

Malware Detection on Android

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

B.K.N. VARDHAN

(Reg. No. 37110313)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **B.K.N. VARDHAN** (Reg. No.37110313) who carried out the project entitled "**MALWARE DETECTION ON ANDROID**" under my supervision from **August 2020** to **March 2021**.

Internal Guide

Dr.J. Refonaa M.E., Ph.D.,

Head of the Department

Dr.L.Lakshmanan M.E., Ph.D
Dr. S. VIGNESHWARI, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KANISSETTY SAI SARAN** and **B.K.N. VARDHAN** hereby declare that the Project Report entitled "**MALWARE DETECTION ON ANDROID**" is done by me under the guidance of **Dr.J. Refonaa. M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J. Refonaa M.E, Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Resource-constrained systems are becoming more and more common as users migrate from PCs to mobile devices. In this project we introduce, a mobile software leveraging machine learning for detecting malware on resource constrained devices. This software executes in background and transparently analyzes the applications as soon as they are installed, i.e., before infecting the device. Permission-based security model of Android restricts applications to access specific resources, but malicious applications can invade more easily in such user-centric pattern. Through the analysis of the Android Permission-based security model and the permission features of Android applications, we establish the permission model to quantify the functional characteristics of the application, and then provide an assessment method in which we use the network visualization techniques and clustering algorithm to determine whether the testing application is potentially malicious application or not so as to help users choose applications before installation. Security testing is known to be a notoriously difficult activity. This is partly because unlike functional testing that aims to show a software system complies with its specification, security testing is a form of negative testing, i.e., showing that. A certain behavior does not exist in the system.

TABLE OF CONTENTS

	ABSTRACT	v
CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 INTROCUCTION	1
	1.2 SCOPE OF THE PROJECT	1
	1.3 DOMAIN OVERVIEW	1
	1.3.1 MACHINE LEARNING	1
2.	LITERATURE SURVEY	6
3.	SYSTEM ANALYSIS	8
	3.1 EXISTING SYSTEM	13
	3.2 DRAWBACKS FOR EXISTING SYSTEM	13
	3.3 PROPOSED SYSTEM	13
	3.4 ADVANTAGES OF PROPOSED SYSTEM	14
4.	REQUIRES SPECIFICATION	15
	4.1 INTRODUCTION	15
	4.2 HARDWARE & SOFTWARE SPECIFICATION	15
	4.3 SOFTWARE DESCRIPTION	16
	4.3.1 ABOUT PYTHON	16
	4.3.2 ABOUT IDE	25
5.	SYSTEM DESIGN	29
	5.1 INTRODUCTION	29
	5.2 ARCHITECTURE DIAGRAM	29

6.	SYSTEM IMPLEMENTATTION	31
	6.1 LIST OF MODULES	31
	6.1.1 MODULE 1: EXPLORATORY DATA EVALUATION	31
	6.1.2 MODULE 2: PRE- PROCESSING	31
	6.1.3 MODULE 3: FEATURE ENGINEERING	32
	6.1.4 MODULE 4: PREDICTION	33
7.	SOFTWARE TESTING	34
	7.1 GENERAL	38
	7.2 TEST DRIVEN DEVELOPMENT	39
	7.3 UNIT TESTING	40
	7.4 BLACKBOX TESTING	41
	7.5 INTEGRATION TESTING	41
	7.6 SYSTEM TESTING	42
	7.7 SANITY TESTING	44
	7.8 REGRESSION TESTING	45
8.	RESULT & ANALYSIS	49
9.	CONCLUSION AND FUTURE ENHANCEMENT	52
	REFERENCES	53
	APPENDIX	
	A. PLAGIARISM REPORT	56
	B. JOURNAL PAPER	60
	C. SOURCE CODE	64

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.3.1.1	CLASSIFICATION	3
1.3.1.2	SUPERVISED LEARNING	4
3.1	WATERFALL MODEL	9
3.2	RAD MODEL	10
4.3.1	PYTHON	16
4.3.1-2	PYTHON IMPLEMENTATING MACHINE LEARNING	17
5.1.1	ARCHITECTURE DIAGRAM	30
7.1	IMPLEMENTATION	37
7.2	BENEFITS OF TESTING	39
7.3	INTEGRATION TESTING	42
7.4	SYSTEM TESTING	43
7.5	PERFORMANCE TESTING	47
8.1	TRAINING & CROSS VALIDATION GRAPH	49
8.2	ANAMOLY SCORE CURVE	49
8.3	HEAT MAP	50
8.4	VARIANCE & NUMBER OF COMPONENTS CURVE	51

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The proposed system has grave consequences for device security, as any app – can still download and execute malicious files without getting detected by the user. In this paper we propose and implement a Runtime detection and prevention system.

1.2 SCOPE OF THE PROJECT

The main contributions of this project therefore are:

- Data Analysis
- Dataset preprocessing
- Training the Model
- Testing of Dataset

1.3 DOMAIN OVERVIEW

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future.

Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

1.3.1 MACHINE LEARNING

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine

learning models and finally, computer science is used for implementing algorithms.

However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyperparameters to reach an optimum result.

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry.

Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it.

These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows-

- Supervised Learning

- Unsupervised Learning

Reinforcement Learning

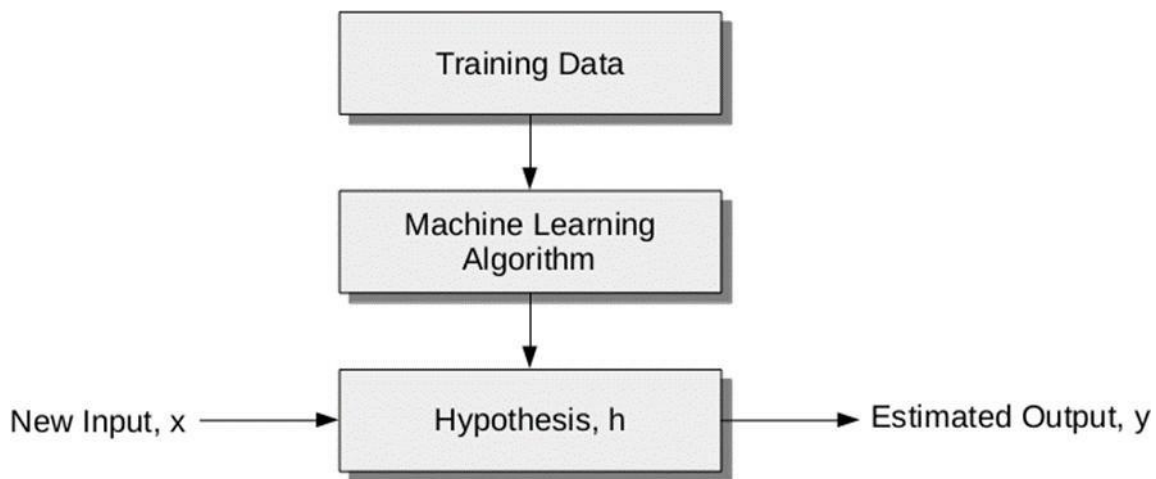


FIG 1.3.1 1 CLASSIFICATION

Supervised Learning

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function $h(x)$ (sometimes called the “hypothesis”). “Learning” consists of using sophisticated mathematical algorithms to optimize this function so that, given input data x about a certain domain (say, square footage of a house), it will accurately predict some interesting value $h(x)$ (say, market price for said house).

This function takes input in four dimensions and has a variety of polynomial terms. Deriving a normal equation for this function is a significant challenge. Many modern machine learning problems take thousands or even millions of dimensions of data to build predictions using hundreds of coefficients. Predicting how an organism’s genome will be expressed, or what the climate will be like in fifty years, are examples of such complex problems.

Under supervised ML, two major subcategories are:

Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum. These systems help us with questions of “How much?” or “How many?”.

Classification machine learning systems: Systems where we seek a yes-or-no prediction, such as “Is this tumor cancerous?”, “Does this cookie meet our quality standards?”, and so on.

In practice, x almost always represents multiple data points. So, for example, a housing price

predictor might take not only square-footage (x1) but also number of bedrooms (x2), number of bathrooms (x3), number of floors (x4), year built (x5), zip code (x6), and so forth. Determining which inputs to use is an important part of ML design. However, for the sake of explanation, it is easiest to assume a single input value is used.

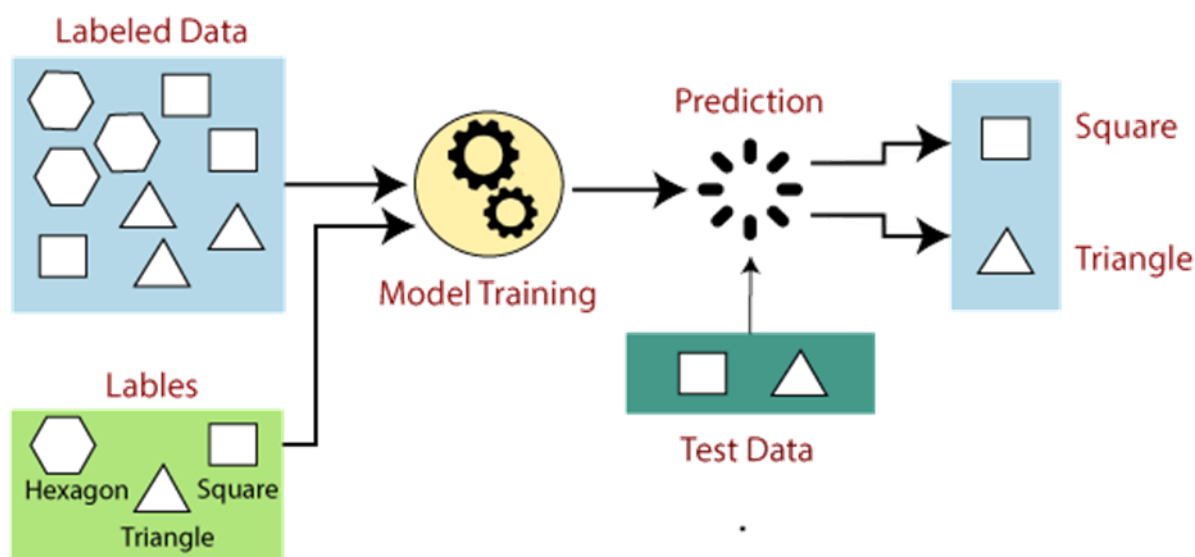


FIG 1.3.1 2 SUPERVISED LEARNING

Steps involved in Supervised Learning:

First Determine the type of training dataset

Collect/gather the labelled training dataset.

Split the training dataset into training dataset, test dataset, and validation dataset.

Determine the input features the training dataset, which should have enough knowledge so that the model can accurately predict the output.

Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.

Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.

Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Regression

Regression algorithms are used if there is a relationship between the input variable and the

output variable. It is used for the prediction of continuous variables, such as Weather

forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

1. Linear regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression

Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering

Random Forest

Decision Trees

Logistic Regression

Support Vector Machines

CHAPTER 2

LITERATURE SURVEY

Behavioural-Based Classification and Identification of Ransomware Variants Using Machine Learning

Due to the changing behavior of ransomware, traditional classification and detection techniques do not accurately detect new variants of ransomware. Attackers use polymorphic and metamorphic techniques to avoid detection of signature-based systems. We use machine learning classification to identify modified variants of ransomware based on their behavior. To conduct our study, we used behavioral reports of 150 ransomware samples from 10 different ransomware families. Our data-set includes some of the newest ransomware samples available, providing an evaluation of the classification accuracy of machine learning algorithms on the current evolving status of ransomware. An iterative approach is used to identify optimum behavioral attributes used to achieve best classification accuracy. Two main parts of this study are identification of the behavioral attributes which can be used for optimal classification accuracy and classification of ransomware using machine learning algorithms. We have evaluated classification accuracy of three machine learning classification algorithms.

Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences

Development and dissemination of malicious software requires the creation of new methods for their detection. Therefore, we began to use proactive technologies that use the test program to detect the presence of certain symptoms, often occurring in malware. Dynamic analysis of the studied program launched for execution. There is a study of how the program interacts with the software environment that is read/write at certain registry keys, files, network activity the use of certain API calls. Due to the fact that studied the program is potentially harmful, to make its execution must be in an isolated environment. This paper discusses proactive methods based on API call analysis and propose a new method using a multiple sequence alignment to identify common traits in malware. The paper considers the scheme to detect malicious software, based on API calls, each of which is implemented in software. Also presented a completely new malware detection scheme based on multiple sequence API calls alignment. This scheme is described in detail and implemented in

software. A test on a set of software and the legitimacy of the viral nature. Testing has shown that the established scheme of competitive shows and identifies malicious software with high accuracy.

The impact of malware evolution on the analysis methods and infrastructure

The huge number of malwares introduced each day demands methods and tools for their automated analyses. Complex and distributed infrastructure of malicious software and new sophisticated techniques used to obstruct the analyses are discussed in the paper based on real-life malware evolution observed for a long time. Their impact on both toolsets and methods are presented based on practical development of systems for malware analyses and new features for existing tools.

Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences

Development and dissemination of malicious software requires the creation of new methods for their detection. Therefore, we began to use proactive technologies that use the test program to detect the presence of certain symptoms, often occurring in malware. Dynamic analysis of the studied program launched for execution. There is a study of how the program interacts with the software environment that is read/write at certain registry keys, files, network activity the use of certain API calls. Due to the fact that studied the program is potentially harmful, to make its execution must be in an isolated environment. This paper discusses proactive methods based on API call analysis and propose a new method using a multiple sequence alignment to identify common traits in malware. The paper considers the scheme to detect malicious software, based on API calls, each of which is implemented in software. Also presented a completely new malware detection scheme based on multiple sequence API calls alignment. This scheme is described in detail and implemented in software. A test on a set of software and the legitimacy of the viral nature. Testing has shown that the established scheme of competitive shows and identifies malicious software with high accuracy.

