# Intelligent Malware Detection And Classification Using Correlation-Based Feature Selection

**Soumya Singhai[1], Prateek Gupta[2], Nagendra Kumar[3]**
[1]Dept of CSE
[2]Prof., Dept of CSE
[1, 2] Shri Ram Institute of Science and Technology, Jabalpur, M.P.

***Abstract***- *Android is the most popular operating system for smartphones and small devices with 86.6% market share. Its open-source nature makes it more prone to attacks creating a need for malware analysis. Main approaches for detecting malware intents of mobile applications are based on either static analysis or dynamic analysis. In static analysis, apps are inspected for suspicious patterns of code to identify malicious segments. However, several obfuscation techniques are available to provide a guard against such analysis. The dynamic analysis on the other hand is a behavior-based detection method that involves investigating the run-time behavior of the suspicious app to uncover malware. The present work extracts the system call behavior of 215 features from 15036 app samples; of these, 9476 were benign samples while the remaining 5560 were malware samples to construct a feature vector for training a classifier. Five data classification techniques including Decision tree, KNN, Logistic Regression, Support Vector Machine and Sequential Learning Model were applied on this dataset. The feature correlation techniques were used to select appropriate features from the set of attributes. These techniques of feature ranking included Chi-square statistics and correlation analysis by determining the weights of the features. After discarding selected features with low ranks, the performances of the classifiers were measured using accuracy and recall. Experiments show that Sequential Neural Network Model after selecting features through correlation analysis outperformed other techniques where an accuracy of 99.00% is achieved with minimal training time.*

***Keywords***- Feature extraction, Malware, Support vector machines, Training, Decision trees; Android malware detection, predictive analytics.

## I. INTRODUCTION

Technology in every aspect of our lives provides us with many conveniences but also causes several problems. One of these problems is the increase in threats to cyber security as technology develops day by day [1], [2]. Another problem is the highly fast-growing amount of data [3]. Ensuring security has become difficult because of the extreme data increases. In addition, some creative hackers have deep knowledge of systems and programming skills that can exploit well protected hosts [4]. In the last five years alone, there have been many attacks with great destructiveness. Few of them are mentioned below:

Equifax Data Breach: One of the most notable cyber security crimes of recent years is the Equifax data breach. In 2017, hackers gained unauthorized access to Equifax systems to obtain sensitive information such as names, dates of birth; Social Security numbers (SSNs), addresses, and driver’s license identities of more than 143 million people [5].

- WannaCry Ransomware Attack: In May 2017, more than 200,000 computers were affected in 150 countries by this attack. The ransomware encrypted files on the affected computers and demanded payment in Bitcoin to restore access. This attack caused widespread disruption, including the closure of several hospitals in England [6].
- Marriott Data Breach: In 2018, Marriott announced that the personal data of up to 500 million guests were stolen. The breach, which has continued since 2014, has affected customers of Marriott properties, including Starwood Hotels [7].
- Capital One Data Breach: In July 2019, Capital One announced that the bank had a data breach that exposed the personal data of more than 100 million customers and applicants. The breach was caused by a misconfigured firewall that allowed a hacker to access data, such as names, addresses, phone numbers, email addresses, and credit scores, stored on Capital One’s cloud servers [8].
- Solar Winds Supply Chain Attack: In December 2020, it was revealed that SolarWinds software was hacked, and malicious code was injected into the Orion network monitoring software. The hack affected several private companies and numerous government agencies [9].
- Colonial Pipeline Ransomware Attack: In May 2021, Colonial Pipeline, which supplies gasoline to the eastern United States, experienced a ransomware incident that resulted in the company’s pipeline being offline for an extended period. The attack was carried out by a Russian hacking group called Dark Side, which demanded a $4.4 million ransom payment in Bitcoin. The attack caused widespread panic and fuel shortages in many states [10].

