield a framework from malevolent program aide objects, ransomware, keyloggers, Trojan ponies, worms,

and spyware. A few items can likewise give insurance against spam, phishing assaults, internet managing an account assaults, progressed diligent dangers and so forth however an antivirus programming too accompanies a few drawbacks. The as a matter of first importance drawback of an antivirus programming is that it backs off the PC. AV is exceedingly resourced subordinate. While antivirus checks the PC, the client can barely play out any errand. Now and again the framework additionally solidifies. An antivirus may not utilize all the location procedures. This implies they can't distinguish a wide range of malware and henceforth can't give quality security. To keep away from this an AV's infection database should be always refreshed.

2.3 MALWARE DETECTION METHODS

All malware acknowledgment frameworks can be isolated into signature-based and direct based strategies. Before going into these methods, it is essential to fathom the stray pieces of two malware examination approaches: static and dynamic malware examination. As it recommends from the name, static examination is performed "statically", i.e. without execution of the report. Strangely, dynamic examination is coordinated on the record while it is being executed for example in the virtual machine.

Static examination can be viewed as "scrutinizing" the source code of the malware and trying to infer the lead properties of the record. Static examination can join distinctive techniques :

AV scanning: If the analyzed archive is a striking malware, without a doubt all against contamination scanners will have the ability to recognize it. Regardless of the way that it might have all the earmarks of being immaterial, thusly of disclosure is as often as possible used by AV shippers or sandboxes to "confirm" their results.

Disassembly: This alludes to turning around the machine code to low level computing construct and deducing the product rationale and goals. This is the most widely recognized and solid technique for static examination. Static examination frequently depends on specific devices. Past the straightforward

investigation, they can give data on insurance procedures utilized by malware. The fundamental favorable position of static examination is the capacity to find all conceivable social situations. Looking into the code itself enables the specialist to see all methods for malware execution, that isn't restricted to the present circumstance. In addition, this sort of investigation is more secure than dynamic, since the record isn't executed and it can't bring about awful outcomes for the framework. Then again, static examination is substantially more tedious. In view of these reasons it isn't normally utilized as a part of true unique situations, for example, hostile to infection frameworks, yet is regularly utilized for look into purposes, e.g. when creating marks for zero-day malware.

Another examination compose is dynamic examination. Not at all like static examination, here the direct of the record is watched while it is executing and the properties and points of the report are instigated from that information. By and large, the archive is continued running in the virtual condition, for example in the sandbox. In the midst of this kind of examination, it is possible to find each social trademark, for instance, opened records, made mutexes, et cetera.

2.4 RELATED WORK

Regardless of the way that not by and large realized, machine learning methodologies for malware disclosure isn't new. A couple of sorts of studies were done in this field, planning to understand the accuracy of different techniques. In his paper "Malware Detection Using Machine Learning" Dragos Gavrilut went for working up a revelation structure in light of a couple of changed perceptron computations. For different figurings, he achieved the accuracy of 69.90%-96.18%. It should be communicated that the figurings that realized best exactness in like manner made the most raised number of false-positives: the most correct one achieved 48 false positives. The most "balanced" count with fitting accuracy and the low false-positive rate had the precision of 93.01%. (Gavrilut, et al. 2009).

The paper "Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks" discusses the revelation procedure in light of changed Random Forest estimation in mix with Information Gain for better component depiction. It should be seen, that the instructive file contains essentially of adaptable executable records, for which incorporate extraction is generally less requesting.

A Static Malware Detection System Using Data Mining Methods proposed extraction procedures in perspective of PE headers, DLLs and API limits and strategies in perspective of Naive Bayes, Most hoisted general precision was proficient with the J48 estimation (99% with PE header feature compose and crossbreed PE header and API work incorporate write, 99.1% with API work feature form). (Baldangombo, Jambaljav and Horng 2013). In Zero-day Malware Detection in light of Supervised Learning Algorithms of API call Signatures , the API limits were used for incorporate depiction again. The best result was refined with a Support Vector Machines estimation with institutionalized polykernel. The exactness of 97.6% was refined, with a false positive rate of 0.025. (Alazab, et al. 2011). As it can be seen, all examinations ended up with different results. From here, we can reason that no united system was made yet neither for acknowledgment nor incorporate depiction. The accuracy of each extraordinary case depends upon the specifics of malware families used and on the genuine execution.

Schultz et al. in the first place proposed the idea of recognizing malware in information mining, utilizing Naive Bayes machine learning calculation. They have detailed an aftereffect of 97.11%. One of the prior works in Android malware discovery utilizing application authorizations are they have executed machine learning calculations for arrangement of android applications to distinguish malware. Firdausi et al. utilized J48 choice tree Machine learning calculation for malware location. They have revealed a consequence of 97%.Sebastian et al. utilizing distinctive traits as purposes. They have revealed 96.02% with Random timberland machine learning calculation. Firewall and IDS additionally assumes a critical part in security. Varma et al. utilized Ant Colony Optimization (ACO) in distinguishing irregularities in firewall control arrangement. Varma et al. utilized fluffy harsh component minimization and ACO seek in the advancement of highlight determination for constant Intrusion Detection System (IDS). Choice Trees are ended up being extremely powerful in arrangement issues of system security. Siddiqui et al. utilized irregular backwoods machine learning calculation for distinguishing malware. They have announced a consequence of 96.6%. Anderson et al are utilized help vector machine (SVM) for malware location. They have revealed an aftereffect of 98.07%. There is a need to distinguish the most performing procedure in recognizable proof of malware in android application information. This examination has thought about a few machine learning calculations and distinguished the best calculation.

2.5 MACHINE LEARNING METHODOLOGY

The central idea of any machine learning errand is to set up the model, in perspective of some estimation, to play out a particular task: gathering, clusterization, backslide, et cetera. Planning is done in perspective of the data dataset, and the model that is gathered is in this way used to make figures. The yield of such model depends upon the hidden errand and the execution. Possible applications are: given data about house characteristics, for instance, room number, size, and cost, envision the cost of the effectively cloud house; in perspective of two datasets with sound remedial pictures and the ones with tumor, orchestrate a pool of new pictures; aggregate pictures of animals to a couple of packs from an unsorted pool.

The procedure comprises of 5 phases:

1. **Data intake:** immediately, the dataset is stacked from the record and is saved in memory.

2. **Data transformation**: At this point, the data that was stacked at arrange 1 is changed, cleared, and institutionalized to be sensible for the computation. Data is changed over with the objective that it lies in a comparative range, has a comparative association, et cetera. Presently incorporate extraction and decision, which are analyzed further, are executed moreover. Despite that, the data is secluded into sets – 'planning set' and 'test set'. Data from the arrangement set is used to produce the model, which is later evaluated using the test set.

