JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# MALWARE DETECTION USING MACHINE LEARNING

**PRIYA. N**
**LECTURER,**
**DEPARTMENT OF CSE,**
**GOVERNMENT POLYTECHNIC IMMADIHALLI,BANGALORE,INDIA**

**Abstract :** The sheer quantity of data and files that need to be analysed for potential threats is the greatest challenge for malware researchers. On a daily basis, researchers find and analyse a significant number of new malwares in order to extract common characteristics. Therefore, it is essential for the investigation of malware characteristics to have a scheme that can ensure and improve the efficacy and accuracy of categorization. This paper proposes a very efficient automated classification method that combines multi-feature selection and machine learning-based fusion. The system demonstrates great performance. Based on our evaluations, it exhibits superior performance and functionality compared to single-featured devices. The proliferation of malware poses a substantial threat as its use continues to expand. Manual heuristic malware assessment is no longer deemed efficient owing to the fast propagation of malware. Therefore, the use of machine learning algorithms to automatically identify and analyse malicious software behaviour is considered a very effective option. Behaviour reports will definitely be generated after the analysis of each malware's behaviour in a simulated environment. It is necessary to pre-process such data by converting them into sparse trajectory models before using machine learning techniques, specifically for classification purposes. The research used Support Vector Machines (SVM) and Random Forests as classifiers. Ultimately, using autonomous behaviour and machine learning techniques may effectively and efficiently detect and classify malware, as shown by this proof of concept. Due to the substantial negative impact that many antivirus programmes have on the user's system and their occasional inability to detect new and hazardous threats, lightweight antivirus programmes are less efficient in discovering and preventing malware. Our approach uses a cloud-based malware classification algorithm, with the main focus being on safeguarding data while minimising any noticeable impact on the user's system. Consequently, we can assess and ascertain if any dubious file is malicious or not without the need to install additional third-party software on user PCs.

## 1. INTRODUCTION

The increasing prevalence of malware tactics poses a significant threat to the security of personal information and contemporary information technology, making anti-malware technology more essential than ever. Despite efforts made by anti-malware solution providers to minimise the harm, a ransomware epidemic called Ransomware successfully infiltrated several schools, commercial enterprises, and government infrastructures. Nevertheless, in order to evade detection, many modifications were developed, and new instances of widespread infections were carried out using innovative methods. The proliferation of several Ransomware strains showed the ability to bypass conventional security measures using contemporary methods. Change and misunderstanding hinder the detection of reality by malware hardware designers, so significantly adding to the increase in the number of malwares. According to McAfee's study, researchers are required to devise a method for eliminating these newly discovered malwares by categorising them based on their family size. The malware species refers to a collection of malevolent organisations that have similar objectives but use distinct methods to accomplish them. Nevertheless, the act of manipulating and obscuring data has made it difficult for scholars to thoroughly examine and classify the information. Hence, it would be advantageous to assess samples with similar attributes, as this might streamline the identification of their potential characteristics and save the researcher's time. Hence, the classification of malware has significance in the process of investigating malware. Before the rapid advancement of machine learning, human methods that relied on signature-based categorization were often required to categorise data. Additionally, the effectiveness of these approaches is greatly impacted by obfuscation and polymorphism schemes. The proliferation of malware, including viruses, worms, Trojan horses, root kits, bonnets, backdoors, and other hazardous software, is rapidly increasing. Traditional antivirus solutions that rely on signature matching are unable to detect polymorphic and previously unknown dangerous executable files. The Internet is facilitating the widespread dissemination of malware worldwide, which is progressively increasing in its level of threat on a daily basis.

1.1 **MALWARE**: Malware, often known as malicious software, is any programme or file that causes harm to a computer user. Malware scanning may detect computer worms, Trojan horses, viruses, and spyware. These malevolent software possess the capability to pilfer, encrypt, or erase sensitive data, manipulate or seize control of vital computing operations, and surreptitiously monitor user computer activities. Malware creators use several physical and virtual methods to distribute malicious software that infects and compromises devices and networks. For instance, malicious software may be introduced to a computer via a USB, and it can also propagate online through drive-by downloads, which inadvertently and automatically transmit harmful software to computers. Phishing is a prevalent technique for disseminating malware, whereby emails that masquerade as authentic messages include malicious files or links that may infect unwary receivers with the malware executable. Advanced malware operations often use a command and control server. It allows malicious individuals to establish communication with compromised devices, search through confidential information, and even remotely control the compromised server or device.