CHAPTER 3

SYSTEM ANALYSIS

One of the primary choices we face for our project implementations is “Which development methodology should we use?” There are various software development methodologies available in the ICT industry. They are

Agile Software Development

Crystal Methods

Dynamic Systems Development Model (DSDM)

Extreme Programming (XP)

Feature Driven Development (FDD)

Joint Application Development (JAD)

Lean Development (LD)

Rapid Application Development (RAD)

Rational Unified Process (RUP)

Scrum

Spiral

Systems Development Life Cycle (SDLC)

Waterfall (a.k.a. Traditional)

Among all these methodologies, we have decided to compare the two most popular and widely used methodologies:

Waterfall Model

RAD Model

Waterfall Model

It is called as traditional approach. Waterfall is a linear method (sequential model) for any software application development. Here application development is segregated into a

sequence of pre -defined phases.

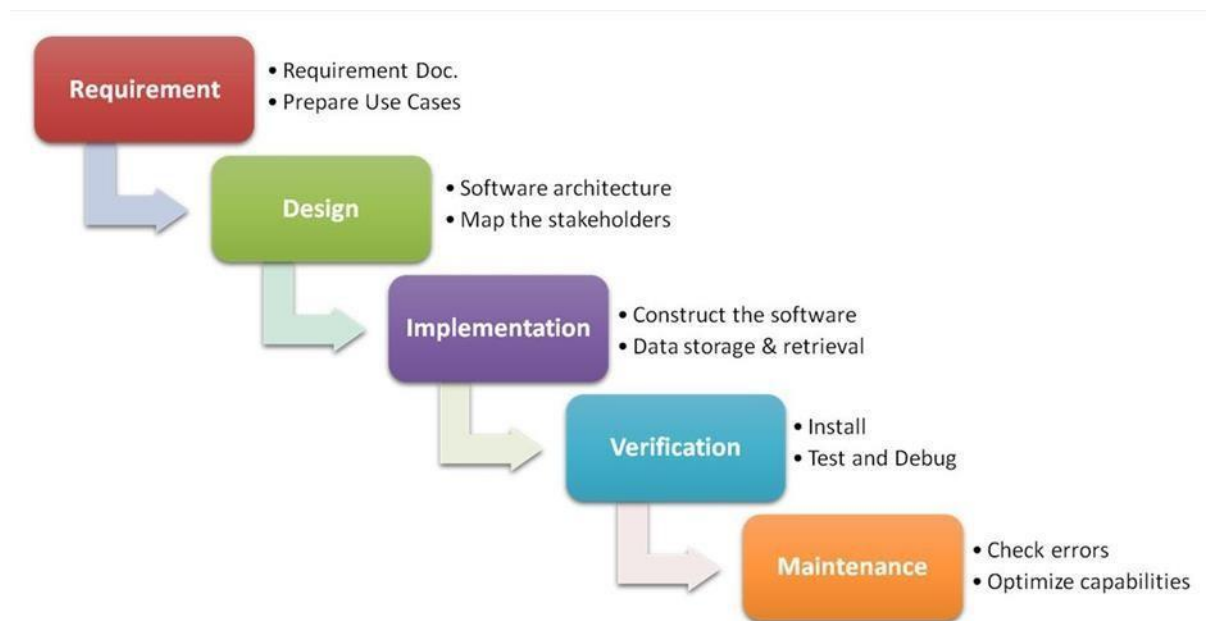


FIG 3- 1 WATERFALL MODEL

(Reference: <https://www.lucidchart.com/blog/waterfall-project-management-methodology>)

Requirement gathering and documentation

In this stage, you should gather comprehensive data approximately what this challenge requires. You may gather this data in a diffusion of ways, from interviews to questionnaires to interactive brainstorming. By means of the stop of this section, the task requirements have to be clean, and also you have to have a necessities file that has been allotted on your team.

System Design

Using the well-known requirements, your team designs the solutions. During this phase, no development will be happening. But the project team starts specification such as programming language or hardware requirements.

Implementation

During this phase software development coding will be happening. Web application

programmers take data from the previous stage and create a functional product. Web application programmers write source code in small pieces, which are integrated at the end of this phase or the beginning of the next.

Testing

Once all coding is done, testing of the product can begin. Testers methodically find and report any problems. If serious issues arise, your project may need to return to phase one for reevaluation.

Delivery/Deployment

In this phase, the solution is complete, and your project team submits the deliverables to be deployed or released

Maintenance

The final solution has been implemented to the client and is being used. As troubles arises, the project team might also want to create patches and updates might also to deal with them. Again, huge troubles may also necessitate a return to segment one.

RAD MODEL

Agile is an iterative, team-based approach to development. This method highlights the speedy delivery of a software application in comprehensive functional components.

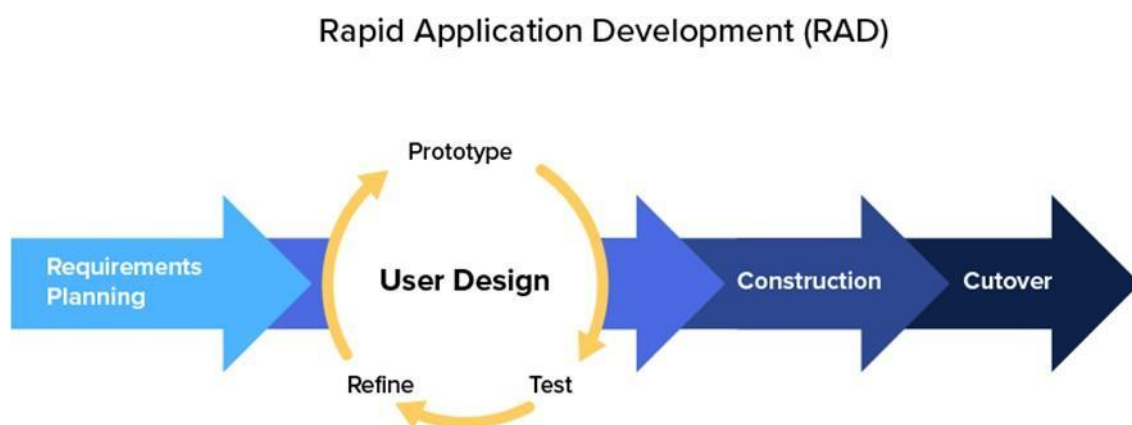


FIG 3- 2 RAD MODEL

References (<https://www.lucidchart.com/blog/rapid-application-development-methodology>)

Phase 1: Requirements planning

This phase is equivalent to a project scoping meeting. Although the planning phase is condensed compared to other project management methodologies, this is a critical step for the ultimate success of the project.

During this stage, web application programmers, clients (software users), and team members communicate to determine the goals and expectations for the project as well as current and potential issues that would need to be addressed during the build.

A basic breakdown of this stage involves:

Researching the current problem

Defining the requirements for the
project

Finalizing the requirements with each stakeholder's approval.

It is important that everyone has the opportunity to evaluate the goals and expectations for the project and weigh in. By getting approval from each key stakeholder and web application programmer, teams can avoid miscommunications and costly change orders down the road.

Phase 2: User Design

Once the project is scoped out, it's time to jump right into development, building out the user design through various prototype iterations.

This is the meat and potatoes of the RAD methodology—and what sets it apart from other project management strategies. During this phase, clients work hand in hand with web application programmers to ensure their needs are being met at every step in the design process. It's almost like customizable software development where the users can test each prototype of the product, at each stage, to ensure it meets their expectations.

All the bugs and kinks are worked out in an iterative process. The web application programmer designs a prototype, the client (user) tests it, and then they come together to communicate on what worked and what didn't.

This method gives web application programmers the opportunity to tweak the model as they go until they reach a satisfactory design.

Both the software web application programmers and the clients learn from the experience to

make sure there is no potential for something to slip through the cracks.

Phase 3: Rapid Construction

Phase 3 takes the prototypes and beta systems from the design phase and converts them into the working model.

Because the majority of the problems and changes were addressed during the thorough iterative design phase, web application programmers can construct the final working model more quickly than they could by following a traditional project management approach.

The phase breaks down into several smaller steps:

Preparation for rapid construction

Program and application development

Coding

Unit, integration, and system testing

The software development team of programmers, coders, testers, and web application programmers work together during this stage to make sure everything is working smoothly and that the end result satisfies the client's expectations and objectives.

This third phase is important because the client still gets to give input throughout the process. They can suggest alterations, changes, or even new ideas that can solve problems as they arise.

Phase 4: Cutover

This is the implementation phase where the finished product goes to launch. It includes data conversion, testing, and changeover to the new system, as well as user training.

All final changes are made while the coders and clients continue to look for bugs in the system.

Benefits of RAD methodology

RAD is one of the most successful software development programs available today, with numerous benefits for both software development teams as well as their clients.

Here are just a few advantages:

- RAD lets you break the project down into smaller, more manageable tasks.

- The task-oriented structure allows project managers to optimize their team's efficiency by

assigning tasks according to members' specialties and experience.

Clients get a working product delivered in a shorter time frame.

Regular communication and constant feedback between team members and stakeholders increases the efficiency of the design and build process.

3.1 EXISTING SYSTEM

The existing system build a formal model of language-based technique in the human-assisted computer-based proof tool Coq using locally nameless representation. Furthermore, we demonstrate effectiveness of locally nameless representation in carrying out formal machine- readable proof of soundness of the language-based technique.

The existing system enable formal reasoning about Android's permission system using computer-based theorem prover tool; a formal model of the Android's permissions is built using the logic behind theorem prover. We augment the language-based security technique with computer-aided verification tool, and locally nameless representation is formalized.

We mechanically (using a computer) check correctness of terms (programs), and build a formal model of Android type checker.

The existing system demonstrate that locally nameless approach facilitates the proof process and guarantees well-typed (correct) Android code, and mechanically carry out formal proof of soundness of the Android type checker.

3.2 DRAW BACKS OF EXISTING SYSTEM

Does not provide information flow guarantees

Put the mobile device and carrier network at risk

May perform the privileged task

Data flows may result in unintended security consequences

High Computation cost

3.3 PROPOSED SYSTEM

In the proposed system, the applications are defined as different types according to their functionalities in the Android markets, and the applications that belong to the same category have most of characteristics of this type of clustering. In other words, they have a higher similarity.

The proposed framework scales both in terms of code size and number of applications by leveraging the unprecedented computational power of cloud computing. The framework uses numerous heuristics and software analysis techniques to intelligently guide the generation of

test cases aiming to boost the likelihood of discovering vulnerabilities. The proposed approach aims to answer the following overarching question: Given advanced software testing techniques and ample processing power, what software security vulnerabilities could be uncovered automatically?

In the proposed approach, a user cannot use a policy of choose one and reject other i.e. the user cannot selectively accept few permissions, while rejecting others in order to install the app. Among the list of all the permissions an app might call for a fishy permission to access some resource among the other valid permissions. The app will still be able to confirm the successful, though insecure, installation.

The main reason for the difference of security application rate is that the classification of categories is in accordance with the requirements of people but not strictly classified by functionality and features of applications.

3.4 ADVANTAGES OF PROPOSED SYSTEM

- Security enforcement systems

- Can be used to type-check well-typed expressions

- Identify internal data communication security detecting the over-privileged applications

- Less Computational cost

- Help user find potentially malicious applications

CHAPTER 4

REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need.

Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management. In this article, we'll discuss the main types of requirements for software products and provide a number of recommendations for their use.

4.2 HARDWARE & SOFTWARE SPECIFICATION

HARDWARE SPECIFICATION

Hardware Requirement

Hardware Minimum Requirement

Disk Space 32 GB or more, 10 GB or more for Foundation Edition

Processor 1.4 GHz 64 bit

Memory 512 MB

Display (800 × 600) Capable video adapter and monitor

SOFTWARE REQUIREMENTS

Backend Technologies

- Python

- Numpy

- Sci-learn

- Eclipse IDE

Frontend Technologies

- Web Technologies

4.3 SOFTWARE DESCRIPTION

4.3.1 ABOUT PYTHON

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems.

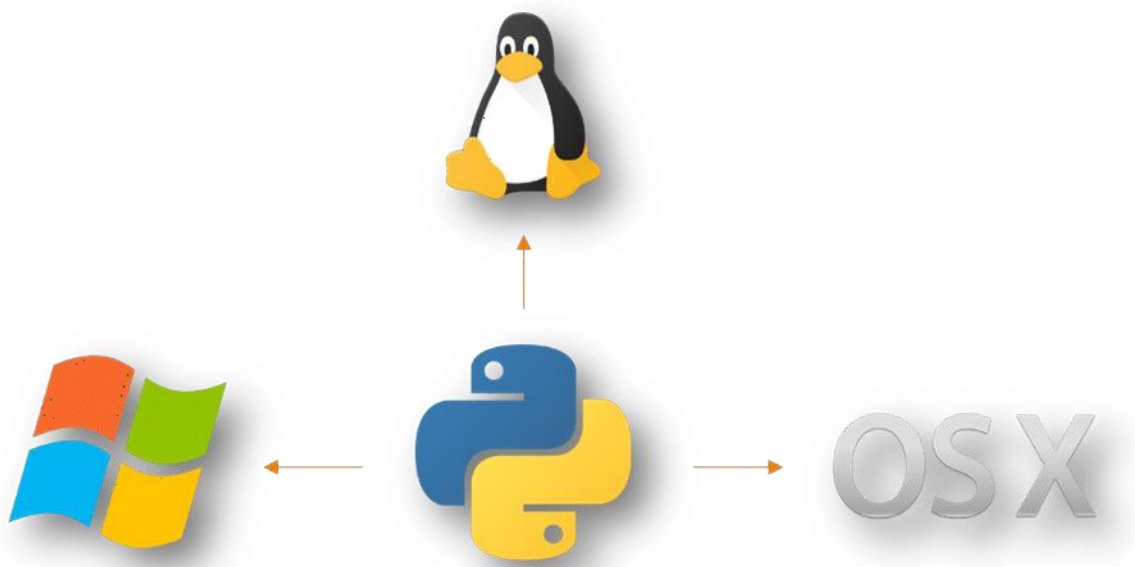


FIG 4.3.1 - 1 PYTHON

Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.