3. **Model Training**: At this stage, a model is developed using the picked figuring.

4. **Model Testing**: The model that was produced or arranged in the midst of stage 3 is had a go at using the test educational list, and the conveyed result is used for building another model, that would consider past models, i.e. "learn" from them.

5. **Model Deployment**: At this stage, the best model is picked (either after the described number of accentuation or when the required result is refined).

CHAPTER 3

PROBLEM FORMULATION

This section presents the center of the issue proclamation that we are attempting to address in this undertaking. It speaks finally about the arrangement of the issue and related ideas.

3.1 PROBLEM STATEMENT

Applications introduced on cell phones ask for access to the delicate data which may prompt security vulnerabilities. Diverse malware named as Botnet, Backdoor, Rootkits, Virus, Worms, and Trojans can assault Android Operating System (OS). Because of these assaults security of the clients is imperiled.

It's vital for antimalware programming to utilize marks just to keep a considerable measure of the shoddy refuse out. In any case, increasingly malware originates from refined designers that avoid signature discovery. All great antimalware programming nowadays should utilize a type of heuristic calculations. Great heuristics can counteract zero-day assaults. One promising kind of heuristic innovation is machine learning malware examination .

Machine learning is a prevalent way to deal with signatureless malware discovery since it can sum up to at no other time seen malware families and polymorphic strains. This has brought about its reasonable use for either essential location motors or supplementary heuristic identifications by hostile to malware sellers.

The fundamental explanation for utilizing machine-learning strategies is that they empower recognition of a formerly obscure danger utilizing models gained from known malware.

The issue explanation can be figured as -

- Due to the tremendous measure of utilization being produced and circulated each day, Android needs malware examination methods that are not quite the same as some other working framework.

- A hearty machine learning based malware identification method is a key necessity in the present digital world, particularly in the Android space, where a great many new malware are accounted for consistently, and countless new malware are muddled from existing malware.
- The machine learning method utilized should attempt to settle on choices about regardless of whether investigated code is hurtful in view of a progression of attributes. A few qualities may rank higher than different characteristics. So code that is resolved to be kindhearted may have a few attributes that the product considers to be a conceivable sign of malware. Malware is advancing quickly, so the calculations must advance quickly also. It's a consistent, progressing process.

3.2 PROBLEM SOLUTION

This task proposes a reasonable and about ideal security arrangement by utilizing machine learning approaches particularly centered around directed procedures. Our point is to give the best till date answer for malware location in Android OS. Since reusability of code and simple dispersion of utilization permits malware creators to effectively keep away from signature-based location, clients are just shielded from malware that is identified by most as of late refreshed marks however not shielded from new malware (i.e. zero-day assault).

Along these lines, we will likely discover an answer that can procedure an application, remove highlights and attempt to anticipate whether the application under process might be Malware or Benign. The reason for this work is to decide the best component extraction, include portrayal, and grouping techniques that outcome in the best precision.

This work presents endorsed strategies for machine learning based malware portrayal and acknowledgment, and furthermore the principles for its execution. Also, the examination performed can be significant as a base for moreover investigate in the field of malware examination with machine learning procedures.

CHAPTER - 4

PROPOSED WORK

The conventional strategy which distinguishes the malware in view of the mark can't recognize obscure applications. Along these lines, in this paper, we show a machine learning-based framework for the discovery of malware on Android gadgets.

Machine learning is a method that enables PCs to take in and enhance from their past encounters without being expressly modified. As such, we can envision Machine Learning as an understudy who gains from their earlier slip-ups. Machine learning centers around the advancement of PC programs that can change when presented to new information. The procedure of machine learning scans through information to search for designs..

4.1 PROPOSED SYSTEM

In this venture, we propose an Intelligent Malware Detection System for Android OS. The system utilized is called "Various Ensemble Technique for Android Devices". The Malware identifier comprises of two subcomponents that are :

1) Feature Extractor
2) Malware Classifier

The proposed technique has predominantly three phases. Right off the bat, the consent fields are separated from the android show record of the applications. Second, a database of the considerable number of authorizations for both typical and malware information is built up lastly the machine learning calculations are utilized to group and recognize the malware in Android applications.

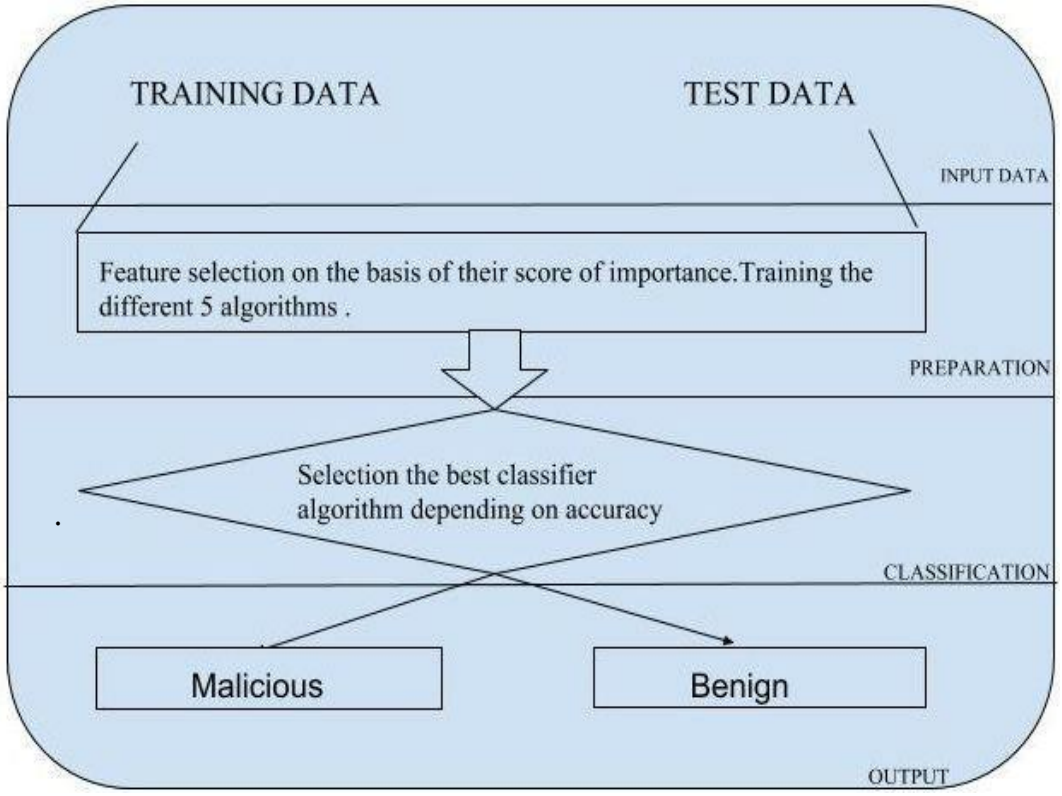

Fig 4.1 Architecture of proposed model

4.1.1 Feature Extractor

The significant capacity of the element extractor is to remove finish executables that compare to identify covered up and questionable procedures. Once the covered up and questionable executables are extricated, the Malware Detector and Malware Classifier research them independently to determine any vindictive substance is available in the recreated executables.