The stolen information includes classified intelligence data, financial records, and personal data. Research related to the impact of cyber security on organizations and individuals estimates that more than 1.8 million cyber security workers will be needed by the end of 2023. It is also said that organizations will spend at least $100 billion each year on cyber security protection [11], [12], [13]. It has been becoming harder to defend computer-based systems against cyber attacks. An average of 240 days to detect an intrusion is just one example. Furthermore, with the emergence of new types of attacks, the complexity of attacks is increasing daily, and security vulnerabilities are constantly increasing. It is getting increasingly harder to catch up with this speed and prevent attacks. Considering these situations, it has been seen that traditional computer algorithms used in cyber security could not identify zero-day attacks over time. For this reason, in cyber security, numerous Machine Learning (ML) techniques such as Deep Learning (DL) and Reinforcement Learning (RL) have made important developments recently [14], [15], [16].

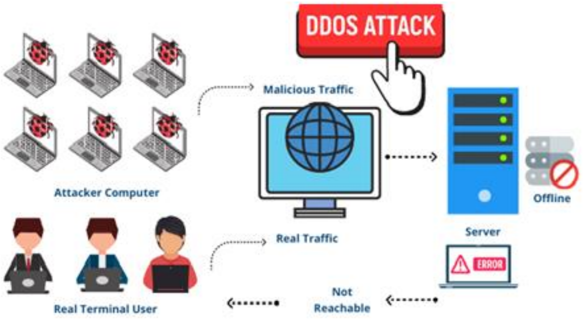


Figure1.1: DDoS Attack Example. [17]

1.2 Types of Malwares Malware, short for malware, is software designed by cyber attackers to access or damage a computer or network, often without the victim's knowledge. Malware is defined as any Software designed to cause direct harm. Although global malware volume decreased by 39% in 2020, the malware strain persists. At the same time, some types of malware link together when they use similar attacks, such as using explosives, meaning ready-made attacks that are sometimes caused by the victims themselves; use network phishing and social engineering tactics to distribute malware directly to victims; or from mobile malware, which is malware that targets mobile devices. The most recommended types of malwares are

A. Malware viruses [18] A malware virus is a type of malware that usually takes the form of code inserted into an application, program, or system and distributed by the victim. Most types of malwares are similar to viruses in that they require a host (e.g. A device) to survive. They remain dormant until an attack occurs, perhaps resulting from a user downloading an email link (usually an .exe file, meaning an "executable" file). From there, the virus multiplies and spreads copies of itself from one computer to another, causing maximum damage. Finally, malware can: • Capture applications. • Send infected files to individuals • Steal data • Eliminate DDoS attacks • Attack ransomware software attacks. Example: iloveyou virus, 2000.

B. Trojan malware [18] A Trojan is a type of malware that disguises itself as real software, application, or file, causing users to download their devices and lose control. Once installed, trojans can do what they are designed to do, such as damage, destroy, steal, or cause other problems with your data or network. Trojan malware, also known as Trojan horses or Trojan horse viruses, is often sent via email links, downloaded websites, or direct messages. Like viruses, these require user action. Ultimately, trojan malware can: • • • • • Delete, modify, or steal data Spy on users Access networks Launch DDoS attacks Take remote control of devices Examples: Zeus/Zbot, 2011: This banking Trojan leveraged keystroke logging to steal credentials and also account balances.

C. Ransomware [18] As the name suggests, ransomware is malware that comes with a ransom. It locks and encrypts the victim's device or data and demands a ransom to regain access. How does ransomware occur? This is often because victims accidentally download malware via email links or links from unknown sources. Once installed, the malware can create a backdoor that allows hackers to access the device and then begin encrypting data, locking the entire owner out of the device until they pay a ransom to regain membership. It is worth noting that ransomware is increasingly being paid in crypto currency, sometimes called crypto-malware. • Ultimately, ransomware can: Hold devices hostage • • Make data inaccessible through encryption Result in financial loss Examples: WannaCry, 2017: this ransomware attack targeted thousands of computers running the windows operating system worldwide and spread to corporate partners worldwide. Victims are asked to pay a bitcoin fee to get their data back.

D. Bots or botnets [18] Other times, a bot can be like a "spider"; this could mean a program that scans the internet to find when hacking is happening and where security infrastructure is being used, or if you prefer, robots doing this. A botnet is a type of malware that enters a device through malicious code. In some cases, botnets attack devices directly and cybercriminals can control devices remotely. Finally, botnets or botnets can: • Launch DDoS attacks • Record activities including keystrokes, webcams, and screenshots • Email through your device can send phishing emails • Hackers can use device remotely Examples: Mirai, 2016: This botnet attack targeted Internet of Things devices and, from there, leveraged DDoS attacks.