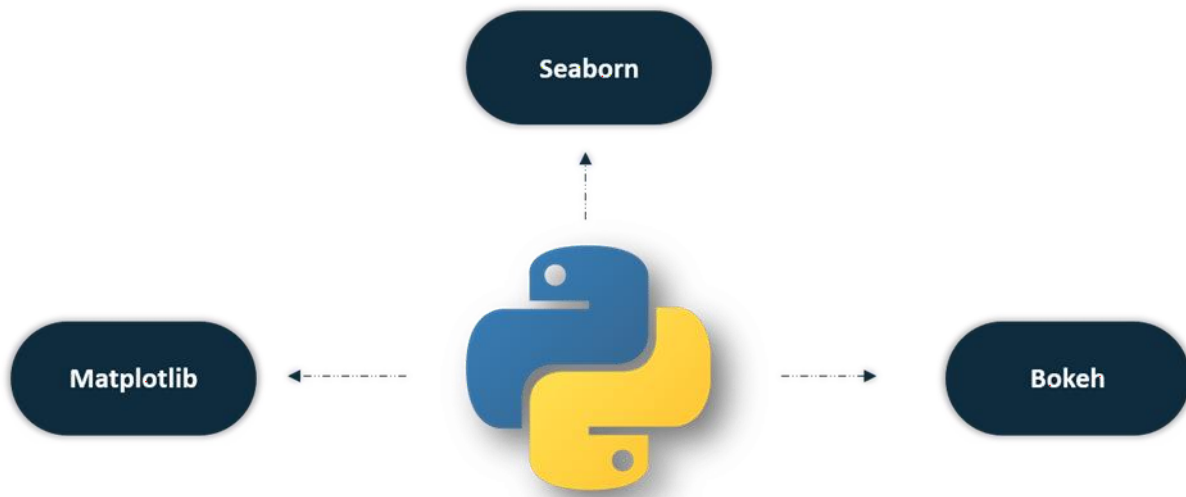


FIG 4.3.1 - 2 PYTHON IMPLEMENTING MACHINE LEARNING

In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

Python Line Structure

Python coding style comprises physical lines as well as logical lines or statements. A physical line in a Python program is a sequence of characters, and the end of the line terminates the line sequence as opposed to some other languages, such as C and C++ where a semi-colon is used to mark the end of the statement. A logical line, on the other hand, is composed of one or more physical lines. The use of a semi-colon is not prohibited in Python, although it's not mandatory. The NEWLINE token denotes the end of the logical line. A logical line that only contains spaces, comments, or tabs are called blank lines and they are ignored by the interpreter.

As we saw that in Python, a new line simply means that a new statement has started. Although, Python does provide a way to split a statement into a multiline statement or to join multiple statements into one logical line. This can be helpful to increase the readability of the statement. Following are the two ways to split a line into two or more lines:

Explicit Line Joining

In explicit line joining, we use a backward slash to split a statement into a multiline statement.

Implicit Line Joining

Statements that reside inside [], {}, or () parentheses can be broken down into two or more physical lines without using a back slash.

Multiple Statements on a Single Line

In Python, it is possible to club multiple statements in the same line using a semi-colon; however, most programmers do not consider this to be a good practice as it reduces the readability of the code.

Whitespaces and Indentation

Unlike most of the programming languages, Python uses indentation to mark a block of code. According to Python coding style guideline or PEP8, we should keep an indent size of four.

Most of the programming languages provide indentation for better code formatting and don't enforce to have it. But in Python it is mandatory. This is why indentation is so crucial in Python.

Comments in any programming language are used to increase the readability of the code. Similarly, in Python, when the program starts getting complicated, one of the best ways to maintain the readability of the code is to use Python comments. It is considered a good practice to include documentations and notes in the python syntax since it makes the code way more readable and understandable to other programmers as well, which comes in handy when multiple programmers are simultaneously working on the same project.

The code can only explain how it does something and not why it does that, but Python comments can do that. With Python comments, we can make documentations for various explanations in our code itself. Comments are nothing but tagged lines of codes which increase the readability of a code and make it self-explanatory. There are different ways of creating comments depending on the type of comment we want to include in our code.

Following are different kinds of comments that can be included in our Python program:

- Single Line Comments

- Docstring Comments

- Multiline Comments

Single line Python comments are marked with # character. These comments end at the end of the physical line, which means that all characters starting after the # character (and lasts till the end of the line) are part of the comment.

Python has the documentation strings (or docstrings) feature which is usually the first statement included in functions and modules. Rather than being ignored by the Python Interpreter like regular comments, docstrings can actually be accessed at the run time using the dot operator.

It gives programmers an easy way of adding quick notes with every Python module, function, class, and method. To use this feature, we use triple quotes in the beginning of the documentation string or comment and the closing triple quotes at the end of the documentation comment. Docstrings can be one-liners as well as multi-liners.

Unlike some programming languages that support multiline comments, such as C, Java, and more, there is no specific feature for multiline comments in Python. But that does not mean that it is totally impossible to make multiline comments in Python. There are two ways we can include comments that can span across multiple lines in our Python code.

Python Block Comments: We can use several single line comments for a whole block. This type of comment is usually created to explain the block of code that follows the Block comment. Python Block comment is the only way of writing a real comment that can span across multiple lines. It is supported and preferred by Python's PEP8 style guide since Block comments are ignored by Python interpreter or parser. However, nothing is stopping programmers from using the second 'non-real' way of writing multiline comments in Python which is explained below.

Using Docstrings: Docstrings are largely used as multiline comments in Python by many programmers since it is the closest thing to having a multiline comment feature in Python. While it is not wrong to use docstrings when we need to make multiline comments, it is important to keep in mind that there is a significant difference between docstrings and comments. Comments in Python are totally ignored by the Python Interpreter, while docstrings, when used inside the Python function, can be accessed at the run time.

Python Data Types

One of the most crucial part of learning any programming language is to understand how data is stored and manipulated in that language. Users are often inclined toward Python because of its ease of use and the number of versatile features it provides. One of those features is dynamic typing.

In Python, unlike statically typed languages like C or Java, there is no need to specifically

declare the data type of the variable. In dynamically typed languages such as Python, the interpreter itself predicts the data type of the Python Variable based on the type of value assigned to that variable.

Advantages of Python

Universal Language Construct

Support both High Level and Low-Level Programming

Language Interoperability

Fastest Development life cycle therefore more productive coding environment

Less memory used because a single container holds multiple data types and each type doesn't require its own function

Learning Ease and open-source development

Speed and user-friendly data structure

Extensive and extensible libraries.

Simple & support IoT

And many more.

Based on the 3 languages that are described above, we decided to use Python as programming language for developing this e-Voting web based application. The important motives are

Easy to learn, even non experienced programmers can use it. Ex: spacing and tabbing instead of extra syntax

Interactive mode

Large and comprehensive standard libraries

Python programs resemble to that of pseudo-code. This makes it a basis and a must have for beginner programmers due to its extreme ease a difficulty when compared to C++ Java, Perl, and so forth

About Pandas

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The Data Frame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is

used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy,

plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandas' Data Frames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and Data Frames.

1. Series: a pandas Series is a one-dimensional data structure ("a one-dimensional ndarray") that can store values — and for every value it holds a unique index, too.
2. DataFrame: a panda DataFrame is a two (or more) dimensional data structure — basically a table with rows and columns. The columns have names and the rows have indexes.

Those who are familiar with R know the data frame as a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frame's rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc.

Now, Data Frames in Python are very similar: they come with the Pandas library, and they are defined as two-dimensional labeled data structures with columns of potentially different types.

In general, you could say that the Pandas DataFrame consists of three main components: the data, the index, and the columns.

Firstly, the DataFrame can contain data that is:

- a Pandas DataFrame
- a Pandas Series: a one-dimensional labeled array capable of holding any data type with axis labels or index. An example of a Series object is one column from a DataFrame.
- a NumPy ndarray, which can be a record or structured
- a two-dimensional ndarray

- dictionaries of one-dimensional ndarray's, lists, dictionaries or Series.

Some of the key features of Python Pandas are as follows:

It provides DataFrame objects with default and customized indexing which is very fast and

efficient.

- There are tools available for loading data of different file formats into in-memory data objects.
- It is easy to perform data alignment and integrated handling of missing data in Python Pandas.
- It is very simple to perform pivoting and reshaping of data sets in Pandas.
- It also provides indexing, label-based slicing, and sub-setting of large data sets.
- We can easily insert and delete columns from a data structure.
- Data aggregation and transformations can be done using group by.
- High-performance merging and joining of data can be done using Pandas.
- It also provides time series functionality.
- Inserting and deleting columns in data structures.
- Merging and joining data sets.
- Reshaping and pivoting data sets.
- Aligning data and dealing with missing data.
- Manipulating data using integrated indexing for DataFrame objects.
- Performing split-apply-combine on data sets using the group by engine.
- Manipulating high-dimensional data in a data structure with a lower dimension using hierarchical axis indexing.
- Subsetting, fancy indexing, and label-based slicing data sets that are large in size.
- Generating data range, converting frequency, date shifting, lagging, and other time-series functionality.
- Reading from files with CSV, XLSX, TXT, among other formats.
- Arranging data in an order ascending or descending.
- Filtering data around a condition.
- Analyzing time series.
- Iterating over a data set.

Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just can't

miss when you're learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices. To work with these arrays, there's a vast amount of high-level mathematical functions operate on these matrices and arrays.

an array is basically nothing but pointers. It's a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and
- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next row.

Or, in other words, an array contains information about the raw data, how to locate an element and how to interpret an element.

With NumPy, we work with multidimensional arrays. We'll dive into all of the possible types of multidimensional arrays later on, but for now, we'll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with. In fact, it's just a different way of thinking about a list of lists. A matrix has rows and columns. By specifying a row number and a column number, we're able to extract an element from a matrix.

We can create a NumPy array using the `numpy.array` function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. Because we want all of the elements in the array to be float elements for easy computation,

we'll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats.

NumPy has several advantages over using core Python mathematical functions, a few of which are outlined here:

- NumPy is extremely fast when compared to core Python thanks to its heavy use of C extensions.
- Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library. Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.
- NumPy comes with a variety of built-in functionalities, which in core Python would take a fair bit of custom code.

Matplotlib

Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data visualization, and this property can be achieved by making use of Matplotlib.

Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy. Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams.

Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well.

Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB, if it is used in combination with Numpy and Scipy. Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB. This can make it extremely easy for MATLAB users to migrate to matplotlib.

Matplotlib can be used to create publication quality figures in a variety of hardcopy formats

and interactive environments across platforms.

Another characteristic of matplotlib is its steep learning curve, which means that users usually make rapid progress after having started. The official website has to say the following about this: "matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code."

4.3.2 ABOUT IDE

IDE – Integrated Development Environment. It is software suite. It contains different instruments for web application software engineers to compose and test applications.

Programming Web application software engineers utilize wide scope of devices habitually all through application source code creation, building and application testing. Programming Development apparatuses frequently incorporate word processors for source code creation, code libraries for reuse, compilers and test stages.

An Integrated Development Environment (IDE) is a product application that gives a programming domain to streamline creating and troubleshooting programming. Instead of playing out every one of the means required to make an executable program as disconnected individual assignments, it brings every one of the devices required into one application and workspace. Every one of the devices has a familiarity with the earth, and they cooperate to display a consistent improvement set for the designer.

An IDE typically has an editor and a compiler window. There, we can write our code, run it, and compile it. Hence, it becomes much easier to develop applications using an IDE. The major reasons for using an IDE are given below:

- We can easily write our code in its text editor window.
- To store the resource files of a project, we have a project editor window.
- It makes debugging of the code easier.
- An IDE's autosuggest feature helps us in writing efficient code.
- We can import other projects for integrating with the existing one.

Without an IDE, a product web application software engineer must choose, send, incorporate and deal with these instruments independently. An IDE brings huge numbers of those improvement related instruments all together system, application or administration. The

incorporated toolset is intended to streamline programming advancement and can distinguish

and limit coding missteps and grammatical errors.

- ✓ Komodo
- ✓ NetBeans
- ✓ PyCharm
- ✓ Python for Visual Studio.NET
- ✓ PyDev
- ✓ PyStudio
- ✓ LiClipse (Eclipse)

LiClipse is at the core for all Python based application development activities. It's been available for many years. LiClipse is a set of plugins to enhance Eclipse. It improves the overall Eclipse experience. It is best IDE tool to develop Python Applications.

Advantages of LiClipse:

- Perfect Debugger
- Error checking features
- An enhanced ability to analyze the code for quality and security concerns
- Auto complete feature

Based on the various IDE that are described above, we decided to use LiClipse as IDE (Integrated Development Environment) for developing this e-Voting web-based application.

Important Features That Make LiClipse our IDE of Choice

- Rich Reusable Libraries are available
- Model Driven Development
- Enterprise Python Tooling
- Powerful Debugging options

IDE is a program that is used to write applications (other programs) in a variety of programming languages. It allows you to complete many distinct program development steps in one place (program window). A program which allows you to do this is known as an integrated development environment, or IDE. The program is free and easy to import onto your home computer.