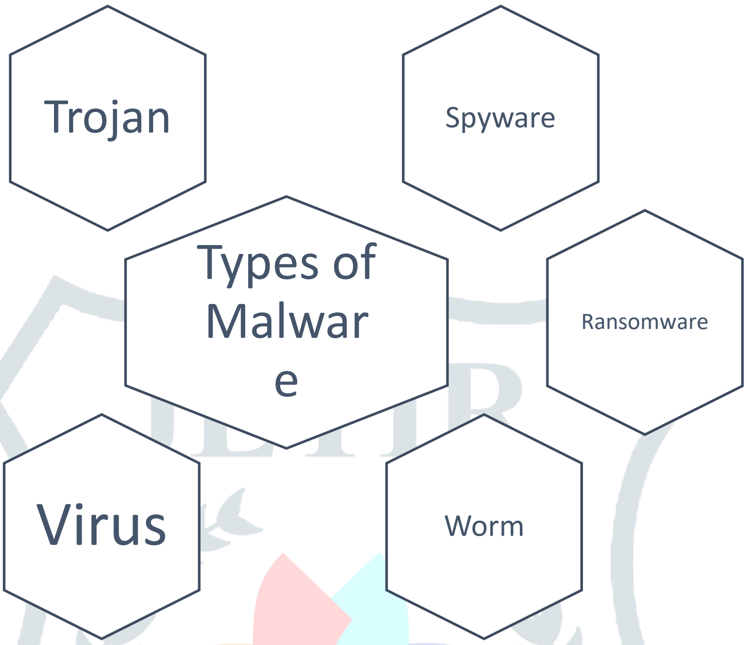


**Figure 1.1 Types of Malwares**

1.2 **MALWARE DETECTION:** Malware is becoming more and more common, which poses a serious danger to the digital world. Various security measures, such as Anti-Virus (AV) techniques, have been developed and novel approaches have been explored to control and minimise the harm caused by malware. These AV approaches may be classified into two primary categories: signature-based and non-signature-based. Scanning approach is employed by security software that depends on signatures. Verify the signatures in dubious files (a certain byte sequence). This approach is very effective in detecting and mitigating known malware, but it is highly ineffective in dealing with "zero-day" (Bilge and Dumitras, 2019) and "unknown" threats. Additionally, it is very efficient (Hodgson, 2005; Murugan and Kuppusamy, 2011; Kumar and Pant, 2021). Attackers may have more opportunities to strategize their assaults due to the limitations of signature-based techniques, which are restricted by signature databases and the arduous and intricate process of forging signatures. The attacker's likelihood of success in the operation will be much diminished if they use established tools and processes (Potter and Day, 2021). Christodorescu and Jha (2004) state that cybercriminals are developing novel anti-malware evasion tactics in order to effectively carry out cybercrimes. Identifying malware is a vital and complex phase in an anti-malware approach. The anti-virus industry uses many malware recognition methods that have been researched in academic literature.

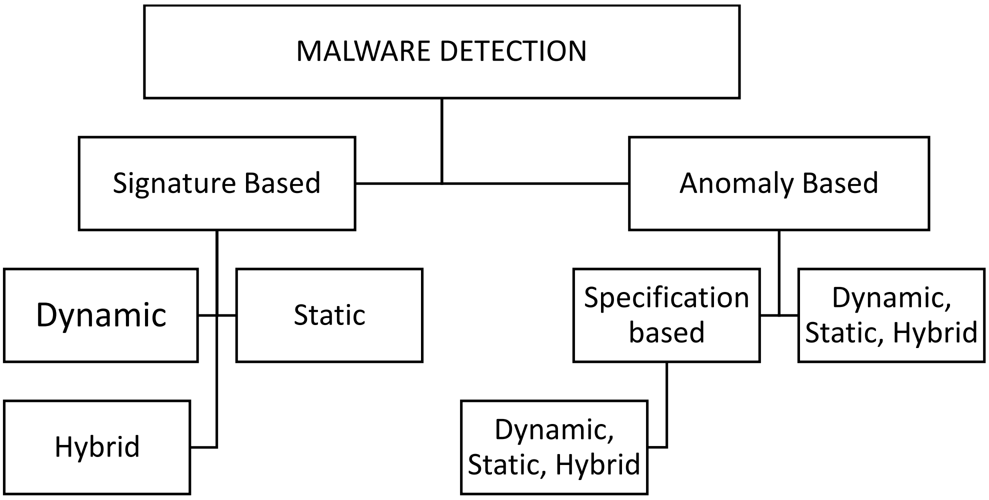


**Figure 1.2: Types of Malware Detection**

### 1.2.1 Signature-based Finding

When using signature-based detection for identifying malicious applications, the signature is regarded as an incidental outcome. A signature is composed of bytes retrieved from pre-existing malware. Dynamic signature-based detection differs from static signature-based detection by running the software in a secure environment and simultaneously confirming the signature. Signature-based detection is ineffective when dealing with "zero-day" and "unknown" malware, but it performs better when dealing with known malware. A signature-based detection system is limited to the signature database and requires regular updates of newly created signatures. It also requires storage capacity that is proportional to the number of signatures kept on the end host.

