**VISVESVARAYA TECHNOLOGICAL UNIVERSITY** Jnana Sangama, Belagavi – 590018



**A Project Report**

**on**

**MALWARE DETECTOR FOR ANDROID PLATFORM USING MACHINE LEARNING AND DEEP LEARNING**

*Submitted in partial fulfillment for the award of degree of*

**Bachelor of Engineering**

**in**

# COMPUTER SCIENCE AND ENGINEERING

Submitted by

|  |  |
| --- | --- |
| **Ms. RAKSHITHA J** | **1BG13CS082** |
| **Ms. S VARSHA** | **1BG13CS086** |
| **Mr. SUDESH KUMAR M** | **1BG13CS107** |
| **Mr. SUHAS RAINA** | **1BG13CS108** |

**Internal Guide**

**Ms. Priyanka S**

Assistant Professor, Dept. of CSE

BNMIT



**B. N. M. Institute of Technology**

12th Main, 27th Cross, Banashankari II Stage, Bengaluru - 560 070

**Department of Computer Science and Engineering**

2016-2017

**B. N. M. Institute of Technology**

12th Main, 27th Cross, Banashankari II Stage, Bengaluru - 560 070

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



##### **CERTIFICATE**

Certified that the project entitled **Malware Detector for Android Platform Using Machine Learning and Deep Learning** carried out by

**Ms. Rakshitha J** **1BG13CS082**

**Ms. S Varsha** **1BG13CS086**

**Mr.** **Sudesh Kumar M**  **1BG13CS107**

**Mr.** **Suhas Raina**  **1BG13CS108**

are bonafied students of VIII Semester B.E, **B.N.M Institute of Technology** in partial fulfillment for the Bachelor of Engineering in COMPUTER SCIENCE AND ENGINEERING, of the **Visvesvaraya Technological University,** Belagavi, during the academic year 2016 - 2017. It is certified that all corrections / suggestions indicated for the internal assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

## Ms. Priyanka S Dr. Sahana D. Gowda Dr. Krishnamurthy G N

## Assistant Professor Professor and HOD Principal

## Dept. of CSE, BNMIT Dept. of CSE, BNMIT BNMIT

## 

**Examiner 1:**

**Examiner 2:**

**ACKNOWLEDGEMENT**

Co-operation, Determination, Dedication and Deduction are the four wheels that drive any project. & through the pages of this report, a few words of gratitude to all those distinguished personalities who guided and inspired us for the completion of this project.

We would like to thank **Shri. Narayan Rao R Maanay**, Secretary for providing excellent academic environment in college.

We would like to sincerely thank **Prof. T. J. Ramamurthy**, Director for having extended his support and encouragement during the course of work.

We would like to express my gratitude to **Dr. M. S. Suresh**, Dean for his relentless support, guidance and encouragement.

We would like to thank **Dr. Krishnamurthy G N,** Principal for his constant encouragement.

We would like to thank **Dr. Sahana D. Gowda,** Head of the Department of Computer Science and Engineering for her support and encouragement towards the completion of the project.

We would like to express my gratitude to our internal guide **Ms. Priyanka S**, Assistant Professor, Department of Computer Science and Engineering who gave us guidance in completing the project work successfully.

We would also like to thank our seminar coordinator **Ms. Pathanjali C**, Assistant Professor, Department of Computer Science and Engineering, for her support and constant encouragement.

**RAKSHITHA J**

**S VARSHA**

**SUDESH KUMAR M**

**SUHAS RAINA**

**ABSTRACT**

Smartphones and mobile tablets are rapidly becoming indispensable in daily life. Android has been the most popular mobile operating system till date, as Google Play Store and other third-party markets, play an important role in the popularity of Android devices. However, the openness of Android makes these markets hot targets for malware attacks and causes countless instances of malware being hidden behind a large number of benign apps that seriously threatens users security and privacy. The main counter-measure today to defend against malware on Android platforms is a risk communication mechanism that warns users about the permissions required before installing each app. The above stated mechanism is rather ineffective as it presents permissions in Android Malware Characterization and Detection Using Deep Learning, thus, requiring too much technical knowledge for a user to be able to differentiate malware from benign apps. Deep learning is a new area of machine learning research that has gained increasing attention in artificial intelligence. The study proposes how to associate the features from the static analysis with features from dynamic analysis of Android apps and characterize malware using deep learning techniques. Hundreds of Android apps, thoroughly tested Malware Detector can perform an in-depth analysis on the features that deep learning essentially exploits to characterize malware. Deep learning is suitable for characterizing Android malware and especially effective with the availability of more training data. Malware Detector can achieve great detection accuracy, which outperforms traditional machine learning techniques.