Android applications are pressed into apk design, and the highlights we are occupied with are scrambled, for example, consents, APIs, activities, IP, and URLs. To separate these highlights, we execute the decoder in light of an open source recompilation apparatus, which unloads applications to meaningful XML or little records. In any case, applications regularly contain heaps of APIs, and we just need the highlights comparing to their fundamental capacities. Along these lines, we characterize a few tweaked extraction rules (likewise arrangements) in XML records.

The Android application dataset contains both malware and kindhearted applications, and Feature Extractor bunches unraveling applications to decoded records removes cause highlights from these documents into the element dataset.

In include extraction, we utilize the decoder to dismember Android applications and concentrate every one of the highlights which compare to their capacities and practices. The extricated highlights can be isolated into four distinct classes.

- Permission Features
- API Features:
- Action Features
- IP And URL Features

4.1.2 Malware Classifier

In this venture, five classifier calculations are utilized to group the dataset. Out of which, one is progressively chosen and spared in the classifier index. These classifier calculations are Gaussian Naive Bayes, Random woodland, Decision tree, Adaboost, Gradient Boosting. Every one of these calculations will keep running on the given dataset utilizing the sci-pack learn library. Precision will be computed for each calculation utilizing the standard recipe. Later on, contingent upon the estimation of precision, the best calculation will be spared.

4.2 FEATURE EXTRACTION TECHNIQUE :

The machine learning-based systems by and large require a component vector portrayal of malware, where an element speaks to a specific malware property that can assume a biased part in the order procedure. Removing prejudicial credits relating to malware and speaking to them in an approach to be successfully utilized as a part of a machine picking up setting is a noteworthy test in the space of malware investigation and recognition. Named as highlight extraction in the machine learning worldview, this procedure is an essential to each malware identification method proposed in the writing that utilizes a machine learning calculation.

The executables are utilized as information records in the initial step of highlight

extraction to separate the hexadecimal dump, and after that pre-process the hexadecimal dump to evacuate any unessential data. After the pre-preparing activity, just the byte groupings that speak to a bit of the machine code of the executable got.

Another critical prerequisite for a not too bad list of capabilities is non-repetition. Having repetitive highlights i.e. highlights that framework a similar data, and in addition repetitive data characteristics, that are firmly reliant on each other, can make the calculation one-sided and, consequently, give an off base outcome. Notwithstanding that, if the info information is too enormous to be sustained into the calculation (has excessively numerous highlights), at that point it can be changed to a diminished element (vector, having fewer highlights). The way toward diminishing the vector measurements is alluded to as highlight choice.

4.3 FEATURE SELECTION USING SelectFromModel :

Select From Model is a meta-transformer that can be utilized alongside any estimator that has a coef_ or feature_importances_ trait in the wake of fitting. The highlights are viewed as insignificant and expelled if the correspondingcoef_ or feature_importances_ esteems are underneath the given limit parameter. Aside from determining the edge numerically, there are worked in heuristics for finding 0an edge utilizing a string contention. Accessible heuristics are "signify", "middle" and buoy products of these like "0.1*mean".

4.4 CLASSIFICATION METHODS

In this section, the theoretical establishment is given on each one of the methodologies used as a piece of this wander.

4.4.1 K-nearest neighbors K-Nearest Neighbors (KNN) is one of the slightest complex, be that as it may, correct machine learning figurings. KNN is a non-parametric computation, inferring that it doesn't make any assumptions about the data structure. In authentic issues, data on occasion conforms to the general theoretical suppositions, making non-parametric counts a not too bad response for such issues. KNN indicate depiction is as fundamental as the dataset – there is no

learning required, the entire getting ready set is secured. KNN can be used for both portrayal and backslide issues.

k nearest neighbors. In the backslide issue, the yield would be the property estimation, which is all things considered a mean estimation of the *k* closest neighbors.

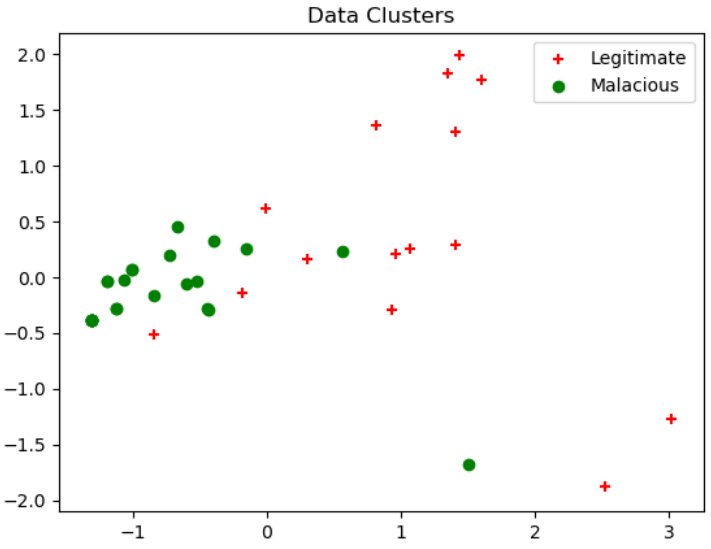

Fig 4.2 : KNN

4.4.2 Support Vector Machines: Support Vector Machines (SVM) is another machine learning count that is all around used for course of action issues. The standard idea relies upon finding such a hyperplane, that would seclude the classes in the best way. The term 'support vectors' insinuates the concentrations lying closest to the hyperplane, that would change the hyperplane position if cleared..

4.4.3 Naive Bayes: Naive Bayes is the game plan machine learning count that relies upon the Bayes Theorem. It can be used for both twofold and multi-class portrayal issues. The essential point relies upon treating every segment openly. Unsophisticated Bayes system surveys the probability of every component self-governingly, paying little notice to any associations, and makes the figure in perspective of the Bayes Theorem. That is the reason this strategy is called "simple" – in authentic issues incorporates every now and again have some level of connection between's each other.