### 1.2.2 Anomaly-based detection

Anomaly-based detection may overcome the constraints of signature-based detection by using a non-signature-based approach. Anomaly-based threat detection does not use malware-specific signatures. Anomaly detection employs a standard profile, and any deviation from it is presumed to be malicious. This study primarily focuses on using machine learning algorithms to identify malware via static analysis, which is a kind of anomaly-based detection. Machine Learning (ML) based malware detectors mostly depend on features (Yan et al., 2013b). The objective of this undertaking is to create and provide novel feature sets that may improve the effectiveness of malware detection. This research use machine learning methods to detect counterfeit Android and Portable Executable (PE) applications by using four distinct sets of attributes. Utilise a rudimentary, expeditious, and effective stationary analysis methodology that you have acquired for the purpose of extracting features. This strategy will result in significant time and energy savings. The thesis addresses the study issue statement and the corresponding research questions in its last part, 1.3.

## 2. PROBLEM STATEMENT

Malware detectors that rely on signatures are effective in detecting known malware that has been previously identified by certain anti-virus companies. However, it lacks the ability to identify newly created malware that has not yet undergone notarization, as well as polymorphic malware that can alter its signatures. Due to the often inadequate accuracy of heuristics-based detectors, there is a high occurrence of false-positive and false-negative outcomes. The need for improved detection is driven by the rapid spread of polymorphic viral transmission. Lightweight antivirus programmes are less efficient in detecting and preventing malware due to the huge negative impact many antivirus programmes have on the user's system and their limited contribution to identifying serious threats. Our malware classification technique prioritises minimising the impact on data protection inside the system. Thus, we can do a thorough examination to determine whether any dubious files are malicious, without requiring the consumer to install any extra software on their own computer.

## 3. LITERATURE SURVEY

This section utilises machine learning methods and static properties to strengthen past efforts in malware identity. Each of the sections focus on the arrangement and display of materials related to the taxonomy of Android applications and malicious Portable Executable (PE) files. This study may help us get a better understanding of the limitations and potential for improvement of current techniques and methods. The inquiry is focused on the static properties that have been previously reported, which include the use of machine learning for malware identification. Prior to exploring additional static properties for classifying Android and PE programmes, these feature sets included the whole of the Portable Executable (PE) header field and the permissions of Android apps.

Feature extraction is the process of computing or extracting values from samples in a dataset. For example, feature extraction in the Information Retrieval (IR) field is simple and fundamental, in contrast to the organised and organised attributes used in recommendation systems. Malware research commonly use programmes in many forms, including Dynamic Link Library (DLL), Executable (EXE), Component Object Model (COM), and code snippets, as datasets. Both static and dynamic analysis may be used to derive characteristics from these binary programme files. Features obtained via dynamic analysis are referred to as dynamic features, whereas characteristics obtained through static analysis methods are referred to as static features. Feature sets may be categorised into three groups: static (features obtained via static analysis), basic heuristic (features obtained from the PE header or strings obtained from the executable body), and dynamic (features obtained during runtime) (Dai et al., 2021). According to a research by Bilar (2019), detection techniques may be seen as a middle ground between static and dynamic heuristics. They are seen as statistically-based structural fingerprints. Komashinskiy et al. provide a comprehensive taxonomy of characteristics, classifying them as either external or interior. Subsequently, the internal attributes are categorised as either static or dynamic. Within each of these classifications, there are other general attributes such as string and n-gram (Komashinskiy and Kotenko, 2020). Yen et al. provide these distinctive feature investigations for utilisation in machine learning-based malware classifiers. In order to categorise malware families, researchers have examined several sets of characteristics, such as byte-n-gram, opcode-n-gram, PE header fields, and dynamic trace. They showcased the optimal method for determining the suitable feature set and amount of characteristics by sharing their findings. Based on the study conducted by Yan et al. (2019a), Decision Tree (DT) has been shown to have the highest accuracy while using the fewest features. It consistently outperforms or performs equally well as Support Vector Machine in all aspects. The parts that follow identify and define popular characteristics, together with relevant current works, in the areas of static (section 2.2.4), dynamic (section 2.2.6), and hybrid (section 2.2.5).

There are three main kinds of structures that are built according to the type of analysis: static, dynamic, and hybrid. In this discussion, we will examine the several categories of static features and their operational mechanisms, while referencing relevant literature sources. Deleting individual static topographies inside the executable file is unnecessary. Reverse engineering is often used to extract static data from binary files. The two most prevalent techniques for pre-processing and extracting static attributes as a representative sample are disassembling and performing hexadecimal dumping of binary files.

