REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

ANDROID MALWARE PREDICTION USING MACHINE LEARNING

Sari Khdhaer MUKHLIF

Master’s Thesis

Supervisor

Asst.Prof. Dr. Sefer KURNAZ

Istanbul, 2023

ANDROID MALWARE PREDICTION USING MACHINE LEARNING

Sari Khdhaer MUKHLIF

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled ANDROID MALWARE PREDICTION USING MACHINE LEARNING prepared by SARİ KHDHAER MUKHLIF and submitted on 13 /4 /2023 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

Asst. Prof. Dr. Sefer KURNAZ

Supervisor

Thesis Defense Committee Members

Asst. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Oğuz KARAN	Department of Software Engineering, Altınbaş University	_____
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, Arel University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Sari MUKHLIF

Signature

DEDICATION

First, I would like to thank Allah Almighty for the power of the mind, health, strength, guidance, knowledge, and skills to complete this study.

This thesis is wholeheartedly dedicated to my parents. There are no words to describe what you mean to me; there is nothing that I can repay for what you have done to me. I will continue to do my best to achieve your expectations.

Lastly, I dedicated this to the family, relatives, and friends who have been encouraging me during this study.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Dr. Sefer KURNAZ for all support during my study. Its great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

ABSTRACT

ANDROID MALWARE PREDICTION USING MACHINE LEARNING

MUKHLIF, Sari Khdhaer

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Sefer KURNAZ

Date: April / 2023

Pages: 38

Increasing daily malware that exploits the Internet has become a severe threat. The manual malicious software (a.k.a., malware) identification is no longer valid and effective due to the high prevalence of malware. Thus, automatic behavior-based malware detection using machine learning techniques seems an effective solution. Various studies have proven the efficiency of machine learning to detect and classify malware files. In this research, several machine learning algorithms, including Decision Tree, Random Forest, Logistic Regression, SVM, and KNN, have been investigated to detect malicious software applications. For classification purpose, five classes, namely Adware, Benign, Ransomware, SMS Malware, and Scareware, have been used. Experimental results demonstrated that machine learning algorithms are practical and efficient for malware detection, and 71% accuracy can be achieved with the help of these algorithms.

Keywords: Machine Learning, Ensemble Learning, Android Malware, Android Malware Attack, Static Analysis, Dynamic Analysis, Multiple Classifier Systems.

TABLE OF CONTEINTS

Pages

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGES

ABBREVIATIONS

KNN	:	K-Neighbors Neighbors
MCS	:	Multiple Classifier Systems
SVM	:	Support Vector Machine
PHAs	:	Potentially Harmful Apps
FPR	:	False Positive Rate
MLP	:	Multi-layer Perceptron.
ML	:	Machine Learning
SMS	:	Message/Messaging Service

1. INTRODUCTION

The malware is specifically configured to attack the security policy of the phone system and prevent damage or unauthorized access. Malware can be divided into several types, such as B. Adware, harmless software, ransomware, Smallware, and malware. More than 205 billion applications were downloaded in 2018, and experts predict that about 260 billion apps will be downloaded by 2022. Within the to begin with half of 2019 alone, more than 57 billion apps were downloaded [11]. Only 0.08 % of Android devices with Google Play applications were affected by potentially harmful applications (PHA), but the PHA infection rate of Android devices with applications other than Google Play installed was 0.68 % [17]. The rapid growth of Android malware has brought severe challenges to the anti-malware system because a large number of malware samples overwhelmed the malware analysis system. A promising way to speed up malware analysis is to divide malware samples into multiple families so that standard malware functions in the same family can detect and scan for malware. However, the accuracy of current classification decisions is limited by two aspects: First, legitimate malware can mislead users to use classification algorithms because most Android malware is created by inserting malicious components into typical applications. Polymorphic variants of Android malware can evade detection by switching attacks [16]. In addition, because of the ubiquity of malware, a new generation of signatures using handheld test pilots for malware analysis is useless in the current situation. On the contrary, dynamic analysis has attracted widespread attention to malware detection and classification. Machine learning dynamic analysis-based solutions such as logistic regression, decision tree, random forest, knn, etc., have excellent performance in detecting malware. In the training data that provides high and low FPR accuracy, machine learning methods must be applied to solve the problem of detecting Android malware from different angles [12].This research attempts to check the performance of specific machine learning algorithms, such as Naive Bayes, J48, Random Forest, and Decision Tree To know their ability to solve the described problems by providing the tools necessary to create this combination and developing new mechanisms that focus on accurately detecting and classifying malware. The research focuses on machine learning classifiers, well-trained models, who are trained to represent a classified example for predicting a new class of unlabeled examples in the future. Machine learning classifiers

can be used to detect malware which can divide the suspicious samples into different categories: Adware, Benign, Ransomware, Smallware, and Scareware.