LiClipse is used for writing programs in Python. There are several steps that must be

completed to set up LiClipse and further steps to start up a new Python project (set of Python components which work together to make up an application).

PyCharm is an Integrated Development Environment (IDE) that provides tools for writing efficient code. It supports various frameworks such as Django, Flask, and Pyramid. PyCharm is majorly used by developers for application development with Python language.

Let's suppose, we are making a project using Python programming language. We know that Python is extensively popular among the developer's community, as well as in industries, due to its following features:

- Simple to code: Python is a very simple language compared to other programming languages. Also, it is a developer-friendly language.
- Object-oriented language: Python is an object-oriented language that uses concepts of classes and objects.
- GUI programming support: We can create graphical user interfaces using Python Tkinter library. Besides, we can use modules such as PyQt5, PyQt4, and wxPython to create a GUI application.
- High-level language: Python is a high-level programming language. We do not need to take care of managing memory or the system architecture while writing Python codes.
- Portability: Python is a portable programming language, i.e., we can run Python codes on any platform. For example, we can run a code written for Windows in Linux, UNIX, and Mac, without changing it.
- Integrated language: Python is an integrated programming language. We can easily integrate C, C++, etc. with Python.
- Interpreted language: Python is an interpreted programming language as well. Python's source code is changed to bytecode. Due to this, it is directly executed without compiling.
- Large set of standard libraries: Python consists of a rich set of modules, packages, and libraries. It helps in various stages of application development.

"notebook" or "notebook documents" denote documents that contain both code and rich text elements, such as figures, links, equations, ... Because of the mix of code and text elements,

these documents are the ideal place to bring together an analysis description, and its results, as well as, they can be executed perform the data analysis in real time.

The Jupyter Notebook App produces these documents.

As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet.

Its two main components are the kernels and a dashboard.

A kernel is a program that runs and introspects the user's code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages.

The dashboard of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary.

CHAPTER 5

SYSTEM DESIGN

5.1 INTRODUCTION

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency.

5.2 ARCHITECTURE DIAGRAM

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components.

What is a diagram? What are the types of diagrams for architecture? The Dragon1 open EA Method makes it very clear: if a diagram does not show a concept, principle or part of a principle, it is NOT an architecture diagram, because it does not show (a part of) the architecture.

There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework can be used at the system planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.

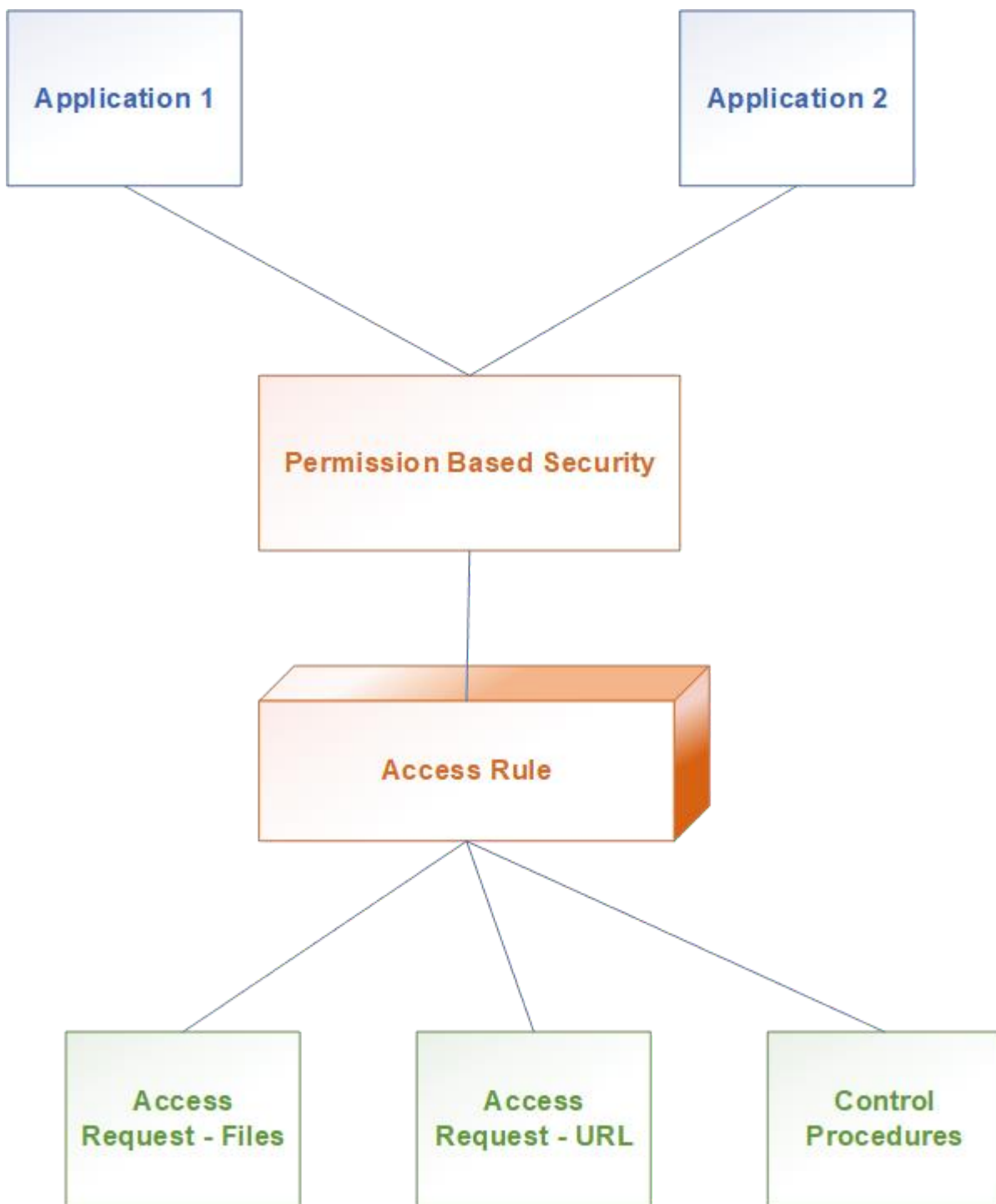


FIG 5.1 - 1 ARCHITECTURE DIAGRAM

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 LIST OF MODULES

Module 1: Exploratory Data Evaluation

Module 2: Pre-processing

Module 3: Feature Engineering

Module 4: Prediction

6.1.1 Module 1: Exploratory Data Evaluation

Exploratory Data Analysis (EDA) is the first step in your data analysis process. Here, you make sense of the data you have and then figure out what questions you want to ask and how to frame them, as well as how best to manipulate your available data sources to get the answers you need. You do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in your existing data, using visual and quantitative methods to get a sense of the story this tells.

Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and data scientists but can be very informative about a particular business.

EDA is performed in order to define and refine the selection of feature variables that will be used for machine learning. Once data scientists become familiar with the data set, they often have to return to feature engineering step, since the initial features may turn out not to be serving their intended purpose. Once the EDA stage is complete, data scientists get a firm feature set they need for supervised and unsupervised machine learning.

6.1.2 Module 2: Pre-processing

Sometimes you may find some data are missing in the dataset. We need to be equipped to

handle the problem when we come across them. Obviously, you could remove the entire line of data but what if you are unknowingly removing crucial information? Of course, we would not

want to do that. One of the most common idea to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data.

The library that we are going to use for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data.

Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import `test_train_split` from `model_selection` library of `scikit`.

6.1.3 Module 3: Feature Engineering

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. For basic guidance, you can refer to the following table for defining correlation co-efficient.

Pearson's Correlation: It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1.

LDA: Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

ANOVA: ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

Chi-Square: It is a is a statistical test applied to the groups of categorical features to evaluate

the likelihood of correlation or association between them using their frequency distribution.

6.1.4 Module 4: Prediction

Once training is complete, it's time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

A good rule of thumb I use for a training-evaluation split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If you have a lot of data, perhaps you don't need as big of a fraction for the evaluation dataset.

Once you've done evaluation, it's possible that you want to see if you can further improve your training in any way. We can do this by tuning our parameters. There were a few parameters we implicitly assumed when we did our training, and now is a good time to go back and test those assumptions and try other values.

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

CHAPTER 7

SOFTWARE TESTING

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual and organization. Projects which contain all documents have a high level of maturity. Careful documentation can save the time, efforts and wealth of the organization.

If the testing or development team gets software that is not working correctly and developed by someone else, so to find the error, the team will first need a document. Now, if the documents are available then the team will quickly find out the cause of the error by examining documentation. But, if the documents are not available then the tester need to do black box and white box testing again, which will waste the time and money of the organization. More than that, Lack of documentation becomes a problem for acceptance.

Benefits of Documentation

- Documentation clarifies the quality of methods and objectives.
- It ensures internal coordination when a customer uses software application.
- It ensures clarity about the stability of tasks and performance.
- It provides feedback on preventive tasks.
- It provides feedback for your planning cycle.
- It creates objective evidence for the performance of the quality management system.

The test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the user's point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as

documentation testing.

This is one of the most cost-effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

the most popular testing documentation files are test reports, plans, and checklists. These documents are used to outline the team's workload and keep track of the process. Let's take a look at the key requirements for these files and see how they contribute to the process.

Test strategy. An outline of the full approach to product testing. As the project moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

Test data. The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end-user in a ready solution.

Test plans. A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. It's the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

Test scenarios. In scenarios, testers break down the product's functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

Test cases. If the test scenario describes the object of testing (what), a scenario describes a procedure (how). These files cover step-by-step guidance, detailed conditions, and current inputs of a testing task. Test cases have their own kinds that depend on the type of testing — functional, UI, physical, logical cases, etc. Test cases compare available resources and current conditions with desired outcomes and determine if the functionality can be released

or not.

Traceability Matrix. This software testing documentation maps test cases and their requirements. All entries have their custom IDs — team members and stakeholders can track the progress of any tasks by simply entering its ID to the search.

External documentation collects information from inner documentation but also emphasize on providing a visual data representation — graphs, diagrams, etc.

External reports — these documents collect information on test results and can describe an entire project or a particular piece of functionality.

Test summary report — the file with final test results and findings, presented to stakeholders.

Bug reports — such files keep track of newly encountered bugs and their fixes. We prefer to keep our bug documentation numbered, so it's easier to mention them in further documentation. Reports are concise and focus on offering tangible solutions. Sometimes, bug reports can only include issue description, if the team hasn't yet found the best approach to fixing the problem.

The combination of internal and external documentation is the key to a deep understanding of all testing processes. Although stakeholders typically have access to the majority of documentation, they mostly work with external files, since they are more concise and tackle tangible issues and results. Internal files, on the other hand, are used by team members to optimize the testing process.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

But as a curiosity or Out of the box thinking, have you ever wondered how developers test their own code? What method do they use to test before releasing code for testing? How is dev-testing important in an agile process? The answer to all this is Unit Testing. I want to educate you on the importance of Unit Testing so that development and testing teams can work more collaboratively to design, test and release an excellent application.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Most of us might know the classic definition of Unit Testing

“Unit Testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code

(unit) written for implementing requirements.



FIG 7 - 1 IMPLEMENTATION

Unit Testing is used to design robust software components that help maintain code and eliminate the issues in code units. We all know the importance of finding and fixing defects in the early stage of the software development cycle. Unit Testing serves the same purpose.

Unit Testing is an integral part of the agile software development process. When a nightly build run unit test suite should run and report should be generated. If any of the unit tests have failed then the QA team should not accept that build for verification.

If we set this as a standard process, many defects would be caught in the early development cycle, saving much testing time.

I know many developers hate to write unit tests. They either ignore or write bad unit test cases due to tight scheduled or lack of seriousness (yea they write empty unit tests, so 100% of them pass successfully ;-)). It's important to write good unit tests or don't write them at all. It's even more important to provide sufficient time and a supportive environment for real benefits.

- Testing can be done in the early phases of the software development lifecycle when other modules may not be available for integration
- Fixing an issue in Unit Testing can fix many other issues occurring in later

development and testing stages

- Cost of fixing a defect found in Unit Testing is very less than the one found in the system

or acceptance testing

- Code coverage can be measured
- Fewer bugs in the System and Acceptance testing
- Code completeness can be demonstrated using unit tests. This is more useful in the agile process. Testers don't get the functional builds to test until integration is completed. Code completion cannot be justified by showing that you have written and checked in the code. But running Unit tests can demonstrate code completeness.
- Expect robust design and development as developers write test cases by understanding the specifications first.
- Easily identify who broke the build
- Saves development time: Code completion may take more time but due to decreased defect count overall development time can be saved.

7.1 GENERAL

Unit Testing frameworks are mostly used to help write unit tests quickly and easily. Most of the programming languages do not support unit testing with the inbuilt compiler. Third-party open source and commercial tools can be used to make unit testing even more fun.

List of popular Unit Testing tools for different programming languages:

- Java framework – JUnit
- PHP framework – PHPUnit
- C++ frameworks – UnitTest++ and Google C++
- .NET framework – NUnit
- Python framework – py.test

- Unit Testing

- Integration Testing
- System Testing
- Sanity Testing
- Smoke Testing
- Interface Testing
- Regression Testing
- Beta/Acceptance Testing
- Non-functional Testing types include:

Performance Testing

- Load Testing
- Stress Testing
- Volume Testing
- Security Testing
- Compatibility Testing
- Install Testing
- Recovery Testing
- Reliability Testing
- Usability Testing
- Compliance Testing
- Localization Testing

7.3 UNIT TESTING

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual

program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of

an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

Unit Testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

Most of us might know the classic definition of Unit Testing –

“Unit Testing is the method of verifying the smallest piece of testable code against its purpose.” If the purpose or requirement failed then the unit test has failed.

In simple words, Unit Testing means – writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

7.4 BLACK BOX TESTING

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks app's functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

7.5 INTEGRATION TESTING

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction

between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger project, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.