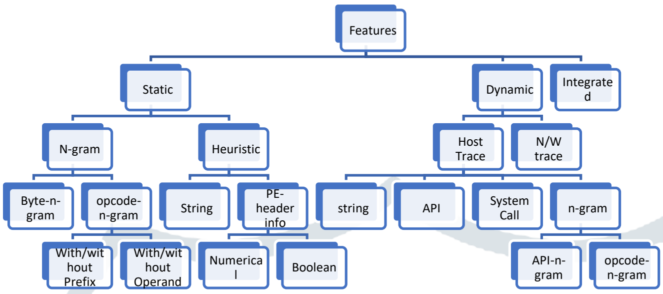


**Figure 3.1 : Machine learning features used for the development of malware classifiers.**

## 3.1 Techniques for analysing malware

**3.1.1 Analysis with dynamic approaches**: Upon execution of the file, information on its properties, including its genuine objectives, is logged. Our file may be executed by using a virtual computer, such as Virtual Box. Through conducting such a study, we may readily ascertain all aspects related to behaviour, such as the ability to identify a file and undo its operations. Methods that use behaviour may be characterised as a combination of static and dynamic analysis. The advantage of using these dynamic methodologies is that we can accurately anticipate the outcomes that will occur after this virus is deployed on an actual computer.

### 3.1.2 Static method analysis

Utilising the source code patterns to uncover the behavioural characteristics of a programme is akin to employing a static approach to analyse malware. Static method analyses are beneficial because they provide a high level of precision and can effectively determine the function and purpose. Common static analysis techniques for malware detection include the following:

#### 3.1.2.1 File Format Inspection

The real objective may often be discerned by examining the file's metadata or structure. Details on aspects such as execution or compilation time and other functionalities may be found, for example, in Windows' portable executable files

#### 3.1.2.2 String Extraction

By analysing the software's output, such as status or error messages, we may get information about the malware's behaviour.

#### 3.1.2.3 The fingerprinting approach

The methodology enables us to calculate intricate entities such as cryptographic hashes and infer hardcoded data such as usernames, file names, and registry strings. Three point one point two point four Performing a virus scan The majority of individuals use antivirus software due to its exceptional ability to identify prevalent types of malware. Sandboxes, which are virtual computers, may use this approach to detect malware.

#### 3.1.2.5. Disassembly method

This is a conventional way for analysing static methods. The objective of this approach is to infer the genuine intention by converting our code into assembly code. A software requirements specification (SRS) encompasses both functional and non-functional requirements, as well as use cases that delineate the anticipated interactions between the product and end users. Software requirement specifications may be used by clients, contractors/suppliers, and other departments in market-driven initiatives, such as marketing and development, to establish clear agreements on the desired and undesired functionalities of a software product. Software prerequisites definition not only provide a strong foundation for evaluating product costs, risks, and timelines, but also enables a comprehensive assessment of requirements before to implementation, hence reducing the need for future redesign. Required software.

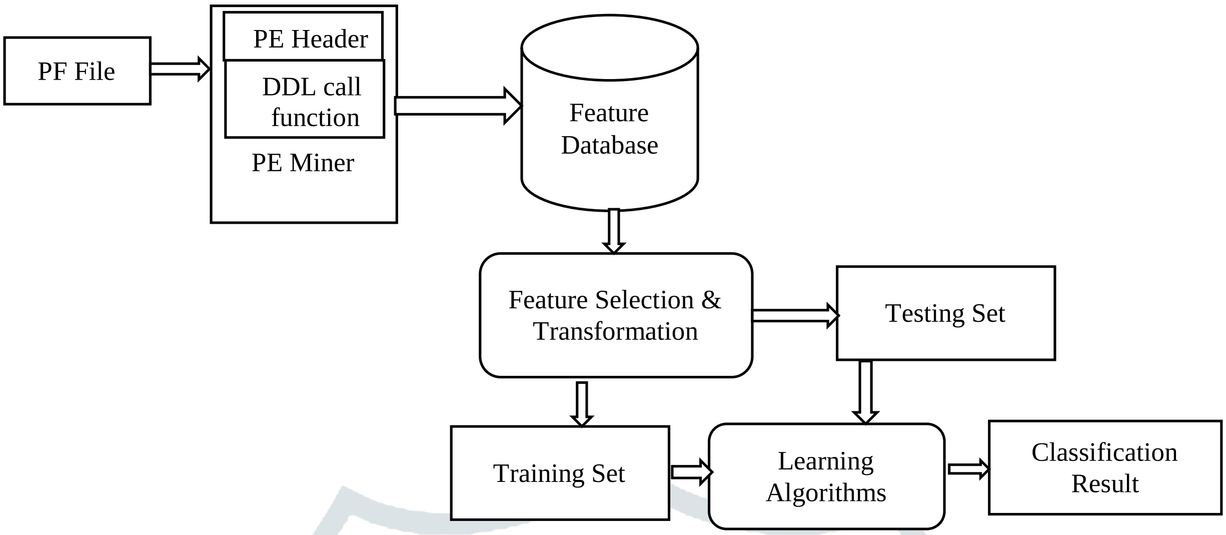


**Figure 3.2 Showcases the architecture designed for the identification and classification of malware.**

The overall flow of the procedure is divided into the following phases based on the architecture:

1) The PE file python library is used to create a data collection. This library helps extract static data from software or programmes. The extracted data is then stored in an Excel file for each software.