E. Adware malware As the name suggests, adware is malware associated with advertisements. Adware, also known as adware, displays unwanted advertisements on your computer, sometimes in the form of pop-up ads that monitor the user's browsing experience. This is sometimes for work. Where adware can go wrong is that these ads damage your information; whether it's selling your information to a third party or using it for theft or credit card fraud. Mobile adware, meaning adware on mobile devices is becoming more common and can spread through third-party app downloads. Finally, adware can: • be annoying • redirect users to malicious websites • install spyware • share user information with us. For example: fireball, 2017: this adware infected approximately 250 million devices by tracking victims' online gaming through browser hijacking.

F. Spyware [18] Spyware is a type of malware that enters a device without the owner's knowledge. This is typically done to monitor online activity, track logins and passwords, or collect sensitive information that could be used for fraudulent purposes. This is also a type of malware; because adware, Trojan malware, and tracking cookies can be considered a type of spyware. Keyloggers are also a popular type of spyware that captures information you type by tracking and recording the keys you type on your keyboard. Finally, spyware can: • invade a person's privacy • collect confidential information, including recording keystrokes • steal information • cause theft or credit card fraud Example: dark hotel, 2014: this key logging spyware targets government and corporate executives using hotel wi-fi.

G. Fileless malware [18] Fileless malware is a type of malware that uses software, applications, and processes built into or built into the operating device to maliciously load and run. In other words, no files are needed to download this type of malware, hence the name Fileless malware. Fileless malware is memory-based, not data-based. Once installed, Fileless malware runs legitimate scripts by default while legitimate processes are in progress. Because of this stealth nature, Fileless malware is difficult to detect. Finally, Fileless malware can: • break antivirus software • steal data Example: astaroth, 2019: this Fileless malware is a real data stealer and mostly attacks windows devices and targets specific countries including Brazil.

Figure 1.2: malware Types.

## II. LITERATURE SURVEY

Malware Detection and Classification Methods Using Machine Learning Algorithms or Heuristics to Detect and Classify Malware, The Features of The Detected Malware Are Classified. We Can Divide Malware Detection and Classification into Five Different Categories: Signature, Behavioral, Heuristic, Model Testing, And Deep Learning. This Number May Increase Depending on The Environment and Technology Used. All Methods and Related Research Materials Are Explained in Detail Below: 1) Signature-Based Malware Detection and Classification: A Signature Is a Bit That Indicates The Structure Of The Program. Since Each Signature Program Is Unique, They Are Often Used for Malware Detection [60]. During The Signature Extraction Process, Static Properties Are First Identified from The Executable File. The Generated Signature Engine Then Uses the Extracted Object to Generate the Signature. Finally, Sign the Document.

When A Suspicious File Needs to Be Marked as Malicious or Malicious, The Signature Of The File Is Extracted As Before And Compared With The Previous Signature. As A Comparison, The Data Structure Is Marked as Bad or Malicious. This Detection Method Is Called Signature-Based Malware Detection. This Detection Method Is Very Fast and Does A Good Job Of Detecting Known Malware. However, It Cannot Detect Zero-Day Malware. Also, According To Scott [61], Signature-Based Malware Detection Is Dead Because It Cannot Detect New Malware, Is Unmeasurable, And Must Rely On Human Interaction From The Affected Party. An Automatic Signature Extraction Method Was Proposed By Griffin Et Al. [62]. The Proposed Method Automatically Extracts Signature Strings Using A Set Of Library Analysis And Various Types Of Heuristics. According To the Paper, Signatures Mostly Appear in Malware Files. Therefore, False Positives Are Reduced. Tang