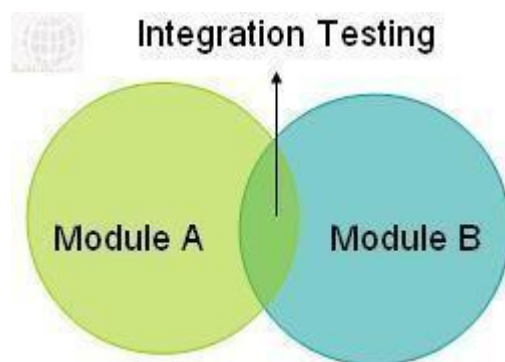


FIG 7 - 3 INTEGRATION TESTING

7.6 SYSTEM TESTING

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.

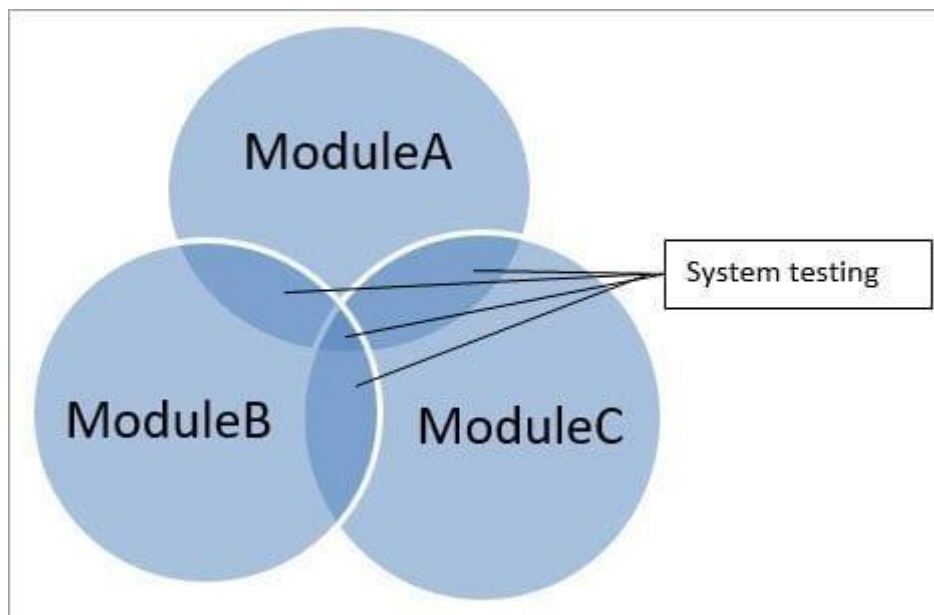


FIG 7 - 4 SYSTEM TESTING

System testing is a method of monitoring and assessing the behaviour of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on users-perspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the

functional specification or system specification documentation. In most cases, it is done next

to the Integration testing, as this testing should be covering the end-to-end system's actual routine. This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the customer's point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing; it seeks to detect both defects within the "inter-assemblages".

7.7 SANITY TESTING

Sanity Testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not.

The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, the build is rejected by the testing team to save time and money. It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing. Let's assume that in an e-commerce project, there are mainly five modules- like login page, home page, user profile page, new user creation, etc. There is a defect in the login page when the password field accepts less than four alpha numeric characters and the requirement mentions that this password field should not be below eight characters. So, the bug is reported by the testing team to the development team to resolve it. When the development team fixes the defect and passes it to the testing team. Then the testing team checks if the changes done are working fine. It also checks if it does not have an impact on other related functionalities. Now there is a functionality to update the password in the user profile page. As part of the sanity check, we would need to validate the login page as well as the profile page to ensure that the checks are working fine at both the

places. Sanity tests are generally performed on build where the production deployment is required immediately - like a critical bug fix.

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as “Day 0” check. It is done at the “build level”.

It helps not to waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

To conduct smoke testing, we don't write test cases. We just pick the necessary test cases from already written test cases.

Do we really write test cases for all testing types? Here in this article, we have given clear idea on choosing testing types to write test cases.

As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In general, we pick minimal number of test cases that won't take more than half an hour to execute.

After receiving a Software build with the minor issues fixes in code or functionality, Sanity testing is carry out to check whether the bugs reported in previous build are fixed & there is regression introduced due to these fixes i.e. not breaking any previously working functionality. The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform.

Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure.

After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically, this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing.

7.8 REGRESSION TESTING

Regression Testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are re-executed in order to

check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.

For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough variety—such as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prioritization—to prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule.

But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent projects from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your company's bottom line.

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. The process incorporates “performance” indicators such as:

- Browser, page, and network response times
- Server request processing times
- Acceptable concurrent user volumes
- Processor memory consumption; number and type of errors that might be encountered with app

Performance testing gathers all the tests that verify an application's speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent

users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application.

Performance testing is the testing that is performed to ascertain how the components of a system are performing under a certain given situation.

Resource usage, scalability, and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of a software product.



FIG 7 - 5 PERFORMANCE TESTING

Software Performance testing is type of testing perform to determine the performance of system to major the measure, validate or verify quality attributes of the system like responsiveness, Speed, Scalability, Stability under variety of load conditions. The system is tested under a mixture of load conditions and check the time required responding by the system under varying workloads. Software performance testing involves the testing of application under test to ensure that application is working as expected under variety of load conditions. The goal of performance testing is not only find the bugs in the system but also eliminate the performance bottlenecks from the system.

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as "Endurance testing" and

“Volume testing”. The main purpose of load testing is to monitor the response time and staying power

of application when system is performing well under heavy load. Load testing comes under the Non-Functional Testing & it is designed to test the non-functional requirements of a software application.

Load testing is performed to make sure that what amount of load can be withstand the application under test. The successfully executed load testing is only if the specified test cases are executed without any error in allocated time.

Simple examples of load testing:

- Testing printer by sending large job.
- Editing a very large document for testing of word processor
- Continuously reading and writing data into hard disk.
- Running multiple applications simultaneously on server.
- Testing of mail server by accessing thousands of mailboxes
- In case of zero-volume testing & system fed with zero

CHAPTER 8

RESULT & ANALYSIS

TRAINING & CROSS VALIDATION SCORE

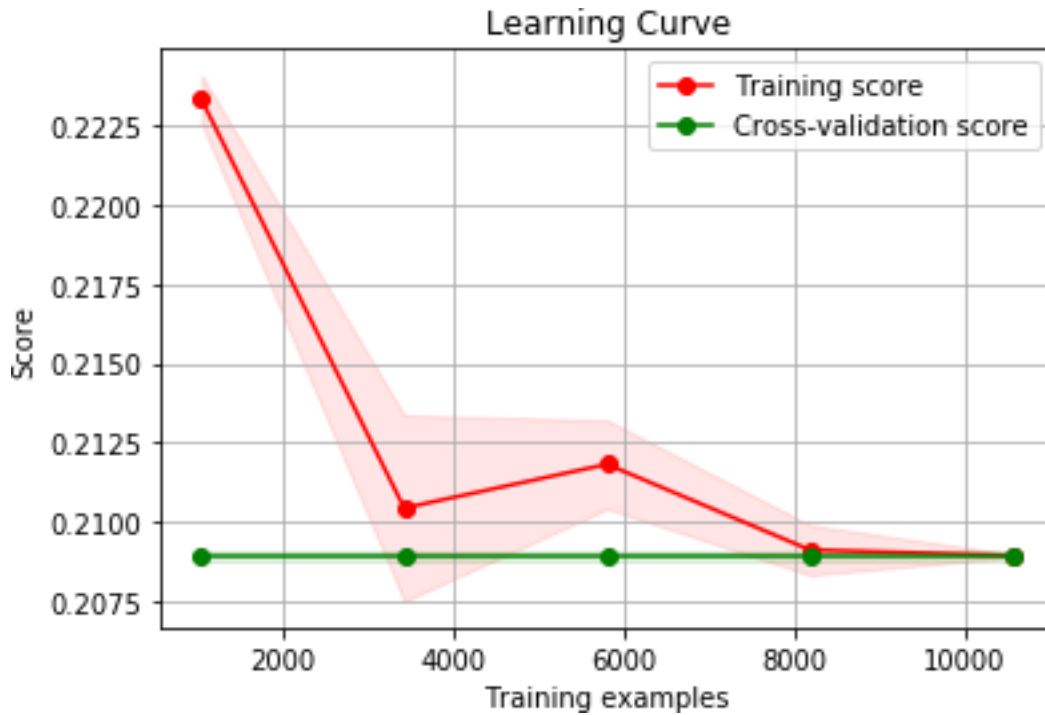


Figure 8-1 TRAINING & CROSS VALIDATION GRAPH

ANOMALY SCORE CURVE

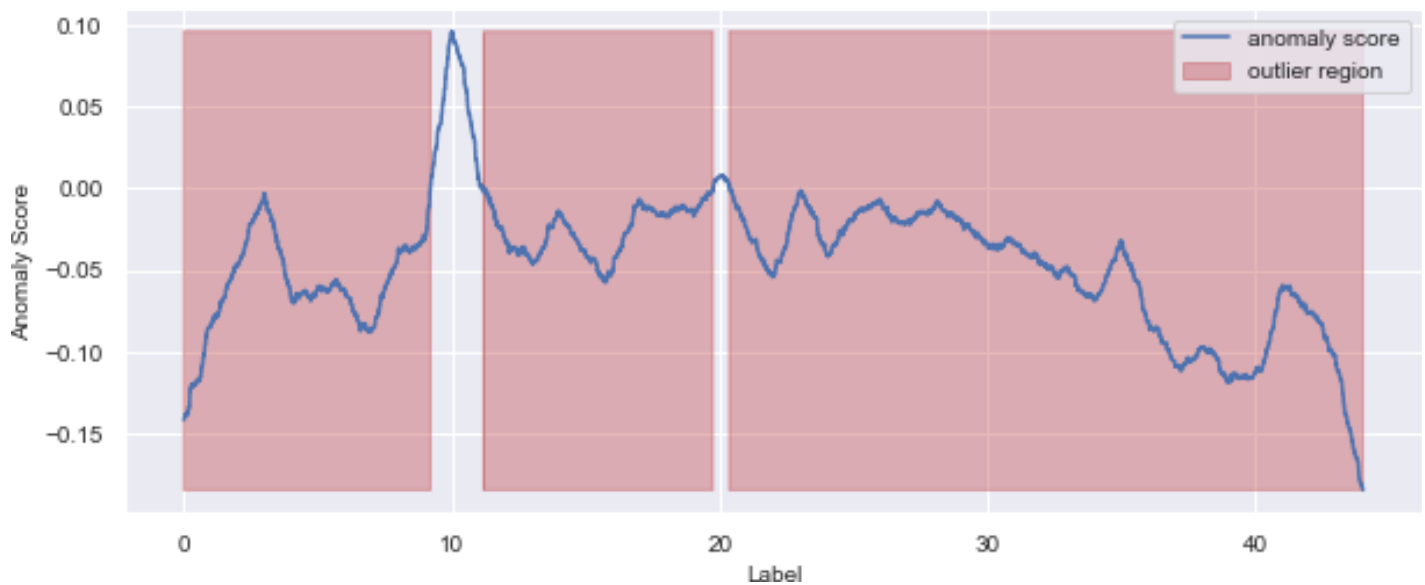


Figure 8-2 ANAMOLY SCORE CURVE

VARIENCE V/S NUMBER OF COMPONENTS CURVE

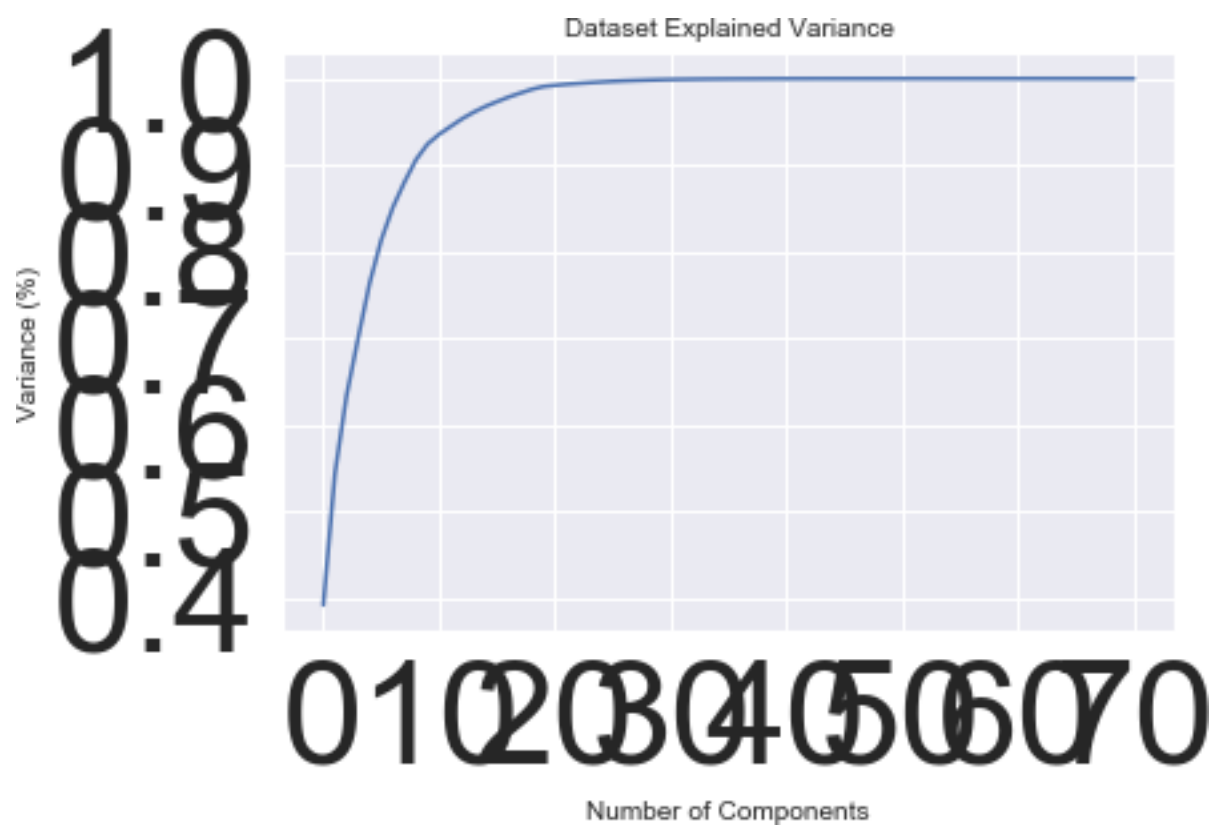


Figure 8-4 VARIENCE V/S NUMBER OF COMPONENTS CURVE

CHAPTER 9

CONCLUSION & FUTURE ENHANCEMENT

9.1 CONCLUSION & FUTURE ENHANCEMENT

The proposed system provides an assessment method in accordance with certain logic, reliable theoretical basis and existing research results in order to help users estimate the applications before installation. Via evaluation and statistical analysis, the assessment method can identify a part of malicious applications, mainly for those that apply for far too much permission or have relatively large differences with other applications which belong to the same category. novice users of Android are not aware of the permissions they have granted at the time of installation. Our proposed system provided the second level of security to the users, without modifying the underlying structure of Android operating system.