2) Data pre-processing Data pre-processing is an essential and crucial step that must be taken prior to feature selection for a model. Eliminate any null values from the dataset throughout the process of data pre-processing. Eliminate the columns that contain category data.

3) Feature Selection Select the attributes that contribute to output prediction or input categorization, or that have a direct correlation with the outcome. After finalising the characteristics that are suitable, divide the dataset into two equal halves. One part will be used for testing and the other for training.

4) Instruction After successfully completing the previous step

### 3.2 Description of the Dataset

The data utilised in this work was sourced from the annual techno-cultural event, Meraz'18, held at IIT Bhilai, including both valid and malicious information. Malicious software, sometimes known as malware, is software designed with the explicit intention of inflicting harm onto a computer system or gaining unauthorised access to it. Authentic files are devoid of any malicious code, making them secure for consumers to use. The files underwent statistical analysis, mostly including the extraction of PE data and the calculation of entropy for various sections of the files.As the competition progresses, more information may be provided to include newly discovered zero-day viruses and assess the reliability of your algorithm. Moreover, engaging in this activity will provide you with an understanding of the challenges that dominant anti-malware software companies, such as Max Secure Software, encounter in their efforts to uphold security.

## 4.PROPOSED METHODOLOGY

The methodology consists of five instructional procedures to discover malware that has damaged the system.

1. **Data Collection**: Data sets are initially retrieved from a file and then saved for future reference. Collecting data may refer to the process of obtaining or measuring information about specific parameters inside a system that already exists, with the purpose of evaluating outcomes and addressing pertinent inquiries.The purpose of data collection is to gather information that can be analysed to get reliable answers to the questions that have been posed.

2. **Data conversion**: The data that was imported in the previous step has now undergone cleansing, normalisation, and modification in order to make it compatible with the algorithm. Data conversion ensures uniformity in terms of range, format, and other relevant factors. At this level, the processes of feature abstraction and collection are also carried out and are further elaborated upon. The data is divided into two sets: a "training dataset" and a "test dataset." The model is constructed using the database from the training set, whereas the evaluation of the model is done using the data from the test set.

3. **Model Training**: At this step, a model is built using the selected technique. The training model refers to the specific dataset that is used to teach a machine learning algorithm. This consists of both the sample output data and the connected sets of input data that influence the outcome. By inputting data into the training model, we may execute the algorithm and evaluate the resulting output by comparing it to the reference output. Discovering the connection results in a revision of the model. In machine learning, the act of instructing an algorithm to determine and acquire the appropriate values for its associated attributes via the input of data is referred to as "model training". While there exists a diverse range of machine learning models, the two most dominant ones are supervised and unsupervised learning.

4. **Model Testing:** The model that was built or validated in the previous step is assessed using test datasets, and the results are used to generate a new model that combines and gains knowledge from previous ones. The phrase "training model" pertains to the procedural processes used in computer programming to instruct a machine learning system on how to identify and utilise appropriate criteria for all of its attributes. Supervised and unsupervised learning are widely used methods for constructing machine learning models.

5. **Deployment of the Model:** The optimal prototype is chosen at the moment (either after a certain number of repeats or as soon as the expected outcome is achieved). The core essence of model deployment is the integration of a machine learning model into an established production environment to enable the model to take inputs and generate outputs. One motivation for deploying your model is to enable other systems, users, or management to access the predictions generated by a trained machine learning model.