2. LITERATURE REVIEW

2.1 MALWARE

2.1.1 The Definition of Malware

Malware is a software program that ignores the user's choice of how they use their computer or network. Malware designed for criminals, malware designed for criminal, political, and dangerous functions is called malware [27]. Malware evaluation structures automatically, because of the massive pattern quantity, frequently depend on allotted computing to efficaciously method all to be had facts. Therefore, the distribution of the evaluation duties among the community nodes is the essential aspect of the consistency and overall performance of those structures: This aspect is known as Scheduled. Malware has grown to be the most significant danger to the records enterprise over the last few years. According to (AV-Test, an unbiased I.T. safety institute, the range of malware is growing yearly at an unheard-of charge, notwithstanding the usage of malware detection strategies [7]. Groups of programmers often make malware: extra frequently than not, they may be truthful trying to shape cash, both through spreading the malware themselves or through imparting it to the maximum increased bidder at the stupid Web. There might also additionally furthermore be different motives for making malware: it can be used as a protest tool, a way of trying out safety, or while a weapon of struggle is among governments.

Despite diverse anti-malware technologies, malware is left undetectable because of the obfuscation strategies that malware writers used to bypass signature-primarily based malware detection systems. One of the significant negative aspects of the signature-primarily based detection gadget is that the signature database is regularly up to date due to the fact that malware is generated fast with particular signatures [24].

According to Cisco 2017 Annual Cybersecurity Report, 95% of malware analyzed had been below 24 hours. This indicates the tempo of improvement of malware. This is complex malware research.

2.1.2 Malware Types

a. Adware: Adware is a safety danger. This is usually used to acquire advertising facts or show commercials for you to generate monetary returns. This chance isn't the simplest

extra, not unusual place than the conventional danger. However, it additionally exploits extra powerful strategies than strategies utilized in traditional malware. Adware is software that is going past the sensible commercials one may assume from an open software program or shared software program. Typically, spyware is a separate software that installs on an equal time as a pc or similar software. Usually, spyware keeps generating commercials, even if the consumer isn't strolling the prerequisite software program [21]. It is widespread, particularly for cellular programs, for the software to be loose whilst showing an advert banner. Then the purchase of the whole model eliminates this demanding banner. However, this cannot be taken into consideration as marketing and marketing software as it's miles a part of the deal.

b. Benign: Unintended detection can come as posting the records at the company's public internet site or sending the documents to the incorrect celebration thru email, fax, or mail.
c. Ransomware: Ransomware is a specific kind of malware that could lock victims' displays and encrypt their documents to gain ransoms, ensuing considerable harm to customers. Mapping ransomware into households is beneficial for figuring out the editions of a recognized ransomware pattern and for lowering analysts' workload [35].
d. (SMS) malware: SPAM like SMS, WhatsApp, and different messaging offerings is a recognized danger that could affect the safety of cellular gadgets through spreading malware on cellular devices. The safety breach may also motive a cellular tool to ship spam [13].
e. Scareware: Scareware is a contemporary-day kind of malware that could pose monetary and privateness threats for beginner customers. Malware assaults generally attempt to trick pc customers into believing that their pc has been inflamed and provide them with an answer through shopping a counterfeit product that purportedly eliminates malware from their computer systems. There are numerous demanding situations in detecting feared software program assaults: obfuscation and the considerable variety of strategies that may be used to perform such an attack.

2.1.3 Facts Mining Strategies

The fast improvement of facts mining strategies has formed gadget studying as a separate PC technological know-how field. It can be taken into consideration a subcategory of the sector of Artificial Intelligence, in which the dominant concept is the capacity of a gadget (a

PC software, an algorithm) to study from its actions. It changed into first cited as "the sector of examining wherein computer systems can study without being explicitly programmed" through Arthur Samuel in 1959. T. Mitchell had an extra formal definition: "A PC software is stated to study from enjoying E regarding a few elegances of T mission and degree P overall performance if its overall mission performance improves in T, as measured through P, with enjoying E" [19].