4.4.4 Decision Tree: decision trees are data structures that have a structure of the tree. In this figuring, the goal is to achieve the most exact result with insignificant

number of the decisions that must be made. Decision trees can be used for both course of action and backslide issues.

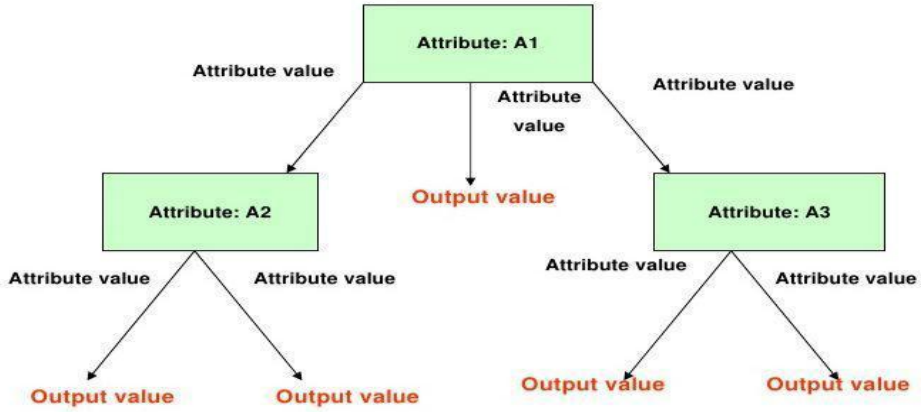

Fig 4.3: Decision tree

4.4.5 Random Forest: Random Forest is a regulated learning calculation. It makes a timberland and makes it some way or another arbitrary. The "woodland" it fabricates, is an outfit of Decision Trees, more often than not prepared with the "sacking" technique. The general thought of the packing strategy is that a blend of learning models builds the general outcome. One major preferred standpoint of irregular backwoods is, that it can be utilized for both arrangement and relapse issues, which shape the lion's share of current machine learning frameworks

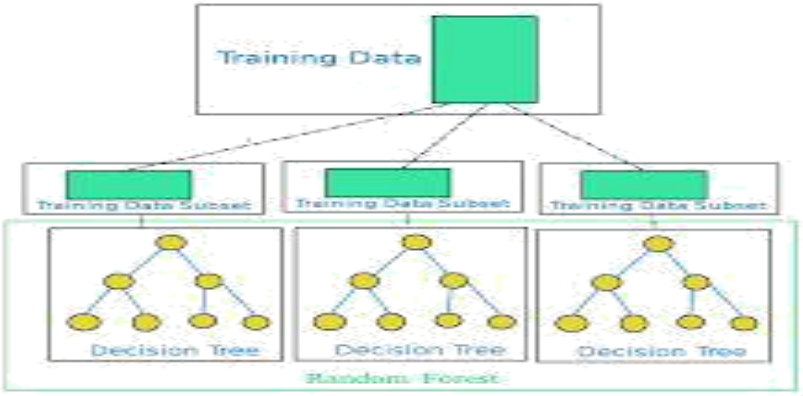

Fig 4.4 : Random forest

Algorithm:-

1. Multiple trees are fabricated generally on the two third of the preparation information. Information is picked haphazardly.

2. Several indicator factors are arbitrarily chosen out of all the indicator factors. At that point, the best split on these chose factors is utilized to part the hub. As per normal procedure, the measure of the picked factors is the square establishment of the total number of all markers for plan, and it is relentless for all trees.

3. Using whatever remains of the information, the misclassification rate is computed. The aggregate blunder rate is computed as the generally out-of-sack mistake rate.

3. Each prepared tree gives its own particular characterization result, giving its own exactness. The class that got the most astounding precision is picked as the outcome.

As in the choice trees, this calculation evacuates the requirement for include choice for expelling unessential highlights – they won't be considered regardless. The fundamental prerequisite for any component assurance with the discretionary woodlands figurings rises when there is a necessity for dimensionality diminishment. Moreover, the out-of-pack botch rate, which was indicated earlier can be seen as the count's own specific cross-endorsement method. This clears the prerequisite for dull cross-endorsement measures, that would should be taken something unique.

CHAPTER 5

CONCEPT OUTLINE

The section talks about the fundamental ideas used in the plan of the proposed approach.

5.1 DECISION TREES:- Decision Trees are a sort of Supervised Machine Learning .

Choice Tree Algorithm Pseudocode:-

- Place the best property of the dataset at the base of the tree.
- Split the preparation set into subsets.
- Subsets ought to be made such that every subset contains information with a similar incentive for a characteristic.
- Repeat stage 1 and stage 2 on every subset until the point that you discover leaf hubs in every one of the branches of the tree.

Advantages:

- Decision Trees are anything but difficult to clarify.
- It brings about an arrangement of standards.
- It takes after an indistinguishable approach from people for the most part take after while deciding.
- Interpretation of an unpredictable Decision Tree model can be streamlined by its representations.
- Even an innocent individual can comprehend rationale.

Disadvantages:

- Information pick up in a choice tree with downright factors gives a one-sided reaction for traits with more prominent no. of classes.
- Calculations can wind up complex when there are numerous class names.

5.2 RANDOM FOREST:- Random Forest is a regulated learning calculation. It

makes a timberland and makes it some way or another arbitrary. The "woodland" it fabricates, is an outfit of Decision Trees, more often than not prepared with the "sacking" technique. The general thought of the packing strategy is that a blend of learning models builds the general outcome. One major preferred standpoint of irregular backwoods is, that it can be utilized for both arrangement and relapse issues, which shape the lion's share of current machine learning frameworks.

Advantage:

- A single choice tree tends to overfit the information. The way toward averaging or consolidating the aftereffects of various choice trees beats the issue of overfitting.
- Random timberlands likewise have less fluctuation than a solitary choice tree. It implies that it works accurately for a substantial scope of information things than single choice trees.

- Random timberlands are to a great degree adaptable and have high exactness.
- They additionally don't require planning of the information. You don't need to scale the information.
- It additionally keeps up exactness notwithstanding when a vast extent of the information are absent.

Disadvantages:

- The principle disservice of Random backwoods is their multifaceted nature.
- They are significantly harder and tedious to develop than the choice trees.
- They likewise require more computational assets and are additionally less natural.
- When you have a vast gathering of choice trees it is difficult to have a natural handle of the relationship existing in the information.
- In expansion, the expectation procedure utilizing arbitrary timberlands is tedious than different calculations.