Et Al. [63] Defined Simple Signatures for Detection Of Polymorphic Genes Using Bioinformatics Techniques. The Proposed Process Is A Negative-Based Signature And Consists Of Three Stages: Identify The Most Significant After Using The Alignment Process, Remove The Background Noise And Simplify The Signature Instructions, Which Is Compatible With Ids (Intrusion Detection Systems). Liu And Sandhu [64] Proposed a Fingerprint-Based Signature Design To Detect Malware In Hardware. The Scheme Works as A Tamper-Proof System and Generates Different Names Based On Cryptographic Hashes. Using These Signatures, Trojans Embedded in Hardware Can Be Detected. Determine Whether the Sample Program Is Malicious or Benign Based on The Written Behavior. This Method Has Three Parts: Extracting the Behavior, Generating the Behavior, And Using Machine Learning Algorithms to Determine Whether the Analysis Is Malicious or Malicious. The Behavior Is Determined by Phone Calls, Api Calls, Or Changes to Data, Access, And Computer Programs. In Other Words, Behavior Is Determined by Analyzing the Order or Frequency of The Call and The Data Logging Process. The Group Acts Sequentially and Uses This System to Achieve Something. Although The Source Code of The Program Changes Over Time, The Behavior of The Program Does Not Change At All. Therefore, Many Types of Malwares Can Be Detected Using This Method. In Addition, New and Previously Unknown Malware Can Be Discovered with This Method. The Biggest Disadvantage of Behavioral Analysis Is That It Does Not Show All the Real Behavior Of Malware In Protected Environments Such As Virtual Machines And Sandboxes. Kolbitsch Et Al. [65]. The Calls Are Converted into A Graph, With Each Node Representing A Call And The Edges Representing A Change In The Call. Use The Results of The Call As Input To The Other System To Determine The Connection Of The Call. The Image Of The Program Should Be Extracted And Compared With Existing Images. For Comparison, The Sample Files Are Labeled As Malware Or Malicious Files. Also, New Behavior Found During Analysis Is Dynamically Added to The Image. Create Feature Vectors Using Attributes And Classifications Made By Ml Algorithms. Singh Et Al. [66] Used Multiple Behavior-Based API Calls To Detect Malware. In This Approach, Various API Implementations Were Developed Using Depth-First Search And N-Grams. Dice Coefficient, Cosine Coefficient And Tversky Index Are Used To Determine The Similarity Of The Software Body And Identify Multiple Apis Simultaneously. Use Machine Learning Algorithms To Identify System Designs. Aslan Et Al. [67] Proposed A Scbm Behavioral Model To Detect Malware. The Information Collected Is Related To Sample Analysis. During Feature Extraction, System Paths And Attributes Are Taken Into Account. This Ensures That Bad Behavior Patterns Are Different From Positive Behavior Patterns. According To The Paper, The Proposed Model Has Fewer Features Than N-Grams And Other Methods In The Literature. Test Results Show That The Proposed Model Can Work Well And Detect Malware Based On Dr, Fpr, F-Score And Accuracy Rate. [68] Proposed A New Hybrid Method Based On Dynamic Analysis Using Cyber Threat Intelligence, Machine Learning And Data Forensics. The Paper States That The Concept Of Big Data Forensics Is Used To Estimate The Reputation Of Intellectual Property At The Pre-Acceptance Stage. Then, Potential Zero-Day Attacks Are Isolated Using A Decision Tree Algorithm For Behavioral Analysis. The Proposed Method Is Evaluated In Terms Of F-Measure, Precision And Recall Scores. According To The Test Results, The Obtained F-Measure, Precision And Recall Scores Are Quite Satisfactory Compared To Other Methods In The Literature. A New Malware Behavior Detection Method Called APTMalInsight Is Proposed In [69]. The Scheme Relies on Call Data And Ontology-Based Knowledge To Identify And Recognize Advanced Persistent Threats (Apts). The Paper Stated That in Terms Of The Acquired Feature Vectors, Apt Malware Can Be Detected And Combined At A High Rate.