REFERENCES

- [1] K. Stenzel, K. Katkalov, M. Borek, and W. Reif, "A model-driven approach to noninterference," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 5, no. 3, pp. 30–43, September 2014.
- [2] S. Rasthofer, S. Arzt, and E. Bodden, "A machine-learning approach for classifying and categorizing android sources and sinks," *Proceedings of the Network and Distributed System Security Symposium*, 2014.
- [3] M. Whitney, H. Richter Lipford, B. Chu, and T. Thomas. "Embedding Secure Coding Instruction into the IDE: Complementing Early and Intermediate CS Courses with ESIDE" In press, *Journal of Educational Computing Research*, 2017
- [4] B. Fisseha Demissie, D. Ghio, M. Ceccato, A. Avancini Identifying Android Inter-app Communication Vulnerabilities Using Static and Dynamic Analysis, *IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2016
- [5] T. Dallas, "13 Must-Know Tips for Keeping Your Phone Secure," [Online]. Available: <https://android.gadgethacks.com/how-to/android-security-13-must-know-tips-for-keeping-your-phone-secure-0162723/>. [Accessed 11 October 2017].
- [6] Khokhlov and L. Reznik, "Data Security Evaluation for Mobile Android Devices," in *IEEE Information Security and Protection of Information Technologies Conference*, St. Petersburg, 2017.
- [7] Kaur and D. Upadhyay, "PeMo: Modifying application's permissions and preventing information stealing on smartphones," 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), Sep. 2014.
- [8] Z. Fang, W. Han, and Y. Li, "Permission based Android security: Issues and countermeasures," *Computers & Security*, vol. 43, pp. 205–218, Jun. 2014.
- [9] L. Xing, X. Pan, R. Wang, K. Yuan, and X. Wang, "Upgrading Your Android, Elevating My Malware: Privilege Escalation Through Mobile OS Updating," *IEEE Symp. Secur. Priv.*, 2014.
- [10] L. Whitney, "Almost no one is using Android marshmallow, still," *CNET*, 2016.

[Online]. Available:
androidmarshmallow- still.

<http://www.cnet.com/news/almost-no-one-is-using->

- [11] J.-K. Park and S.-Y. Choi, "Studying security weaknesses of Android system," International Journal of Security and Its Applications, vol. 9, no. 3, pp. 7–12, Mar. 2015.
- [12] Security and P. Foundation, "Gimme Rat – Android Malware In The Wild," XSec Technologies Pvt Ltd and Cyber Security & Privacy Foundation, India, p.5, mar 2014.
- [13] S. Calzavara, I. Grishchenko, and M. Maffei, "Horndroid: Practical and sound static analysis of android applications by smt solving," in Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P), Mar. 2016, pp. 4762.
- [14] M. I. Gordon, D. Kim, J. H. Perkins, L. Gilham, N. Nguyen, and M. C. Rinard, "Information flow analysis of Android applications in droidsafe," in Proc. NDSS, 2015, pp. 116.
- [15] G. Betarte, J. Campo, M. Cristiá, F. Gorostiaga, C. Luna, and C. Sanz, "Towards formal model-based analysis and testing of Android's security mechanisms," in Proc. 43rd Latin Amer. Comput. Conf. (CLEI), Sep. 2017, pp. 110.
- [16] G. Betarte, J. D. Campo, C. Luna, and A. Romano, "Verifying Android's permission model," in Proc. Int. Colloq. Theor. Aspects Comput., 2015, pp. 485504.

A. PLAGIARISM REPORT



Don't Worry! This report is 100% safe & secure. It's not available publically and it's not accessible by search engines (Google, Yahoo, Bing, etc)

Sentence

MALWARE DETECTION ON ANDROID Kanisetty Sai Saran¹, B. K. N. Vardhan², Dr. J. Refonaa³ 1B.E, Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India 2B.E, Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India 3M.E., Ph.D., Professor, School of Computing, Sathyabama Institute of Science and Technology, Chennai, India Abstract: -Even though android are permission based there are a lot of possible ways by which malicious software can infect our devices by spam or even messages without being our knowledge. By using android permission model, we analyse what permissions the apps are using and, we establish the permission model to quantify the functional characteristics of the application, and then provide an assessment method in which we use the network visualization techniques and clustering algorithm to determine whether the testing application is potentially malicious application or not so as to help users choose applications before installation. INTRODUCTION The given system has some consequences as other apps where malicious files are automatically downloaded and executed in the background without the prior knowledge of the users and can steal the data. In this paper we propose and implement a Runtime detection and prevention system. The existing system demonstrate that locally nameless approach facilitates the proof process and guarantees well-typed (correct) Android code, and mechanically carry out formal proof of soundness of the Android type check. AI is quite possibly the most mainstream strategy of foreseeing the future or ordering data to help individuals in settling on important choices. AI calculations are utilized to be prepared by past outcomes or models through which they gain from past encounters and furthermore examine the previous information. In this way, it prepares the information by perusing and investigating them and finishes up its example and ready to recognize designs to make forecasts about what's to come. Information assumes the main part in the mission learning. By utilizing the above information or learned information we can make precise outcomes and can prepare adequately. For instance, there is a X Ray report of the disease patient, by perusing and the X beam reports and qualities we can prepare the AI calculations so that it can distinguish them. By concentrating such reports and preparing the information we can recognize disease reports and ordinary reports and from various perspectives Machine Learning calculations are helpful to us. Subtleties are acquired genuinely utilizing the information. We can create AI models by utilizing Mathematics and we can actualize those models by utilizing Computer Science methods. The interaction doesn't end exclusively by building AI models. We likewise need to upgrade those models consummately to get better outcomes. Streamlining strategies are utilized to enhance the outcomes for better execution of the calculations EXISTING SYSTEM The current framework constructs a conventional model of language-based procedure. Ongoing investigations expresses that there is a quick development in the malware assaults. Moreover, we make straightforward and reasonable itemized investigation of the malware distinguishing calculations via preparing and testing different methods. DISADVANTAGES OF EXISTING SYSTEM Does not provide information flow guarantees Put the mobile device and carrier network at risk Data flows may result in unintended security consequences May perform the privileged task LITERATURE SURVEY Behavioural-Based Classification and Identification of Ransomware Variants Using Machine Learning On account of the changing behavior of ransomware, ordinary course of action and recognizable proof systems don't correctly recognize new varieties of ransomware. Software engineers use different sorts of procedures like polymorphic and groundbreaking strategies to avoid disclosure of imprints on contraptions. We use AI techniques o appreciate and analyze their direct. To lead our examination, we used direct reports of 150 ransomware tests from 10 unmistakable ransomware families. Our dataset contains different sorts of high successful Ransome item so we can inspect them, giving an evaluation of the plan precision of AI counts on the current propelling status of ransomware. An iterative approach is used to recognize ideal social attributes used to achieve best request exactness. Two principal bits of this assessment are Identification of the direct of the ransomware we use them for the ideal results and we can organize ransomware using our readied figuring. We have evaluated request precision of three AI portrayal figurings Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences Advancement and dispersal of noxious programming requires the production of new techniques for their identification. Accordingly, we started to utilize proactive innovations that utilization the test program to distinguish the presence of specific manifestations, frequently happening in malware. There is an investigation of how the program connects with the product climate that is perused/compose at certain vault keys, documents, network action the utilization of certain APIs. Because of the way that contemplated the program is conceivably destructive, to make its execution should be in a separated climate. This paper talks about proactive strategies dependent on API call investigation and propose another technique utilizing a numerous grouping arrangement to distinguish regular attributes in malware. This paper contains various techniques to recognize pernicious programming and malwares dependent on their programming interface calls and time stamps. We introduced a totally new malware identification technique

dependent on numerous or distinctive grouping API calls arrangement. This strategy is utilized to depict in detail and executed in programming. A test on a bunch of programming and the authenticity of the viral nature. Testing has shown that the set up plan of serious shows and recognizes noxious programming with high exactness. The impact of malware evolution on the analysis methods and infrastructure The huge number of malwares introduced each day demands methods and tools for their automated analyses. Complex and distributed infrastructure of malicious software and new sophisticated techniques used to obstruct the analyses are discussed in this paper based on real-life malware observations observed for a long time ago. Their impact is used on both toolsets and methods are to be presented based on practical development of systems for malware analyses and new features for existing tools. Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences Advancement and dispersal of noxious programming requires the making of new strategies for their recognition. Hence, we started to utilize proactive advances that utilization the test program to identify the presence of specific indications, frequently happening in malware. Dynamic investigation of the read program dispatched for execution. There is an investigation of how the program associates with the product climate that is perused/compose at certain library keys, documents, network action the utilization of specific API calls. We as a whole realize that the program is destructive and we ought to execute in a space called sand box which is isolated. This paper talks about proactive strategies dependent on API call examination and propose another strategy utilizing a numerous arrangement to distinguish basic characteristics in malware. The paper thinks about the plan to identify pernicious programming, in light of API calls, every one of which is executed in programming. We additionally built up another malware distinguishing technique which depends on various or diverse arrangement API calls arrangement. This strategy is depicted in detail and the portrayed technique is executed in programming. Testing has shown that the set up plan of serious shows and distinguishes noxious programming with high exactness.

PROPOSED SYSTEM In the proposed framework, the applications are characterized as various sorts as per their functionalities in the Android markets, and the applications that have a place with a similar classification have the majority of attributes of this kind of grouping. As such, they have a higher comparability. The proposed structure scales both as far as code size and number of utilizations by utilizing the exceptional computational force of distributed computing. The structure utilizes various heuristics and programming investigation procedures to cleverly direct the age of experiments intending to help the probability of finding weaknesses. The proposed approach expects to address the accompanying all-encompassing inquiry: Given progressed programming testing procedures and abundant handling power, what programming security weaknesses could be uncovered naturally? In the proposed approach, a user cannot use a policy of choose one and reject other i.e., the user cannot selectively accept few permissions, while rejecting others in order to install the app. Among the list of all the permissions an app might call for a fishy permission to access some resource among the other valid permissions. The app will still be able to confirm the successful, though insecure, installation. The main reason for the difference of security application rate is that the classification of categories is in accordance with the requirements of people but not strictly classified by functionality and features of application.

ADVANTAGES OF PROPOSED SYSTEM Security enforcement systems Can be used to type-check well-typed expressions Identify internal data communication security detecting the over-privileged applications Less Computational cost Help user find potentially malicious applications.

ALGORITHM USED K-NEAREST NEIGHBOUR ALGORITHM: K closest neighbours is a basic calculation that stores every accessible case and orders new cases dependent on a similitude measure. Kth Nearest Neighbour makes expectations utilizing the preparation dataset straightforwardly. The algorithm can be summarized as: A positive integer k is specified, along with a new sample We select the k entries in our database which are closest to the new sample We find the most common classification of these entries This is the classification we give to the new sample

STEP 1: BEGIN

STEP 2: Input: D =

STEP 3: $x = (x_1, \dots, x_n)$ new instance to be classified

STEP 4: for every tagged instance (x_i, c_i) calculate $d(x_i, x)$

STEP 5: Order $d(x_i, x)$ from lowest to highest, $(i = 1, \dots, N)$

STEP 6: choose the K nearest instances to x: $D_k x$

STEP 7: Assign to x the foremost frequent category in $D_k x$

STEP 8: finish

Expectations are made for another case (x) via looking through the whole preparing set for the K most comparable cases (the neighbours) and summing up the yield variable for those K examples. For relapse this may be the mean yield variable, in arrangement this may be the mode (or generally normal) class esteem. To figure out which of the K occurrences in the preparation dataset are generally like another information a distance measure is utilized. For genuine esteemed info factors, the most well-known distance measure is Euclidean distance. Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (xi) across all input attributes j.

$$\text{Euclidean Distance}(x, x_i) = \sqrt{\sum ((x_j - x_{ij})^2)}$$

LOGISTIC REGRESSION Calculated regression is a sort of polynomial regression. It is a technique to demonstrate a connection between the needy and free factors. It is utilized set up when the information shows an awe-inspiring pattern, and direct regression would not create precise outcomes when contrasted with non-straight relapse. Strategic regression predicts the yield of a downright needy variable. Consequently, along these lines the result ought to be in the worth ought to be founded on classifications. In this way, it very well may be either YES or NO or perhaps 0 or 1 or might be True or False and so forth

UML DIAGRAMS USE CASE DIAGRAMS A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system with its actors, roles, action, artifacts or classes, in order to better understanding, altering, maintaining, or documenting information about the system. Predominantly, UML has been utilized as a universally useful demonstrating language in the field of computer programming. In any case, it has now discovered its way into the recording of a few business cycles or work processes. For instance, action outlines, a sort of UML graph, can be utilized as a trade for flowcharts. They give both a more normalized method of displaying work processes just as a more extensive scope of highlights to improve clarity and proficiency.