5.3 GRADIENT BOOSTING:- Inclination boosting is a machine learning

methodology for backslide and course of action issues. That makes a desire show as a gathering of feeble conjecture models. The accuracy of a judicious model can be bolstered in two distinctive ways: Either by grasping component designing or by applying boosting calculations straight away.

Advantages:-

- Compared with to the "standard thing" boosting, GBM has the accentuation on the effect (y-) amid the development of the relapse trees with the abnormality misfortune work.
- Lots of adaptability with the decision of misfortune capacities, versatile to the qualities of the considered issues.
- GBM has demonstrated its adequacy in a few difficulties.

Disadvantages:-

- Non-unequivocal model (concerning all group techniques)
- M any parameters which can associate and impact vigorously the Behavior of the approach (number of emphasess, regularization parameters and so forth.)
- Over fitting can happen if estimations of parameters are not reasonable
- Computationally serious (particularly when the quantity of trees is high)
- Memory control of the trees.

5.4 ADABOOST ALGORITHM:- AdaBoost is a sort of "Troupe Learning" where different students are utilized to construct a more grounded learning calculation. AdaBoost works by picking a base calculation (e.g. choice trees) and iteratively enhancing it by representing the inaccurately characterized cases in the preparation set. We dole out equivalent weights to all the preparation illustrations and pick a base calculation. At each progression of the emphasis, we apply the base calculation to the preparation set and increment the weights of the inaccurately arranged cases. We emphasize n times, each time applying base student on the preparation set with refreshed weights. The last model is the weighted whole of the n students.

Advantages:-

- Very easy to execute
- Does include determination bringing about a generally basic classifier
- Fairly great speculation (suited for any grouping issue)
- Not inclined to over fitting

Disadvantages:-

- Suboptimal arrangement
- Sensitive to uproarious information and anomalies

5.5 DEPENDENCIES :

- pandas pip introduce pandas
- number pip introduce numpy
- pickle pip introduce pickle
- scipy pip introduce content
- scikit pip introduce - U scikit-learn

CHAPTER 6
IMPLEMENTATION

This part manages the execution of the venture. To effectively recognize malware and generous records, we have utilized machine learning calculations.
The entire usage process can be laid out in the accompanying advances:

1. Obtaining Dataset (utilizing open source)
2. Feature Extraction (utilizing Python 2.7)
3. Feature Selection
4. Application of machine learning techniques for malware discovery and arrangement
5. Evaluation and constant indication of results utilizing diagrams

6.1 OBTAINING DATA SET

Dataset tests of malevolent and favorable apk records were gotten from Kaggle Dataset archive. Aggregate of around 500 documents was taken from the vault. The dataset can likewise be expanded apparently.

6.2 FEATURE EXTRACTION

It includes diminishing the quantity of assets required to portray an extensive arrangement of information. Highlight extraction begins from an underlying arrangement of estimated information and fabricates determined qualities (include) expected to be useful and non-repetitive, encouraging the ensuing learning and speculation steps. Highlights in Android malware examination are the different authorizations looked for by an application. A few or huge numbers of the authorizations might be mistaken and can confer threatening outcomes on the gadget and client information uprightness. Additional tree Classifier has been utilized for include extraction in the venture.

Extra-Tree Classifier: This This class actualizes a meta estimator that fits various randomized choice trees (a.k.a. additional trees) on different sub-tests of the dataset and utilizations averaging to enhance the prescient exactness and control over-fitting.

6.3 FEATURE SELECTION

Our model of highlight determination lessens excess and immaterial highlights to enhance the exactness of the forecast. For our situation, the list of capabilities (acquired from include extraction process) is greatly huge, and the requirement for highlight choice is, along these lines, high.

In our venture, highlight choice procedures have been foreign specifically from scikit-learn library. Depiction has been displayed underneath to indicate how it is done .

6.4APPLICATION OF MACHINE LEARNING METHODS

In this venture, five diverse classifier calculations which are expressed beneath have been utilized, out of which one will be chosen based on assessed precision for the dataset under testing. We have utilized the Decision tree, Random woodland, GBN (Gaussian Naive Bayes), AdaBoost and Gradient boosting calculations.

In the wake of computing the precision of every calculation for the assigned dataset, the program thinks about and chooses the best calculation. A representation has been appeared underneath :

First of all, algorithms are imported as shown below :

```
#Algorithm comparison
algorithms = {
        "DecisionTree": tree.DecisionTreeClassifier(max_depth=10),
        "RandomForest": ske.RandomForestClassifier(n_estimators=50),
        "GradientBoosting": ske.GradientBoostingClassifier(n_estimators=50),
        "AdaBoost": ske.AdaBoostClassifier(n_estimators=100),
        "GNB": GaussianNB()
    }
```

Then, calculation of accuracy of each algorithm is as :

```
results = {}
print("\nNow testing algorithms")
for algo in algorithms:
    clf = algorithms[algo]
    clf.fit(X_train, y_train)
    score = clf.score(X_test, y_test)
    print("%s : %f %%" % (algo, score*100))
    results[algo] = score
```

After calculating the accuracy of each algorithm for the designated dataset, the program compares and selects the best algorithm. An illustration has been shown below :

```
winner = max(results, key=results.get)
print('\nWinner algorithm is %s with a %f %% success' % (winner, results[winner]*100))

print('Saving algorithm and feature list in classifier directory...')
joblib.dump(algorithms[winner], 'classifier/classifier.pkl')
open('classifier/features.pkl', 'w').write(pickle.dumps(features))
print('Saved')
```

After this, the best algorithm is saved in classifier directory for further testing the files as malicious or benign.

CHAPTER 7

RESULT ANALYSIS

This part manages the outcomes delivered an examination of that outcome utilizing different charts and figures.

7.1 PERFORMANCE METRICS CALCULATION FOR DIFFERENT ALGORITHMS AT DIFFERENT TRAINING-TESTING RATIO :

Computing a disarray lattice gives a superior thought of what the arrangement demonstrate is getting right and what kinds of mistakes it is making. Count of perplexity network has been talked about before in this report.

Exactness is ascertained for each calculation in the wake of preparing part. A calculation with the best exactness will be spared and utilized for encourage discovery of records as considerate or pernicious.

Precision is ascertained as No. of right forecasts/no. of aggregate expectations

In terms of TP, TN, FP, FN,

Accuracy ={TP+TN}/{TP+TN+FP+FN}

F1 Score: The F-1 score is the consonant normal of the accuracy and review, where a F-1 score achieves its best an incentive at 1 (idealize exactness and review) and most exceedingly awful at 0.