In Research [92], A Permission-Based Android Malware Detection Framework Was Proposed. Based On Different Horizontal Malware Types, Permissions Are Requested from Two Distributions Based on Android Malware Detection. These Classifications Are Compared with Simple Machine Learning Methods Such as Support Vector Machine, K Nearest Neighbors, Naive Bayes And Decision Trees On Four Different Datasets. In Addition, Bagging, One of The Hybrid Learning Methods, Is Used To Create Different Distributions To Increase The Classification Efficiency. Therefore, Classification Algorithms Based on Linear Regression Models Can Achieve High Performance Without the Need For Highly Complex Classification Algorithms. The Author’s in [93] also differs from state-of-the-art solutions by being non-invasive, since it leverages a process to obtain application’s features that does not infringe licenses and terms of use of applications. In addition, according to experiments performed on a real Android smartphone, our proposal presents the following additional advantages over state-of-the-art solutions: a more efficient process to classify applications that is three times faster and requires ten times less CPU usage in some cases (saving device energy); and a better detection performance, with higher balanced accuracy, nine times less false positive cases, and ten times less false negative cases.

The paper [94] investigates malware classification accuracy using static methods for malware detection based on LightGBM. We extracted the dataset features from PE-file surface analysis and PE-header dumps and customized a binary log loss function to improve all the classification evaluation metrics to a certain extent. We obtained a better result (AUC = 0.979) at α=430 and β=339 than the normal log loss function (AUC = 0.978) on the EMBER dataset. Authors in [95] propose the MalFSM framework. Through the feature selection method, we reduce the 735 opcode features contained in the Kaggle dataset to 16, and then fuse on metadata features (count of file lines and file size) for a total of 18 features, and find that the machine learning classification is efficient and high accuracy. We analyzed the correlation between the opcode features of malicious samples and interpreted the selected features. Our comprehensive experiments show that the highest classification accuracy of MalFSM can reach up to 98.6% and the classification time is only 7.76 s on the Kaggle malware dataset of Microsoft. This paper [96] proposes a machine learning model based on the co-existence of static features for Android malware detection. The new datasets were extracted using Android APK samples from the Drebin, Malgenome and MalDroid2020 datasets. The maximum accuracy, which is 98%, was achieved using the Random Forest algorithm and the co-existence of permissions features at the second combination level. In addition, the experiments show that using the Drebin dataset, the proposed approach achieved an accuracy of about 95%, while the state-of-the-art achieved an accuracy of about 93%. This paper [97] proposes a machine learning model based on the co-existence of static features for Android malware detection. The proposed model assumes that Android malware requests an abnormal set of co-existed permissions and APIs in comparing to those requested by benign applications. To prove this assumption, the paper created a new dataset of co existed permissions and API calls at different levels of combinations, which are the second level, the third level, the fourth level and the fifth level. The extracted datasets of co-existed features at different levels were applied on permissions only, APIs only, permissions and APIs, and APIs and APIs frequencies. To extract the most relevant co-existed features, the frequent pattern growth (FP-growth) algorithm, which is an association rule mining technique, was used. The new datasets were extracted using Android APK samples from the Drebin, Malgenome and MalDroid2020 datasets. To evaluate the proposed model, several conventional machine learning algorithms were used. The results show that the model can successfully classify Android malware with a high accuracy using machine learning algorithms and the co-existence of features. Moreover, the results show that the achieved classification accuracy depends on the classifier and the type of co-existed features. The maximum accuracy, which is 98%, was achieved using the Random Forest algorithm and the co-existence of permissions features at the second combination level. This study [108] explores the application of Random Forests, a machine learning algorithm, in the field of cybersecurity. Specifically, it investigates the effectiveness of Random Forests in malware detection and intrusion detection. Through experiments conducted on relevant datasets, the study demonstrates the robust performance of Random Forests in accurately classifying malware samples and detecting various types of network attacks. The interpretability of Random Forests also provides valuable insights for security analysts to understand the indicators and behavioral patterns of malware and attacks. In this research, we propose a behaviourally based approach to identifying malware. Since new malware families and variations are constantly being found on the internet and dark web, they constitute a particularly severe danger. This approach [109] yields prediction algorithms that can analyze the actions of malware to unearth new variations and families. There are a total of 3540 rows describing features of corrupted files and 6999 describing properties of uncorrupted files in the dataset. To achieve this, I used the 98.19% accurate Random Forest machine learning method and the 96.77% accurate KKN machine learning algorithm.