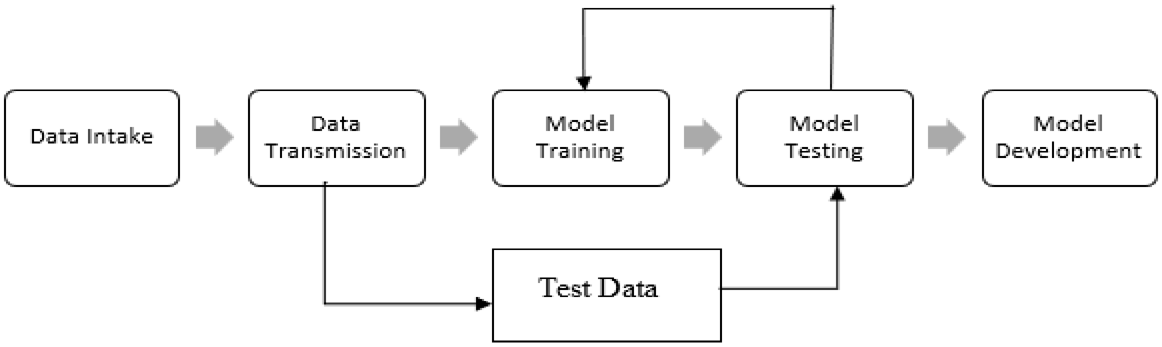


**Figure 4.1 Process of Work Flow**

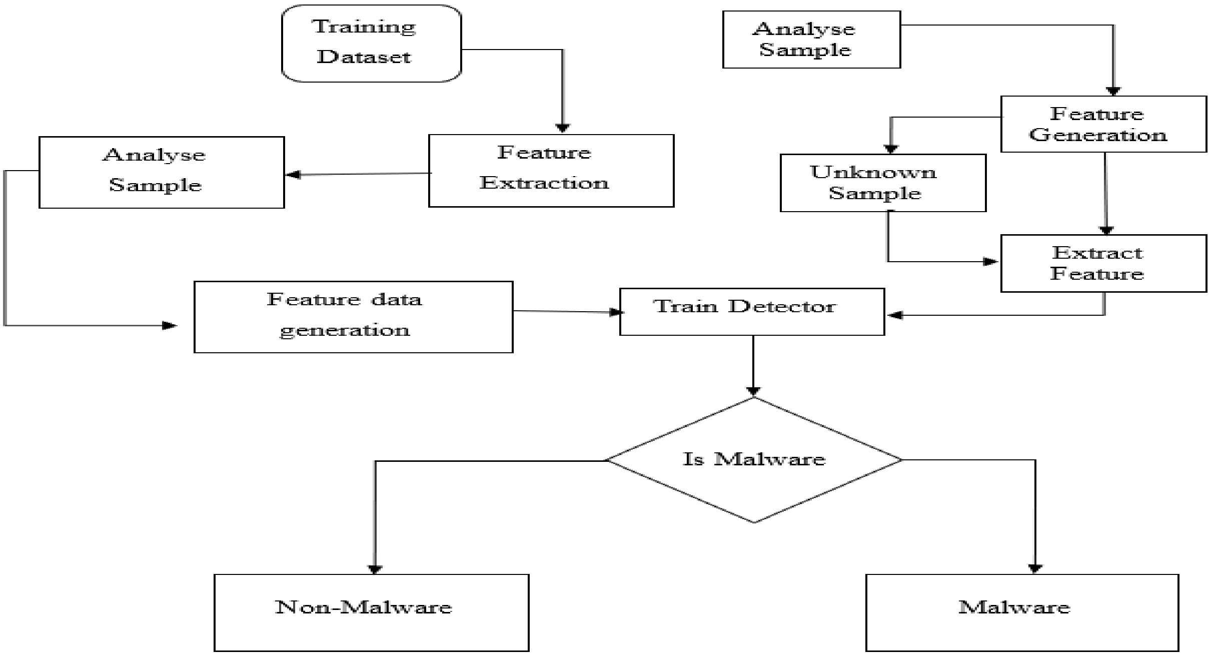


**Figure 4.2 presents a proposed technique for the detection and classification of malware.**

## 5. RESULTS AND DISCUSSION

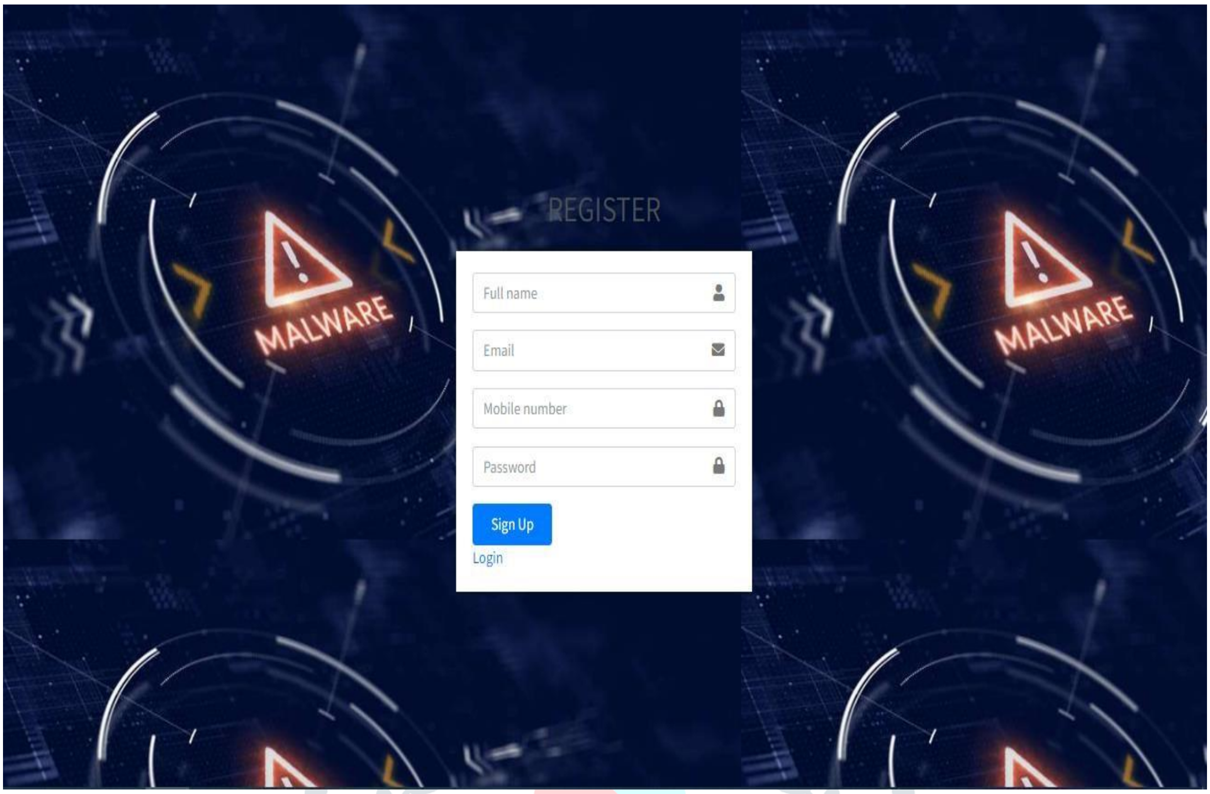
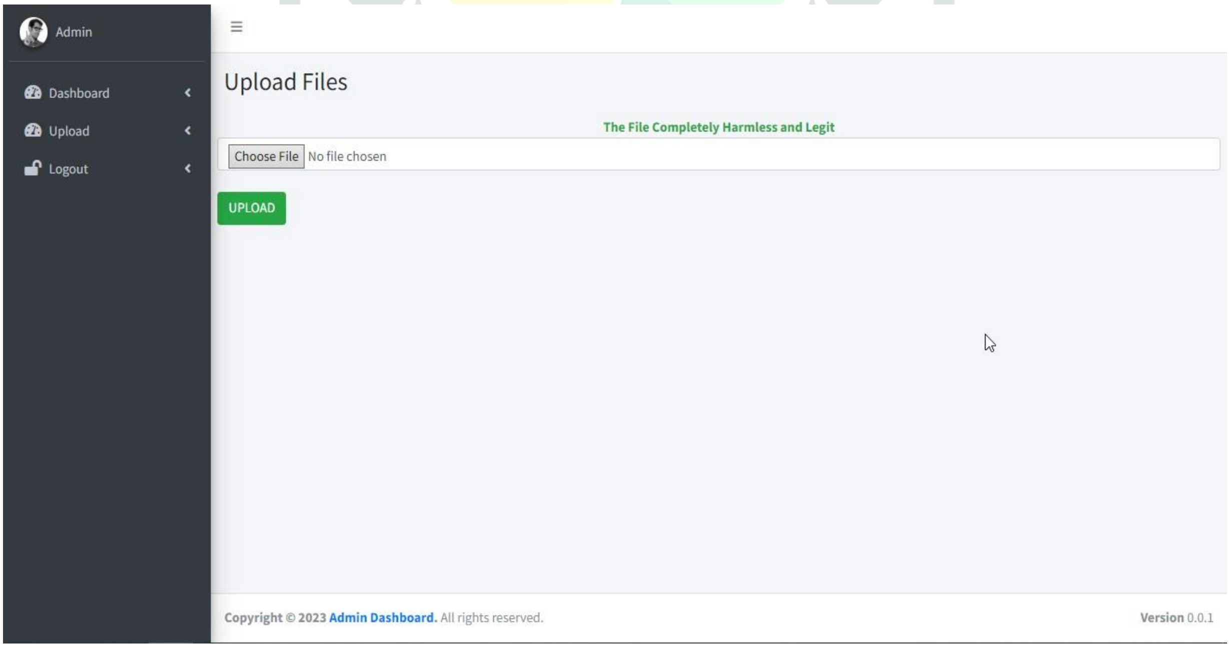


**Figure 5.1: Login Form**



Upload Files

The File Completely Harmless and Legit

Copyright © 2023 Admin Dashboard. All rights reserved.

Version 0.0.1

**Figure 5.2: Upload Files**

Based on our analysis, the use of various feature selection methods results in a diverse amount of features and varying accuracy for distinct models. It seems that one model achieves the highest level of accuracy for validation, but it does not get the highest level of accuracy for testing. Nevertheless, the strategy of limiting features recursively in the same model yields the highest accuracy in both validation and testing. The chosen method for this approach is Random Forest. Consequently, we have determined that the model is the most suitable choice for analysis, particularly for these datasets. To enhance the work, a more sophisticated model may be developed for the multi-class classifier.In order to enhance its ability to categorise each instance of malicious software.

# REFERENCE

[1] Parmjit Kaur, Sumit Sharma, “Google Android A Mobile Platform: A Review. ” In Recent Advances in Engineering and Computational Sciences (RAECS), 2020, pp. 1-5. IEEE, 2020.