7.1.1 AT THE RATIO OF 70:30 (TRAINING-TESTING RATION):

The accuracy of different algorithms :

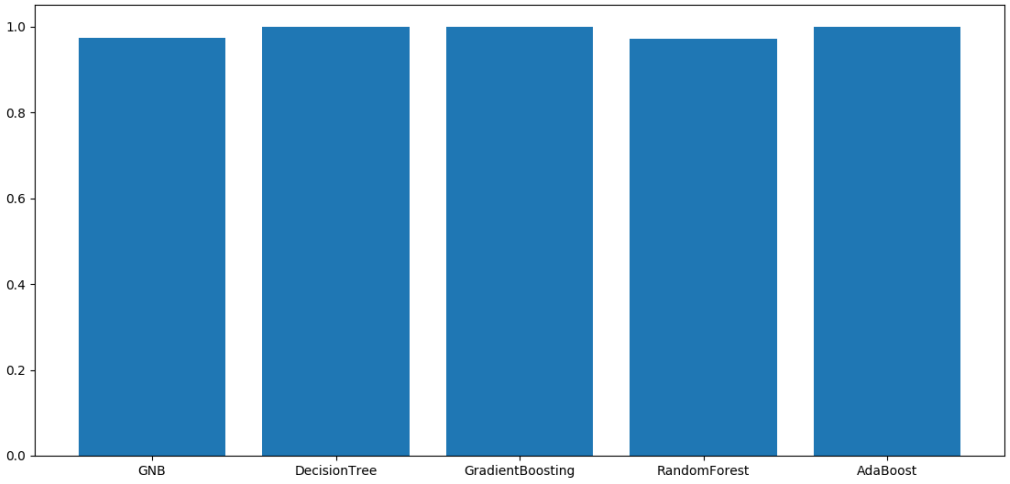

Fig 7.1 Accuracy for different algorithms

Out of five calculations, the program chooses best of the five progressively and details its perplexity network as given below :

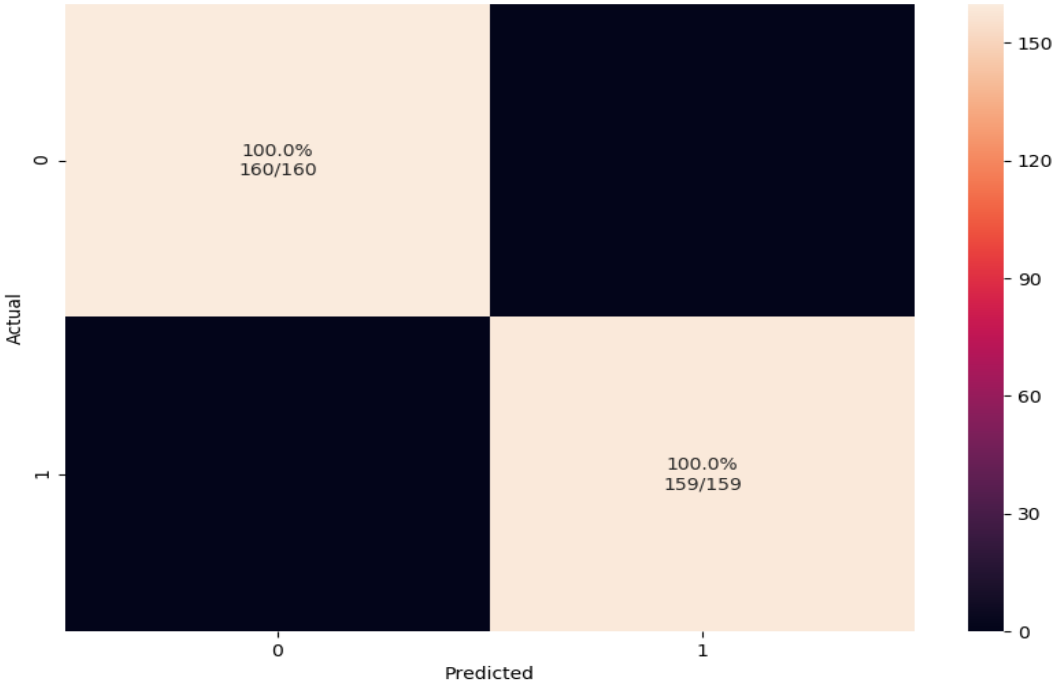

Fig. 7.2 Confusion matrix for best algorithm

F1 score, FPR, and FNR of the best algorithm are shown below as :

```
Best algorithm is DecisionTree with a 100.000000 % success
F1 Score is  : 100.000000 %
Confusion Matrix
Predicted  False  True  __all__
Actual
False         63     0       63
True           0    57       57
__all__       63    57      120
False positive rate : 0.000000 %
False negative rate : 0.000000 %
```