Fig:1 Use Case Diagram ARCHITECTURE DIAGRAM A

design outline is a graphical portrayal of a bunch of ideas, that are essential for an engineering, including their standards, components and segments. What is an outline? What are the sorts of graphs for engineering? The Dragon1 open EA Method makes it extremely understood: if a graph doesn't show an idea, rule or part of a guideline, it's anything but an engineering outline, since it doesn't show (a piece of) the design. There are numerous sorts of engineering graphs, similar to a product design outline, framework design chart, application design outline, security engineering outline, and so forth. For framework designers, they need framework engineering graphs to comprehend, explain, and impart thoughts regarding the framework structure and the client prerequisites that the framework should uphold. It's a fundamental structure can be utilized at the framework arranging stage assisting accomplices with understanding the design, talk about changes, and impart goals unmistakably. Fig:2-Architecture Diagram SEQUENTIAL DIAGRAM Grouping chart model the progression of rationale inside your framework in a visual way, empowering you both to archive and approves your rationale, and ordinarily we use them for both examination and configuration reason. Arrangement chart are the most mainstream UML for dynamic demonstrating, which centers around distinguishing the conduct inside your framework. Fig:3-Sequential Diagram SCREEN SHOTS Fig 1 Anomaly Report Fig 2 Training X Cross Validation score REFERENCES K. Stenzel, K. Katkalov, M. Borek, and W. Reif, "A model-driven approach to non-interference," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), vol. 5, no. 3, pp. 30-43, September 2014. S. Rasthofer, S. Arzt, and E. Bodden, "A machine-learning approach for classifying and categorizing android sources and sinks," Proceedings of the Network and Distributed System Security Symposium, 2014 M. Whitney, H. Richter Lipford, B. Chu, and T. Thomas. "Embedding Secure Coding Instruction into the IDE: Complementing Early and Intermediate CS Courses with ESIDE" In press, Journal of Educational Computing Research, 2017 B. Fisseha Demissie, D. Ghio, M. Ceccato, A. Identifying Android Inter-App Communication Vulnerabilities Using Static and Dynamic Analysis, IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILE Soft), 2016 T. Dallas, "13 Must-Know Tips for Keeping Your Phone Secure,"[Online]. Available: <https://android.gadgetsacks.com/how-to/android-security-13-mustknow-tips-for-keeping-your-phone-secure-0162723/>. [Accessed 11 October 2017]. Khokhlov and L. Reznik, "Data Security Evaluation for Mobile Android Devices," in IEEE Information Security and Protection of Information Technologies Conference, St. Petersburg, 2017. Kaur and D. Upadhyay, "PeMo: Modifying application's permissions and preventing information stealing on smartphones," 2014 5th International Conference - Confluence the Next Generation Information Technology Summit (Confluence), Sep. 2014. Z. Fang, W. Han, and Y. Li, "Permission based Android security: Issues and countermeasures," Computers & Security, vol. 43, pp. 205-218, Jun. 2014.

Report Title:	Malware Detection On Android
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/229460b0058f5a33696d5f645a6ca67393ce01615471987
Report Generated Date:	11 March, 2021
Total Words:	2418
Total Characters:	16316
Keywords/Total Words Ratio:	81.85%
Excluded URL:	No
Unique:	89%
Matched:	11%

Sentence wise detail:

Keywords Density		
One Word	2 Words	3 Words
program 1.63%	api calls 0.44%	institute science technology 0.22%
work 1.33%	malicious software 0.3%	science technology chennai 0.22%
info 1.19%	noxious programming 0.3%	frequently happening malware 0.15%
information 1.11%	chennai india 0.22%	noxious programming high 0.15%
programming 1.11%	euclidean distance 0.22%	testing shown set 0.15%

Plagiarism Report

By check-plagiarism.com

B. JOURNAL PAPER

MALWARE DETECTION ON ANDROID

Kanisetty Sai Saran¹, B. K. N. Vardhan², Dr. J. Refonaa³

¹B.E, Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

²B.E, Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

³M.E., Ph.D., Professor, School of Computing, Sathyabama Institute of Science and Technology, Chennai, India

Abstract: -Even though android are permission based there are a lot of possible ways by which malicious software can infect our devices by spam or even messages without being our knowledge. By using android permission model, we analyse what permissions the apps are using and, we establish the permission model to quantify the functional characteristics of the application, and then provide an assessment method in which we use the network visualization techniques and clustering algorithm to determine whether the testing application is potentially malicious application or not so as to help users choose applications before installation.

I. INTRODUCTION

The given system has some consequences as other apps where malicious files are automatically downloaded and executed in the background without the prior knowledge of the users and can steal the data. In this paper we propose and implement a Runtime detection and prevention system. The existing system demonstrate that locally nameless approach facilitates the proof process and guarantees well-typed (correct) Android code, and mechanically carry out formal proof of soundness of the Android type check. AI is quite possibly the most mainstream strategy of foreseeing the future or ordering data to help individuals in settling on important choices. AI calculations are utilized to be prepared by past outcomes or models through which they gain from past encounters and furthermore examine the previous information. In this way, it prepares the information by perusing and investigating them and finishes up its example and ready to recognize designs to make forecasts about what's to come.

Information assumes the main part in the mission learning. By utilizing the above information or learned information we can make precise outcomes and can prepare adequately. For instance, there is a X Ray report of the disease patient, by perusing and the X beam reports and qualities we can prepare the AI calculations so that it can distinguish them. By concentrating such reports and preparing the information we can recognize disease reports and ordinary reports and from various perspectives Machine Learning calculations are helpful to us. Subtleties are acquired genuinely utilizing the information. We can create AI models by utilizing Mathematics

and we can actualize those models by utilizing Computer Science methods.

The interaction doesn't end exclusively by building AI models. We likewise need to upgrade those models consummately to get better outcomes. Streamlining strategies are utilized to enhance the outcomes for better execution of the calculations

II. EXISTING SYSTEM

The current framework constructs a conventional model of language-based procedure. Ongoing investigations expresses that there is a quick development in the malware assaults. Moreover, we make straightforward and reasonable itemized investigation of the malware distinguishing calculations via preparing and testing different methods.

A. DISADVANTAGES OF EXISTING SYSTEM

- Does not provide information flow guarantees
- Put the mobile device and carrier network at risk
- Data flows may result in unintended security consequences
- May perform the privileged task

III. LITERATURE SURVEY

Behavioural-Based Classification and Identification of Ransomware Variants Using Machine Learning

On account of the changing behavior of ransomware, ordinary course of action and recognizable proof systems don't correctly recognize new varieties of ransomware. Software engineers use different sorts of procedures like polymorphic and groundbreaking strategies to avoid disclosure of imprints on contraptions. We use AI techniques o appreciate and analyze their direct. To lead our examination, we used direct reports of 150 ransomware tests from 10 unmistakable ransomware families. Our dataset contains different sorts of high successful Ransome item so we can inspect them, giving an evaluation of the plan precision of AI counts on the current propelling status of ransomware. An iterative approach is used to recognize ideal social attributes used to achieve best request exactness. Two

principal bits of this assessment are Identification of the direct of the ransomware we use them for the ideal results and we can organize ransomware using our readied figuring. We have evaluated request precision of three AI portrayal figurings

Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences

Advancement and dispersal of noxious programming requires the production of new techniques for their identification. Accordingly, we started to utilize proactive innovations that utilization the test program to distinguish the presence of specific manifestations, frequently happening in malware. There is an investigation of how the program connects with the product climate that is perused/compose at certain vault keys, documents, network action the utilization of certain APIs. Because of the way that contemplated the program is conceivably destructive, to make its execution should be in a separated climate. This paper talks about proactive strategies dependent on API call investigation and propose another technique utilizing a numerous grouping arrangement to distinguish regular attributes in malware. This paper contains various techniques to recognize pernicious programming and malwares dependent on their programming interface calls and time stamps. We introduced a totally new malware identification technique dependent on numerous or distinctive grouping API calls arrangement. This strategy is utilized to depict in detail and executed in programming. A test on a bunch of programming and the authenticity of the viral nature. Testing has shown that the set up plan of serious shows and recognizes noxious programming with high exactness.

The impact of malware evolution on the analysis methods and infrastructure

The huge number of malwares introduced each day demands methods and tools for their automated analyses. Complex and distributed infrastructure of malicious software and new sophisticated techniques used to obstruct the analyses are discussed in this paper based on real-life malware observations observed for a long time ago. Their impact is used on both toolsets and methods are to be presented based on practical development of systems for malware analyses and new features for existing tools.

Detection of Malicious Software on Based on Multiple Equations of API-calls Sequences

Advancement and dispersal of noxious programming requires the making of new strategies for their recognition. Hence, we started to utilize proactive advances that utilization the test program to identify the presence of specific indications, frequently happening in malware. Dynamic investigation of the read program dispatched for execution. There is an investigation of how the program associates with the product climate that is perused/compose at certain library keys, documents, network action the utilization of specific API calls. We as a whole realize that the program is destructive and we ought to execute in a space called sand box which is isolated. This paper talks about proactive strategies dependent on API call examination and propose another strategy utilizing

a numerous arrangement to distinguish basic characteristics in malware. The paper thinks about the plan to identify pernicious programming, in light of API calls, every one of which is executed in programming. We additionally built up another malware distinguishing technique which depends on various or diverse arrangement API calls arrangement. This strategy is depicted in detail and the portrayed technique is executed in programming. Testing has shown that the set up plan of serious shows and distinguishes noxious programming with high exactness.

IV. PROPOSED SYSTEM

In the proposed framework, the applications are characterized as various sorts as per their functionalities in the Android markets, and the applications that have a place with a similar classification have the majority of attributes of this kind of grouping. As such, they have a higher comparability.

The proposed structure scales both as far as code size and number of utilizations by utilizing the exceptional computational force of distributed computing. The structure utilizes various heuristics and programming investigation procedures to cleverly direct the age of experiments intending to help the probability of finding weaknesses. The proposed approach expects to address the accompanying all-encompassing inquiry: Given progressed programming testing procedures and abundant handling power, what programming security weaknesses could be uncovered naturally?

In the proposed approach, a user cannot use a policy of choose one and reject other i.e., the user cannot selectively accept few permissions, while rejecting others in order to install the app. Among the list of all the permissions an app might call for a fishy permission to access some resource among the other valid permissions. The app will still be able to confirm the successful, though insecure, installation.

The main reason for the difference of security application rate is that the classification of categories is in accordance with the requirements of people but not strictly classified by functionality and features of application.

B. ADVANTAGES OF PROPOSED SYSTEM

- Security enforcement systems
- Can be used to type-check well-typed expressions
- Identify internal data communication security detecting the over-privileged applications
- Less Computational cost
- Help user find potentially malicious applications.

V. ALGORITHM USED

1. K-NEAREST NEIGHBOUR ALGORITHM:

K closest neighbours is a basic calculation that stores every accessible case and orders new cases dependent on a similitude

measure. Kth Nearest Neighbour makes expectations utilizing the preparation dataset straightforwardly.

The algorithm can be summarized as:

A positive integer k is specified, along with a new sample

- We select the k entries in our database which are closest to the new sample
- We find the most common classification of these entries
- This is the classification we give to the new sample

STEP 1: BEGIN STEP 2: Input: $D =$

STEP 3: $x = (x_1 \dots x_n)$ new instance to be classified

STEP 4: for every tagged instance (x_i, c_i) calculate $d(x_i, x)$

STEP 5: Order $d(x_i, x)$ from lowest to highest, $(i = 1 \dots N)$

STEP 6: choose the K nearest instances to x : D_{Kx}

STEP 7: Assign to x the foremost frequent category in D_{Kx}

STEP 8: finish

Expectations are made for another case (x) via looking through the whole preparing set for the K most comparable cases (the neighbours) and summing up the yield variable for those K examples. For relapse this may be the mean yield variable, in arrangement this may be the mode (or generally normal) class esteem.

To figure out which of the K occurrences in the preparation dataset are generally like another information a distance measure is utilized. For genuine esteemed info factors, the most well-known distance measure is Euclidean distance. Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (x_i) across all input attributes j .

$$\text{Euclidean Distance } (x, x_i) = \sqrt{\sum ((x_j - x_{ij})^2)}$$

2. LOGISTIC REGRESSION

Calculated regression is a sort of polynomial regression. It is a technique to demonstrate a connection between the needy and free factors. It is utilized set up when the information shows an awe-inspiring pattern, and direct regression would not create precise outcomes when contrasted with non-straight relapse.

Strategic regression predicts the yield of a downright needy variable. Consequently, along these lines the result ought to be in the worth ought to be founded on classifications. In this way, it very well may be either YES or NO or perhaps 0 or 1 or might be True or False and so forth

VI. UML DIAGRAMS

1. USE CASE DIAGRAMS

A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system with its actors, roles, action, artifacts or classes, in order to better understanding, altering, maintaining, or documenting information about the system.

Predominantly, UML has been utilized as a universally useful demonstrating language in the field of computer programming. In any case, it has now discovered its way into the recording of a few business cycles or work processes. For instance, action outlines, a sort of UML graph, can be utilized as a trade for flowcharts. They give both a more normalized method of displaying work processes just as a more extensive scope of highlights to improve clarity and proficiency.