[2] Egele, Manuel, Theodoor Scholte, Engin Kirda, and Christopher Kruegel. "A survey on automated dynamic malware-analysis techniques and tools. " ACM Computing Surveys (CSUR) 44, no. 2 (2019):

[3] Vargas, Ruben Jonathan Garcia, Eleazar Aguirre Anaya, Ramon Galeana Huerta, and Alba Felix Moreno Hernandez, "Security controls for Android" In Computational Aspects of Social Networks (CASoN), 2019 Fourth International Conference on, pp. 212-216, IEEE, 2019.

[4] Vinod, P., R. Jaipur, V. Laxmi, and M. Gaur. "Survey on malware detection methods. " In Proceedings of the 3rd Hackers’ Workshop on Computer and Internet Security (IITKHACK’09), pp. 74- 79. 2021.

[5] Mohd Afizi, Mohd Shukran, Wan Sharil and Sham Bin Sharif, “Android Augmented Reality System In Malaysia Military Operations – Unit Positions, ” in Australian Journal of Basic and Applied Sciences, Vol. 6, Issue 8, p79, Aug2019.

[6] Blasing, Thomas, Leonid Batyuk, A-D. Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak. "An android application sandbox system for suspicious software detection" In Malicious and Unwanted Software (MALWARE), 2021 5th International Conference on, pp. 55-62, IEEE, 2021. A_pinto “Android Malware 400% increase ” [ Available online]: http://cybersecurity. mit. edu/2019/11/android-malware-400-increase.

[7] Johnson, Ryan, Zhaohui Wang, Corey Gagnon, and Angelos Stavrou. "Analysis of Android Applications' Permissions. " In Software Security and Reliability Companion (SERE-C), 2019 IEEE Sixth International Conference on, pp. 45-46. IEEE, 2019. 722 Parmjit Kaur and Sumit Sharma

[8] Kern, Michael, and Johannes Sametinger. "Permission Tracking in Android. " InUBICOMM 2019, The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 148-155. 2019.

[9] Yang, Liu, Nader Boushehrinejadmoradi, Pallab Roy, Vinod Ganapathy, and Liviu Iftode. "Short paper: enhancing users' comprehension of android permissions. " In Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices, pp. 21-26. ACM, 2019.

[10] Eric Struse, Julian Seifert, Sebastian, Ullenbeck, Enrico Rukzio and Christopher Wolf. “Permission Watcher: Creating User Awareness of Application Permissions in Mobile Systems” ” in proceedings 2019 springer, pp. 65-80. Springer, 2019.

[11] Sarma, Bhaskar Pratim, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. "Android permissions: a perspective combining risks and benefits. " In Proceedings of the 17th ACM symposium on Access Control Models and Technologies, pp. 13- 22. ACM, 2019.

[12] Nauman, Mohammad, Sohail Khan, and Xinwen Zhang. "Apex: extending android permission model and enforcement with user-defined runtime constraints. " In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, pp. 328- 332. ACM, 2021.Feng, T.; Akhtar, M.S.; Zhang, J. The future of artificial intelligence in cybersecurity: A comprehensive survey. EAI Endorsed Trans. Create. Tech. 2021, 8, 170285.

[13] Sharma, S.; Krishna, C.R.; Sahay, S.K. Detection of advanced malware by machine learning techniques. In Proceedings of the SoCTA 2017, Jhansi, India, 22–24 December 2017. [Google Scholar]

[14] Chandrakala, D.; Sait, A.; Kiruthika, J.; Nivetha, R. Detection and classification of malware. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–3. [Google Scholar] [CrossRef]

[15] Zhao, K.; Zhang, D.; Su, X.; Li, W. Fest: A feature extraction and selection tool for android malware detection. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 714–720. [Google Scholar]

[16] Akhtar, M.S.; Feng, T. Detection of sleep paralysis by using IoT based device and its relationship between sleep paralysis and sleep quality. EAI Endorsed Trans. Internet Things 2022, 8, e4. [Google Scholar] [CrossRef]

[17] Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Using convolutional neural networks for classification of malware represented as images. J. Comput. Virol. Hacking Tech. 2019, 15, 15– 28. [Google Scholar] [CrossRef][Green Version]

[19] Firdaus, A.; Anuar, N.B.; Karim, A.; Faizal, M.; Razak, A. Discovering optimal features using static analysis and a genetic search based method for Android malware detection. Front. Inf. Technol. Electron. Eng. 2018, 19, 712– 736. [Google Scholar] [CrossRef]

[18] Dahl, G.E.; Stokes, J.W.; Deng, L.; Yu, D.; Research, M. Large-scale Malware Classification Using Random Projections And Neural Networks. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing-1988, Vancouver, BC, Canada, 26–31 May 2022; pp. 3422–3426. [Google Scholar]

[20] Akhtar, M.S.; Feng, T. An overview of the applications of artificial intelligence in cyber security. EAI Endorsed Trans. Create. Tech. 2021, 8, e4. [Google Scholar]