## III. PROPOSED WORK

We propose a method for generating feature corresponding to each malware and benign software using strong correlated features. The main approach is to select those features only which have co-rrelation values greater than 0.5 for subsequent model training and classification. We used the matrix values based on relational index and will use convolutional neural network designed to recognise the high-dimensional features and experimentally demonstrated the feasibility of the scheme. We propose a neural network model based on sequential networks for the classification task of malware detection and experimentally demonstrate the high accuracy of our malware detection model. In this work, we extract features from the existing 215 features which are highly relevant for malware detection and classification. Finally, the selected features are experimentally analysed in reconstructing the high-dimensional features of the malware performance. We designed a sequential neural network and other machine learning classifiers based to perform the classification and detection task for malware. And experiments were conducted using datasets we collected from Derbin-215. The experimental results show that our method is more accurate than traditional machine learning methods for malware detection models based on new selected features. In the feature extraction phase for malware detection, used to extract the corresponding features of the software in addition to extracting the corresponding static feature information, such as API calls, permission information, etc, and dynamic feature information, such as network activity, log _les, etc. This feature extraction solution is more automated and simpler than other manual feature extraction methods. Automatic extraction of software features using correlation values requires consideration of the data representation. So that it can better extract the key features and ensure the accuracy of the test results. We propose a approach to malware detection, which is designed based on the sequential neural network. The Figure 4.1 below illustrates the overall structure and main tasks of our malware detection method. First, benign files and malware are transformed into corresponding dataset. Afterwards, the new features are created using matrix model and are passed through data sampling techniques to balance the dataset before training the models. The detailed design process will be described in the subsequent sections.

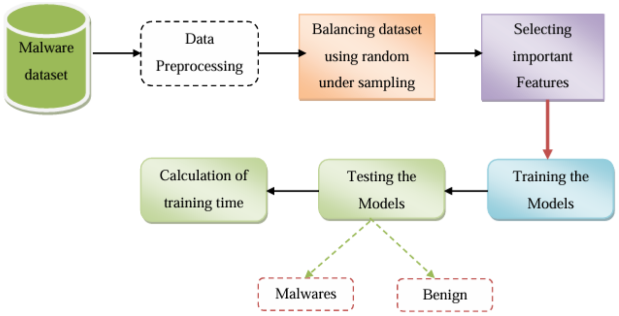

Malware dataset
Data Preprocessing
Balancing dataset using random under sampling
Selecting important Features
Training the Models
Testing the Models
Calculation of training time
Malwares
Benign

Figure 3.1: Proposed work.

### 3.1Data Preprocessing

Selecting Important Features Training the Models At the initial stage, data preprocessing is performed to improve the quality of the actual data. As regards classification, it is vital for choosing features to signify the class the new record would concern. The main task of the Pre-processing of feature data phase is to provide an input data for the neural network model. The advantages of using this method are twofold. Firstly, this method of extracting software features using highly correlated features will increase the detection accuracy and second it will reduce the training time by reducing the overfitting problems of the classifiers. Following tasks are done for cleaning the dataset for proper training. • Convert non-numeric columns to numeric. • Replace missing values with NaN. • Replace '?' with NaN before conversion. • Perform Label Encoding on Class type (0 for B and 1 for S). • Drop rows with values NaN.

Count the occurrences of each class. Calculate the majority and minority class labels. • Separate majority and minority class samples. Oversample the minority class matches the majority class. • Combine the oversampled minority class with the majority class. Shuffle the balanced dataset. • Now, balanced data contains the balanced dataset with equal instances of both classes. The difference between our work and previous work [97] is that we try to extract the important features only required and are highly correlated of the methods. Analysing the feasibility of such a scheme is a major part of our research. The redundant information causes high pre-processing overhead and reduces the accuracy and increases the training time of the model classification at the later stage. We try to reduce the training time and increase the accuracy of the models.

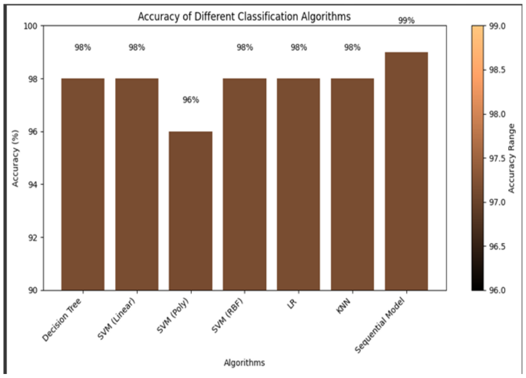
4.3 Feature Selection • Select Features with correlation greater than 0.5. • Divide the data into train and test set. All features (with correlation greater than 0.5.) are set to X and the target class is set to y. • Divide the dataset with 80:20 ratios.