Fig. 7.3 F1-score, FPR, and FNR

7.1.2 AT THE RATIO OF 20:80 (TRAINING-TESTING RATIO):

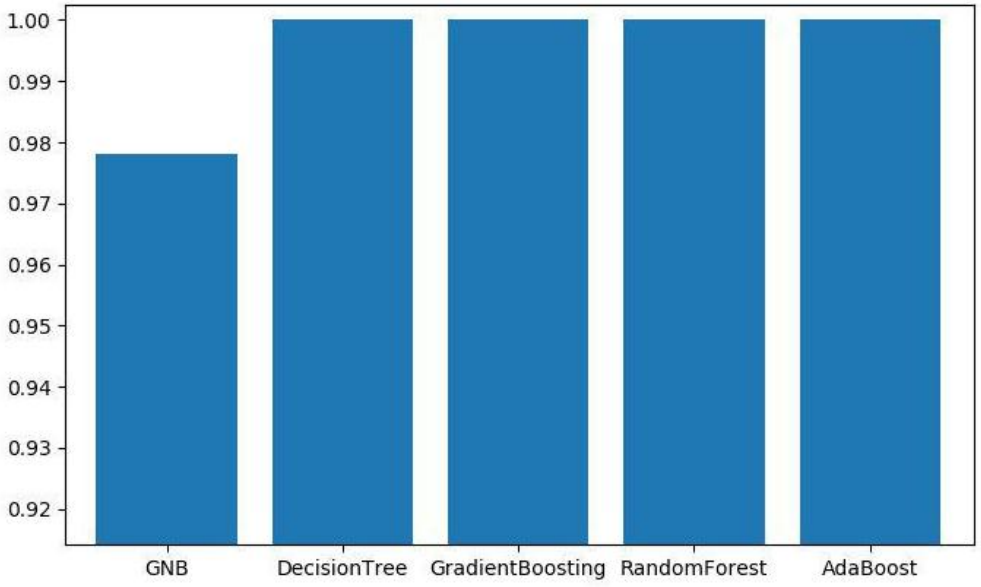

Fig. 7.4 Accuracy for different algorithms

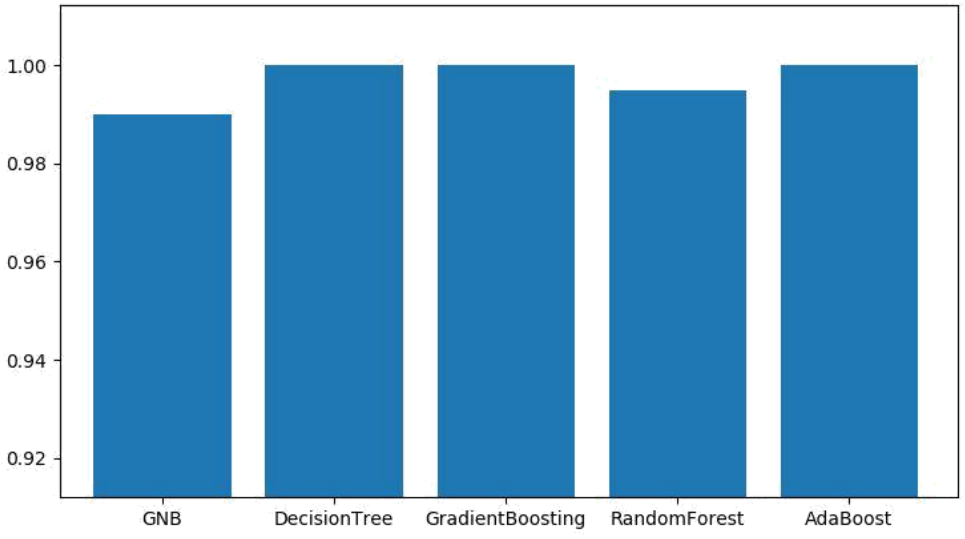

Fig. 7.5 Confusion matrix for best algorithm

F1 score, FPR, and FNR of the best algorithm are shown below as :

```
Best algorithm is DecisionTree with a 100.000000 % success
F1 Score is  : 100.000000 %
Confusion Matrix
Predicted  False  True  __all__
Actual
False        156     0      156
True           0   163      163
__all__      156   163      319
False positive rate : 0.000000 %
False negative rate : 0.000000 %
```