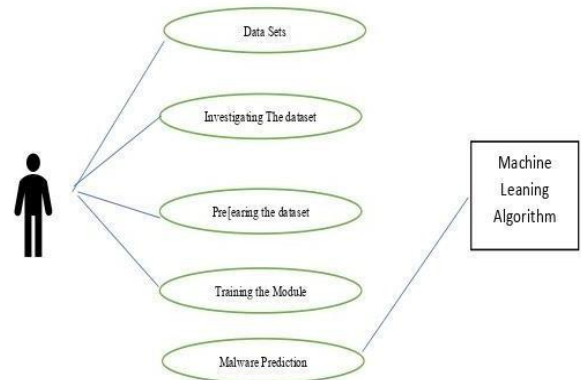


Fig:1 Use Case Diagram

2. ARCHITECTURE DIAGRAM

A design outline is a graphical portrayal of a bunch of ideas, that are essential for an engineering, including their standards, components and segments.

What is an outline? What are the sorts of graphs for engineering? The Dragon1 open EA Method makes it extremely understood: if a graph doesn't show an idea, rule or part of a guideline, it's anything but an engineering outline, since it doesn't show (a piece of) the design.

There are numerous sorts of engineering graphs, similar to a product design outline, framework design chart, application design outline, security engineering outline, and so forth.

For framework designers, they need framework engineering graphs to comprehend, explain, and impart thoughts regarding the framework structure and the client prerequisites that the framework should uphold. It's a fundamental structure can be utilized at the framework arranging stage assisting accomplices with understanding the design, talk about changes, and impart goals unmistakably.

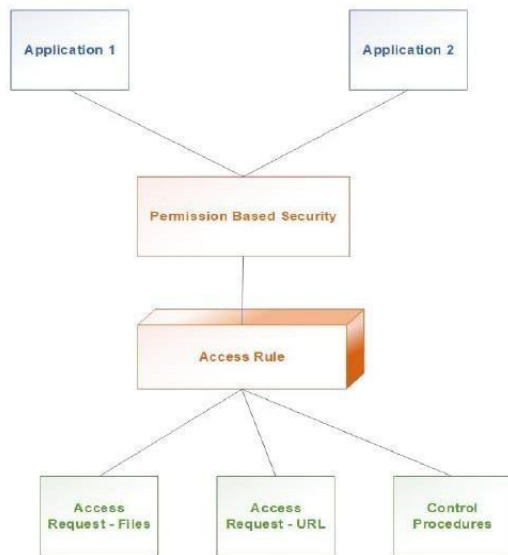


Fig.2-Architecture Diagram

3. SEQUENTIAL DIAGRAM

Grouping chart model the progression of rationale inside your framework in a visual way, empowering you both to archive and approves your rationale, and ordinarily we use them for both examination and configuration reason. Arrangement chart are the most mainstream UML for dynamic demonstrating, which centers around distinguishing the conduct inside your framework.

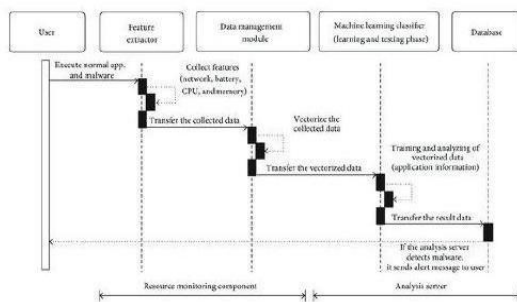


Fig.3-Sequential Diagram

VII. SCREEN SHOTS

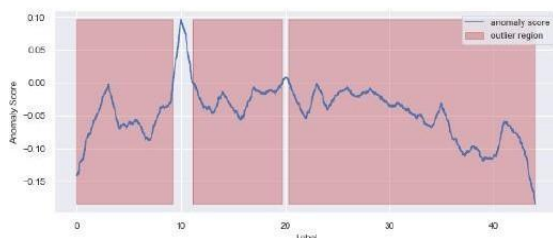


Fig 1 Anomaly Report

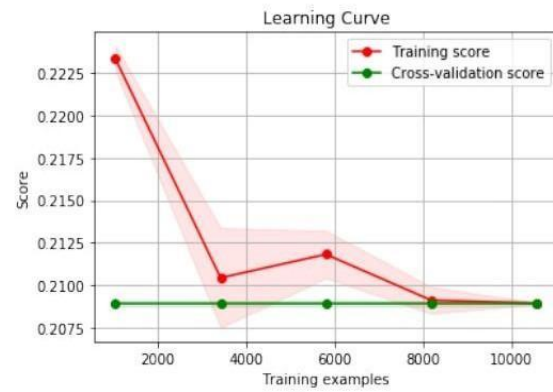


Fig 2 Training X Cross Validation score

VIII. REFERENCES

- [1] K. Stenzel, K. Katkalov, M. Borek, and W. Reif, "A model-driven approach to non-interference," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), vol. 5, no. 3, pp. 30–43, September 2014.
- [2] S. Rasthofer, S. Arzt, and E. Bodden, "A machine-learning approach for classifying and categorizing android sources and sinks," Proceedings of the Network and Distributed System Security Symposium, 2014.
- [3] M. Whitney, H. Richter Lipford, B. Chu, and T. Thomas. "Embedding Secure Coding Instruction into the IDE: Complementing Early and Intermediate CS Courses with ESIDE" In press, Journal of Educational Computing Research, 2017
- [4] B. Fisseha Demissie, D. Ghio, M. Ceccato, A. Identifying Android Inter-App Communication Vulnerabilities Using Static and Dynamic Analysis, IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILE Soft), 2016
- [5] T. Dallas, "13 Must-Know Tips for Keeping Your Phone Secure,"[Online]. Available: <https://android.gadgethacks.com/how-to/android-security-13-mustknow-tips-for-keeping-your-phone-secure-0162723/>. [Accessed 11 October 2017].
- [6] Khokhlov and L. Reznik, "Data Security Evaluation for Mobile Android Devices," in IEEE Information Security and Protection of Information Technologies Conference, St. Petersburg, 2017.
- [7] Kaur and D. Upadhyay, "PeMo: Modifying application's permissions and preventing information stealing on smartphones," 2014 5th International Conference - Confluence the Next Generation Information Technology Summit (Confluence), Sep. 2014.
- [8] Z. Fang, W. Han, and Y. Li, "Permission based Android security: Issues and countermeasures," Computers & Security, vol. 43, pp. 205–218, Jun. 2014.

C. SOURCE CODE

```
import pandas as pd

%matplotlib inline

df = pd.read_csv('data/balanced_dataset.csv')

df = df.drop([df.columns[0]], axis=1)

df.drop_duplicates(keep=False, inplace=True)

df.head()

df.shape

df.describe()

df['Label'].value_counts()

len(df['Label'].value_counts())

df['Label'].isnull().sum()

df.isnull().sum()

df.dropna(subset=['Label'], inplace=True)

import sys

sys.path.append("..")

from analysis_scripts.classes_distribution import plot_all_classes_distribution

plot_all_classes_distribution(df)

plot_all_classes_distribution(df[df.Label != 'BENIGN'])

plot_all_classes_distribution(df[df.Label.str.contains('SCAREWARE')])

plot_all_classes_distribution(df[df.Label.str.contains('ADWARE')])

plot_all_classes_distribution(df[df.Label.str.contains('RANSOMWARE')])

plot_all_classes_distribution(df[df.Label.str.contains('SMSMALWARE')])

df['Timestamp'] = pd.to_datetime(df['Timestamp'], errors='coerce')

df.select_dtypes(exclude=['int', 'float', 'datetime'])
```

```
df['Packet Length Std'] = pd.to_numeric(df['Packet Length Std'], errors='coerce')
```

```

df['CWE Flag Count'] = pd.to_numeric(df['CWE Flag Count'], errors='coerce')

df.select_dtypes(exclude=['int', 'float', 'datetime'])

df['Flow ID'] = df['Flow ID'].astype('str')

df['Source IP'] = df['Source IP'].astype('str')

df['Destination IP'] = df['Destination IP'].astype('str')

len(df['Flow ID'].unique())

df.shape

df.groupby('Label').describe()

from sklearn.preprocessing import LabelEncoder

class_encoder = LabelEncoder()

df["Label"] = class_encoder.fit_transform(df["Label"])

df.Label.unique()

df["Flow ID"] = LabelEncoder().fit_transform(df["Flow ID"])

df["Source IP"] = LabelEncoder().fit_transform(df["Source IP"])

df["Destination IP"] = LabelEncoder().fit_transform(df["Destination IP"])

nunique = df.apply(pd.Series.nunique)

cols_to_drop = nunique[nunique ==

1].index
df.drop(cols_to_drop, axis=1,

inplace=True) df.shape

import scipy.stats

scipy.stats.anderson(df['Label'], dist='norm')

from analysis_scripts.correlation import plot_correlation

plot_correlation(df[df.columns.difference(['Label', 'Timestamp'])], 'kendall')

from analysis_scripts.correlation import drop_lin_correlated

filtered_out_features = drop_lin_correlated(df[df.columns.difference(['Label', 'Timestamp'])],

```

method="kendall")

```

filtered_out_features.columns.values

len(filtered_out_features.columns.values)

plot_correlation(filtered_out_features,
'kendall')

from analysis_scripts.feature_selection import select_k_best

from sklearn.feature_selection import mutual_info_classif, f_classif

frs_mi = select_k_best(mutual_info_classif, df[df.columns.difference(['Label', 'Timestamp'])],
df.Label)

frs_anova = select_k_best(f_classif, df[df.columns.difference(['Label', 'Timestamp'])], df.Label)

set(filtered_out_features.columns.values) & set(frs_anova) & set(frs_mi)

plot_correlation(df[list(set(filtered_out_features.columns.values) & set(frs_anova) &
set(frs_mi))], 'kendall')

from statsmodels.stats.outliers_influence import variance_inflation_factor

from statsmodels.tools.tools import add_constant

def calc_vif(df):

    X = add_constant(df)

    vif_df = pd.Series([variance_inflation_factor(X.values, i) for i in range(X.shape[1])],
index=X.columns).to_frame()

    vif_df = vif_df.sort_values(by=0, ascending=False).rename(columns={0: 'VIF'})

    vif_df = vif_df.drop('const')

    return vif_df

vif = calc_vif(df[df.columns.difference(['Label', 'Timestamp'])])

vif[vif.VIF <= 5].head()

vif_kendall = calc_vif(filtered_out_features)

vif_kendall[vif_kendall.VIF <= 5]

```

```
vif_anova = calc_vif(df[frs_anova])
```

```
vif_anova
```

```
vif_anova[vif_anova.VIF <= 5]
```

```
vif_mi = calc_vif(df[frs_mi])

vif_mi

vif_mi[vif_mi.VIF <= 5]

vif_intersect = calc_vif(df[list(set(filtered_out_features.columns.values) & set(frs_anova) &
set(frs_mi))])

vif_intersect

set(vif_mi[vif_mi.VIF <= 5].index) & set(vif_intersect.index) & set(vif_anova[vif_anova.VIF <=
5].index) & set(vif_kendall[vif_kendall.VIF <= 5].index)

from sklearn.decomposition import PCA

from sklearn.preprocessing import MinMaxScaler

import numpy as np

import matplotlib.pyplot as plt


scaler = MinMaxScaler(feature_range=[0, 1])

data_rescaled = scaler.fit_transform(df[df.columns.difference(['Label', 'Timestamp'])])


pca = PCA().fit(data_rescaled) #Plotting the Cumulative Summation of the Explained Variance

plt.figure()

plt.plot(np.cumsum(pca.explained_variance_ratio_))

plt.xlabel('Number of Components', size = 10)

plt.ylabel('Variance (%)', size = 10)

plt.title('Dataset Explained Variance', size = 10)

plt.rc('xtick', labels=10)

plt.rc('ytick', labels=10)

plt.show()

train_features = df[df.columns.difference(['Label', 'Timestamp'])]
```

```
model = PCA(n_components=20).fit(train_features)
```

```

X_pc = model.transform(train_features)

n_pcs= model.components_.shape[0]

most_important = [np.abs(model.components_[i]).argmax() for i in range(n_pcs)]

initial_feature_names = df[df.columns.difference(['Label', 'Timestamp'])].columns.values

most_important_names = [initial_feature_names[most_important[i]] for i in range(n_pcs)]

dic = {'PC{}'.format(i+1): most_important_names[i] for i in range(n_pcs)}

pca_df = pd.DataFrame(dic.items())

pca_df

from sklearn.ensemble import IsolationForest

isolation_forest = IsolationForest(n_estimators=100)

isolation_forest.fit(df['Label'].values.reshape(-1, 1))

xx=np.linspace(df['Label'].min(),
df['Label'].max(), len(df[df.columns.difference(['Timestamp'])])).reshape(-1,1)

anomaly_score = isolation_forest.decision_function(xx)

outlier = isolation_forest.predict(xx)

plt.figure(figsize=(10,4))

plt.plot(xx, anomaly_score, label='anomaly score')

plt.fill_between(xx.T[0], np.min(anomaly_score), np.max(anomaly_score),
                where=outlier== -1, color='r',
                alpha=.4, label='outlier
                region')

plt.legend(fontsize=10)

```

```
plt.ylabel('Anomaly Score', size=10)
```

```
plt.xlabel('Label', size=10)
```

```
plt.rc('xtick',  
labels=10) plt.rc('ytick',  
labels=10) plt.show()  
df['Original_label'] = class_encoder.inverse_transform(df.Label)  
df['Original_label'].unique()  
df.to_csv('encoded_dataset.csv', index=False)
```