A gadget studying mission's fundamental concept is to educate the version, primarily based totally on a few algorithms, to execute a particular task: classification, clustering, regression, and others. Training is achieved primarily based totally on the entry facts set, and the version created later is used to make predictions. The output of this version is primarily based totally on the primary mission and implementation. Possible programs are: giving facts approximately residence attributes, including room range, length, and price, predicting formerly unknown residence price. Another example: Based on datasets with wholesome clinical photos and people containing a tumor, classify a fixed of recent photos; Cluster photos of animals into many corporations of an unsorted pond [6].

To broaden a more profound understanding, it's miles well worth going through the gadget studying method's overall workflow, as proven in Figure 2.1.

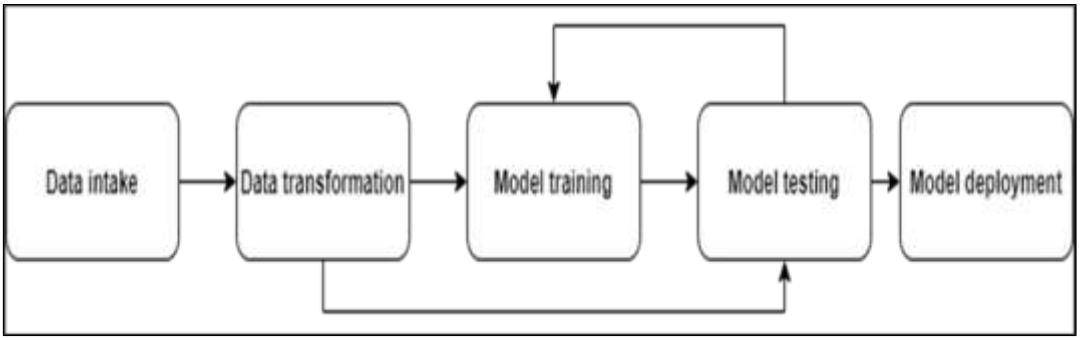

Figure 2.1: The General Workflow Process

The multiple classifier systems are a valuable technique for dealing with the intricate pattern recognition problem. In (MCS), many complementary classifiers are combined by a grouping mechanism to get a better classification result, as shown in Figure 2.2.

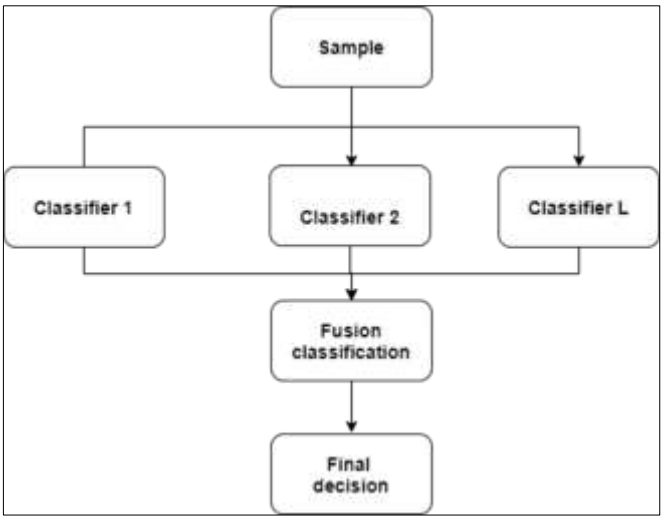

Figure 2.2: The Components of the Multiple Classifier Systems.

MCS can improve classification accuracy, but not all classifier groups can receive accuracy updates. The integration between attendees' workbooks is crucial. This integration is called MC-S flavor. Although MCS is a practical model, it cannot guarantee that the classification accuracy will improve. The integration between element classifiers is essential to improve classification efficiency [15]. Related research Since PCs can provide many services, and mobile devices have recently become more and more popular. The number of security threats on mobile devices and services. This article describes techniques used to detect malware and malware on mobile devices. As part of the research, an Android malware detection system based on approved machine learning technology was developed.