**CONTENTS**

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO** |
| **Acknowledgement** | **i** |
| **Abstract** | **ii** |
| **Contents** | **iii** |
| **List of Figures** | **v** |
| **List of Tables** | **vi** |
| **List of Snapshots** | **vii** |
|  |  |
| **Chapter 1 Introduction** | **1** |
| 1.1 Overview | 1 |
| 1.1.1 Malware | 1 |
| 1.1.2 Android | 1 |
| 1.2 Motivation | 1 |
| 1.3 Problem Statement | 2 |
| 1.4 Summary | 3 |
|  |  |
| **Chapter 2 Literature Survey** | 6 |
|  |  |
| **Chapter 3 System Requirements** | 17 |
| 3.1 Introduction | 17 |
| 3.2 Functional Requirements | 17 |
| 3.2.1 Java Development Kit | 18 |
| 3.2.2 Android Application Kit | 19 |
| 3.3 Non-Functional Requirements | 20 |
| 3.4 Software Requirements | 21 |
| 3.5 Hardware Requirements | 21 |
|  |  |
| **Chapter 4 System Design** | 22 |
| 4.1 Existing Model | 23 |
| 4.1.1 Architecture – Existing System | 23 |
| 4.2 Proposed Model | 24 |
| 4.2.1 Architecture – Proposed System | 24 |
| 4.3 Data Flow Diagram | 25 |
| 4.3.1 Data Flow Diagram – Level 0 | 25 |
| 4.3.2 Data Flow Diagram – Level 1 | 25 |
| 4.4 Class Diagram | 27 |
| 4.5 Sequence Diagrams | 29 |
| 4.5.1 Sequence Diagram – Feature Extraction Phase | 29 |
| 4.5.2 Sequence Diagram – Machine Learning/Deep Learning Engine | 30 |
| 4.6 Summary | 31 |
|  |  |
| **Chapter 5 Implementation** | 33 |
| 5.1 Introduction | 33 |
| 5.2 Techniques | 33 |
| 5.2.1 Machine Learning | 33 |
| 5.2.2 Artificial Neural Networks | 36 |
| 5.2.3 Deep Learning | 40 |
| 5.2.4 ARFF File Format | 44 |
| 5.2.5 Apktool | 44 |
| 5.3 Algorithms | 44 |
| 5.3.1 Support Vector Machine | 44 |
| 5.3.2 Restricted Boltzmann Machine | 48 |
| 5.3.3 Deep Belief Networks | 50 |
|  |  |
| **Chapter 6 Results** | 53 |
|  |  |
| **Chapter 7 Testing** | 60 |
| 7.1 Test Cases | 60 |
| 7.1.1 Best Case | 61 |
| 7.1.2 Worst Case | 62 |
| 7.2 Accuracy | 62 |
|  |  |
| **Chapter 8 Conclusion** | 64 |
| **References** | 65 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Title** | **Page No** |
| Figure 1.1 | Feature Extraction Module Architecture | 4 |
| Figure 1.2 | Classifier Output Scatter Plot | 5 |
| Figure 2.1 | Traditional Malware Detection System Architecture | 23 |
| Figure 4.1 | Proposed Malware Detection System Architecture | 24 |
| Figure 4.2 | Data Flow Diagram of the Malware Detector – Level 0 | 25 |
| Figure 4.3 | Data Flow Diagram of the Malware Detector – Level 1 | 25 |
| Figure 4.4 | Class Diagram of the Proposed System | 27 |
| Figure 4.5 | Sequence Diagram – Feature Extraction Phase | 29 |
| Figure 4.6 | Sequence Diagram – Machine Learning/Deep Learning Engine | 30 |
| Figure 7.1 | Machine learning versus deep learning accuracy | 63 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Title** | **Page No** |
| Table 2.1 | Summary of Literature Survey | 16 |
| Table 7.1 | Table for test cases | 61 |

**LIST OF SNAPSHOTS**

|  |  |  |
| --- | --- | --- |
| **Snapshot No** | **Title** | **Page No** |
| Snapshot 6.1 | Output for case 1 | 54 |
| Snapshot 6.2 | Output for case 2 | 55 |
| Snapshot 6.3 | Output for case 3 | 56 |
| Snapshot 6.4 | Output for case 4 | 57 |
| Snapshot 6.5 | Output for case 5 | 58 |
| Snapshot 6.6 | Output for case 6 | 59 |