## IV. RESULTS

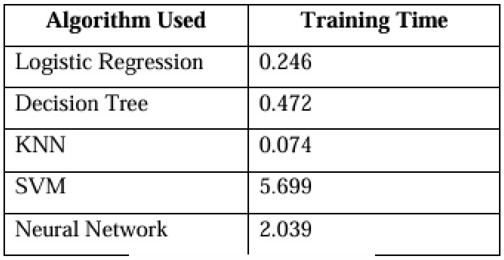
Android malware has continued to grow in volume and complexity posing significant threats to the security of mobile devices and the services they enable. This has prompted increasing interest in employing machine learning to improve Android malware detection. In this thesis, we present a novel classifier approach based on a multiple feature selection concepts that enables effective combination of machine learning algorithms for improved accuracy. The framework generates a model by training base classifiers by using Android Malware dataset named “Drebin dataset for malwares”.

The most dangerous side of this point is the end user's lack of cyber security awareness, unconscious, and carelessness. So, it is very important to take countermeasures to secure these devices and deal with such malicious software before it is presented to the end user. When evaluated in this context, the detection of the huge number of malware that targeted Android phones has led to the need to work in both academic and industrial fields in order to tackle this situation. For this reason, it has become widespread to detect malicious software in Android systems using artificial intelligence applications. This study aims to detect malware based on artificial intelligence methods and using DREBIN [107] well-known benchmark Android malware dataset.

Drebin-215 consists of vectors of 215 features from 15036 app samples; of these, 9476 were benign samples while the remaining 5560 were malware samples from the Drebin project [2]. The Drebin samples are also publicly available and widely used in the research community.



The table 5.1below shows the training time of the algorithms used in this research.



| Algorithm Used | Training Time |
|---|---|
| Logistic Regression | 0.246 |
| Decision Tree | 0.472 |
| KNN | 0.074 |
| SVM | 5.699 |
| Neural Network | 2.039 |

Table 5.1: Training Time.

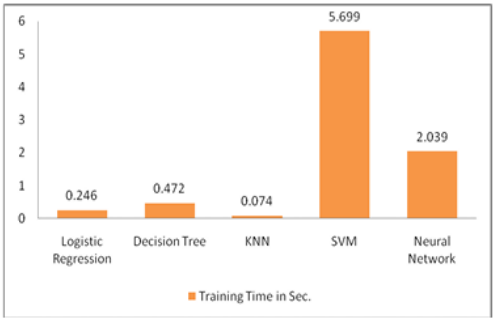


Figure 5.30: Training Time Comparison.

Figure 5.30 shows that the training time for SVM is highest and for KNN is lowest. So we can conclude that KNN classifiers train the models fastly.

## V. CONCLUSION

SS Malicious software (Malware) is one of the foremost threats on the Internet today. Many problems of data security arise due to the unstopping propagation of malware. In the past few years, several techniques have been developed for the in-depth inspection of malware, ranging from static code review to dynamic exploration of malware behavior. However, the previous work lacked a complete system for the timely detection of malware combining analysis and detection along with signature and machine learning-based detection engines. Therefore, it was much required to present architecture to detect both existing as well as novel malware in the network. Keeping that in view, we presented well-integrated architecture of an Anti-Malware System (AMS). We conducted the experiments using the Drebin-215 and Figshare dataset. Two state-of-the art models were used as discriminators. The accuracies were enhanced by a good percentage for the models.

We plan to conduct experiments with more datasets, such as Malimg and MaleVis and other datasets. Expanding our research endeavours in this direction involves delving into several potential research areas. A crucial aspect is the comprehensive assessment of the AMS’s performance across diverse datasets to guarantee its adaptability to various malware families and their distinct variations. This evaluation should encompass datasets featuring a spectrum of file formats, sizes, and obfuscation techniques.