Fig. 7.6 F1-score, FPR, and FNR

7.1.3 AT THE RATIO OF 50:50 (TRAINING-TESTING RATIO):

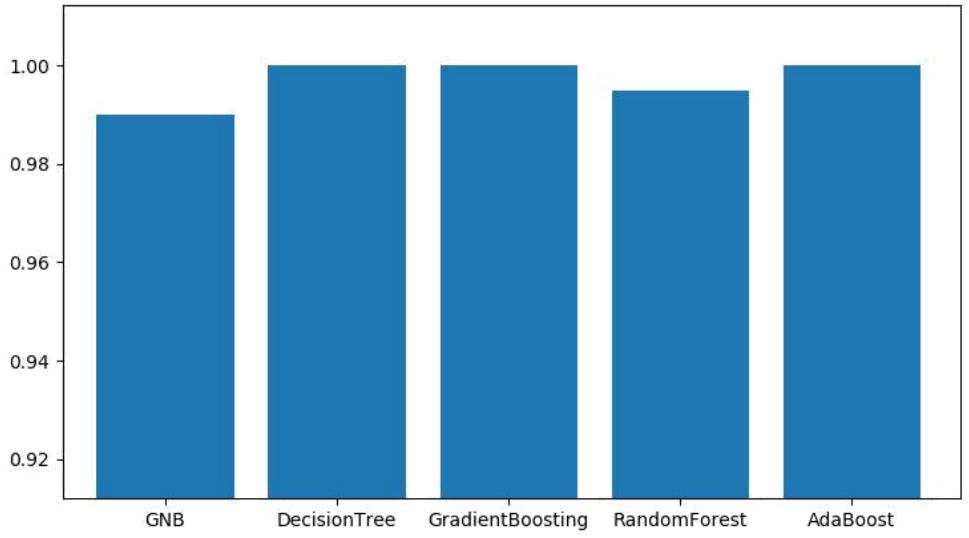

Fig. 7.7 Accuracy of the different algorithm

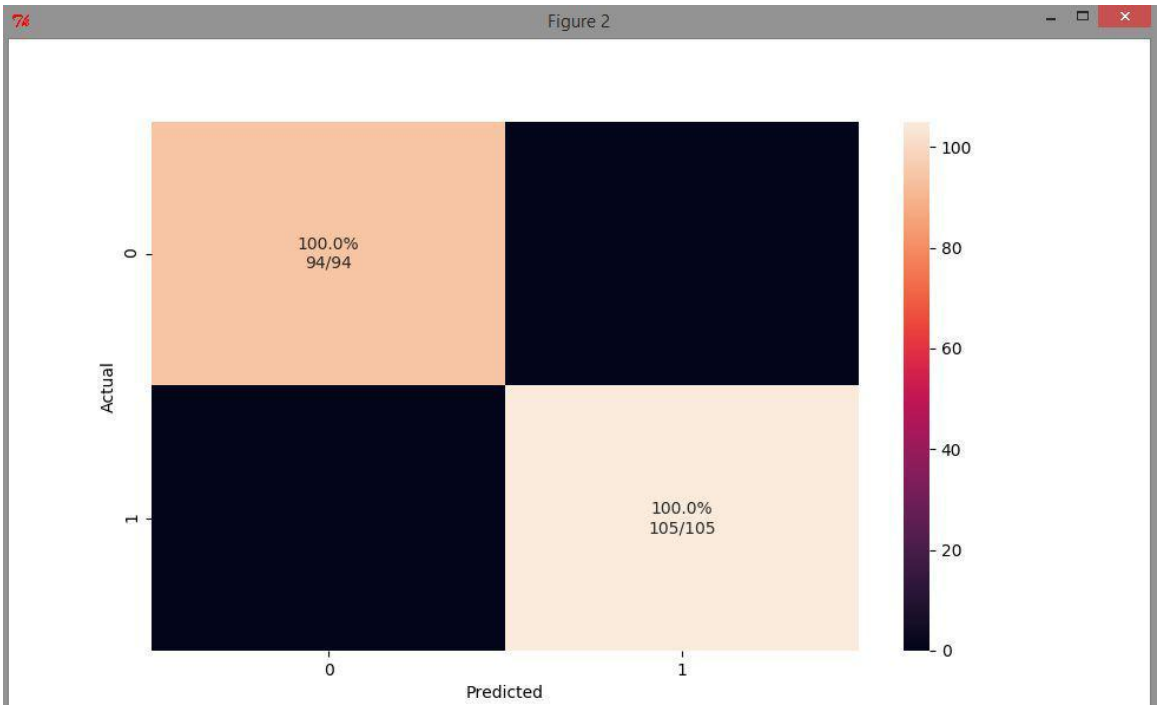

Fig. 7.8 Confusion matrix for best algorithm

F1 score, FPR, and FNR of the best algorithm are shown below as :

```
Best algorithm is DecisionTree with a 100.000000 % success
F1 Score is  : 100.000000 %
Confusion Matrix
Predicted  False  True  __all__
Actual
False         99     0       99
True           0   100      100
__all__       99   100      199
False positive rate : 0.000000 %
False negative rate : 0.000000 %
```

Fig. 7.9 F1-score, FPR, and FNR

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

This part clarifies the conclusion and conceivable work after this undertaking.

8.1 CONCLUSION

By and large, the objectives characterized for this investigation were accomplished. The coveted component extraction and portrayal strategies were chosen and the chose machine learning calculations were connected and assessed. The coveted element portrayal technique was chosen to be the joined network, illustrating the recurrence of fruitful and fizzled API calls alongside the arrival codes for them. This was picked in light of the fact that it diagrams the real conduct of the record.. In grouping issues, distinctive models gave diverse outcomes. The most minimal exactness was accomplished by Naive Bayes (70.15%) took after by AdaBoost (98.48%). The most elevated precision was accomplished with the Random Forest models, and it was equivalent to 100%.

In light of the outcomes portrayed previously, it is prescribed to actualize the grouping in light of the Random Forest strategy for arrangement, as it brought about the best exactness and elite.

8.2 FUTURE SCOPE

1. **Use wider dataset:** A Although the dataset that was utilized as a part of this examination is expansive, covering the vast majority of the malware writes that are pertinent to the cutting edge world, it does not cover every conceivable kind. Despite that, understand that the model may have the ability to envision the cases of the families that it has seen previously. In a manner of speaking, in a genuine application, the best measure of possible families should be used before was made by the necessity for picking the critical API calls and emptying the overabundance ones. For help execution, simply the APIs that were perceived as appropriate in this examination can be used. This will reduce

the measure of time required for data preprocessing, decrease the execution essentials of the machine on which the anathe dispatch of the assignment for authentic circumstances.

2. **Use pre-selected APIs:** In this work, the immense overhead in the data taking care of lysis is being done and decrease the level of feature decision to be made. Regardless, it should be seen, that for more exact depiction, the imperative APIs should be expelled from the best possible dataset. Moreover, it is urged to pick the germane APIs per malware family, as this will realize another level of versatility and accuracy.

3. GUI Implementation: Graphic UI can likewise be additionally produced for this venture which was absent in the task,

REFERENCES

[1] Source: **G Data Security Blog.**

[2] **J. Quinlan**, Induction of decision trees. Machine learning, vol. 1, no. 1, pp. 81106, 1986

[3] **A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss**, an anomaly: A behavioral malware detection framework for Android devices J. Intell. Inf. Syst., 38(1):161190, 2012.

[4] **J. Kephart and W. Arnold**, Automatic extraction of computer virus signatures In Proceedings of 4th Virus Bulletin International Conference, pages 178 184, 1994

[5]**Android-Apktool,**A tool for reverse engineering Android apk files https://code.google.com/p/android-apktool/

[6]**O'Brien, Dick.**2016. Dridex: Tidal waves of spam pushing dangerous financial Trojan. *Symantec Corporation.* Pirscoveanu, Radu-Stefan. 2015. Clustering Analysis of Malware Behavior *Aalborg University.*

[7] **Prasad, B. Jaya, Haritha Annangi, and Krishna Sastry Pendyala**.2016. Basic static malware analysis using open source tools. Reddy, Krishna Sandeep, and Arun Pujari. 2006.

[8] **N-gram analysis for computer virus detection. Rieck, Konrad, Philipp Trinius, Carsten Willems, and Thorsten Holz**.2011. Automatic Analysis of Malware Behavior using Machine Learning. *Journal of Computer Security.*

[9]**Symantec Security Response**.2016. Locky ransomware on an aggressive hunt for victims. WWW document. Available at: https://www.symantec.com/connect/blogs/locky-ransomware-aggressive-huntvictims[Accessed 15 February 2017] Thirumuruganathan, Saravanan. 2010.

[10]**A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm**. WWW document. Available at: https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailedintroduction-to-k-nearest-neighbor-knn-algorithm/. [Accessed 15 February 2017]

[11] **Venables, W. N., and D. M. Smith**.2016. An Introduction to R. Villeneuve, Nart, and James T. Bennett. 2014.

[12] **XtremeRAT: Nuisance or Threat?**WWW document. Available at: https://www.fireeye.com/blog/threatresearch/2014/02/xtremerat-nuisance-or-threa t.html. [Accessed 15 February 2017] VirusTotal. 2017.

[13]**VirusTotal**.WWW page. Available at: https://virustotal.com/. [Accessed 15 February 2017]

[14] **O. Tripp, M. Pistoia, S.J. Fink, M. Sridharan, and O. Weisman. Taj**: effective taint analysis of web applications. In *ACM Sigplan Notices*, volume 44, pages 87–97. ACM, 2009.

[15] **Sara Yin.**‘most sophisticated' android trojan surfaces in China. December 2010. Accessed March 18, 2011. http://www.pcmag.com/article2/0,2817,2374926,00.asp.

[16]**Aliyev, Vusal.** 2010. Using honeypots to study skill level of attackers based on the exploited vulnerabilities in the network. The Chalmers University of Technology. Aquino, Maharlito. 2014.

[17] **Fake BACS Remittance Emails Delivers Dridex Malware.**WWW document. Available at: https://blog.cyren.com/articles/fakebacs-remittance-emails-delivers-dridex-malwa re.html. [Accessed 15 February 2017].

[18] **Kai Zhao, Dafang Zhang, Xin Su, Wenjia Li**. "Fest: A feature extraction and selection tool for Android malware detection" , *2015 IEEE Symposium on Computers and Communication (ISCC), 2015*