The developed system was analyzed using a random forest algorithm, support vector machine, and artificial neural network [30]. Android applications can be found according to user needs. Software that protects smart authorization lists. The naïve Bayes classifier has exceeded the model construction time [32]. This work will provide behavioral models that can be implemented through static or dynamic analysis. Discussed behavior detection techniques to use machine learning algorithms to create social media-based malware classification and detection models [3]. This document introduces a system for classifying malware based on general behavioral characteristics. Using malware families can measure the similarity between malware families with carefully selected features, usually in the same family. The proposed similarity measure allows us to classify the attack behavior and tactical characteristics of malware. In addition, community detection algorithms are applied to increase the modularity in each network aggregated malware family. To ensure a high degree

of classification accuracy, a method for determining the optimal weight of the selected feature as a measure of the similarity of the hypothesis is proposed.

They are finding out which characteristics are essential to illustrate the similarities between malware examples. Finally, an intuitive graphical representation of malware samples will be provided to help understand the distribution and similarity of malware networks. In an experiment, the proposed system uses the actual malware data set to classify malware with an accuracy of 97%, and the prediction accuracy of K-fold cross-validation is 95% [14]. This study extracted the system call behaviors of 216 malicious applications and 278 typical applications to create feature vectors to train the classifier.

Seven data classification methods, including decision trees, random forests, gradient magnification trees, K-NN, artificial neural networks, support vector machines, and deep learning, are all connected to this data set. Use three function classification methods to select functions from a group of 337 attributes (system calls). Methods to classify features include information retrieval, chi-square statistics, and correlation analysis with weighted features. After excluding specific characteristics in the low range, accuracy and completeness are used to measure the performance of the classifier. Experiments show that the support performance of vector machine (SVM) is better than other methods after feature selection through correlation analysis. These methods can achieve 97.16 % accuracy and 99.54 % memory (for applications). Malicious intentions of Android applications [25].

Use machine learning classification to measure ransomware variations based on their behavior. They utilized behavior reports for 150 ransomware illustrations and ten diverse ransomware arrangements. The data set contains some of the latest available ransomware samples, which provide estimates of the classification accuracy of machine learning algorithms at the current development stage. Iterative methods are used to determine the best behavior characteristics that will be used to achieve the best behavior. By using machine learning algorithms, the researcher may estimate the scoring accuracy of the three machines learning to score calculations as they are collecting the best practices such as unique behavioral characteristics used in ransomware research reports and rankings.

To obtain the best classification results, they conducted experiments to select behavioral characteristics. This method of operation determines many behavioral traits that can be used: classify ransomware to achieve the best health classification and calculate the accuracy of the score. Three machine learning algorithms are provided in (WEKA). J48. The decision

tree algorithm was rated as the best sporting performance. Machine learning confirmed variants of the reliable ransomware example. The new trend of malware is to bypass and detect classification. Behaviors compared to previous behaviors can be circumvented by the detection system by introducing new samples into strange behaviors or by tricking the detection and classification system into behaviors like other ransomware families—the intention of the variant [5].

2.2 MACHINE LEARNING

2.2.1 Supervised Learning

Supervised learning may be a branch of artificial intelligence and machine learning that's learned by employing a bundle of information to train calculations that precisely classify information or predict results. When information is set within the built model, the show is altered to a way better fit, which gets to be a portion of the cross-investigation preparation. Administered learning makes a difference illuminate a wide assortment of real-world issues. An example is sorting spam into a separate file in your inbox. Supervised learning uses some combination of training to teach models to achieve desired goals. The correct inputs and outputs are within the training data set. So, model by model can learn over time. Through the loss function, the accuracy of the algorithms is measured, and they adjust themselves so that errors are minimized. Supervised learning is divided into two types of problems during data research: regression and classification. Regression is used to understand the relationship between independent and dependent variables. It is generally used on forecasts, for example, forecasting the revenues of a particular company. Logistic regression, multiple regression, and linear regression are popular regression algorithms.

Classification uses algorithms to assign test data to specific categories accurately. It learns about particular parts of the data set and makes some conclusions about identifying or naming these entities. Standard classification algorithms are decision trees, random forest, k-nearest neighbor, support vector machines, and linear classifiers [23].

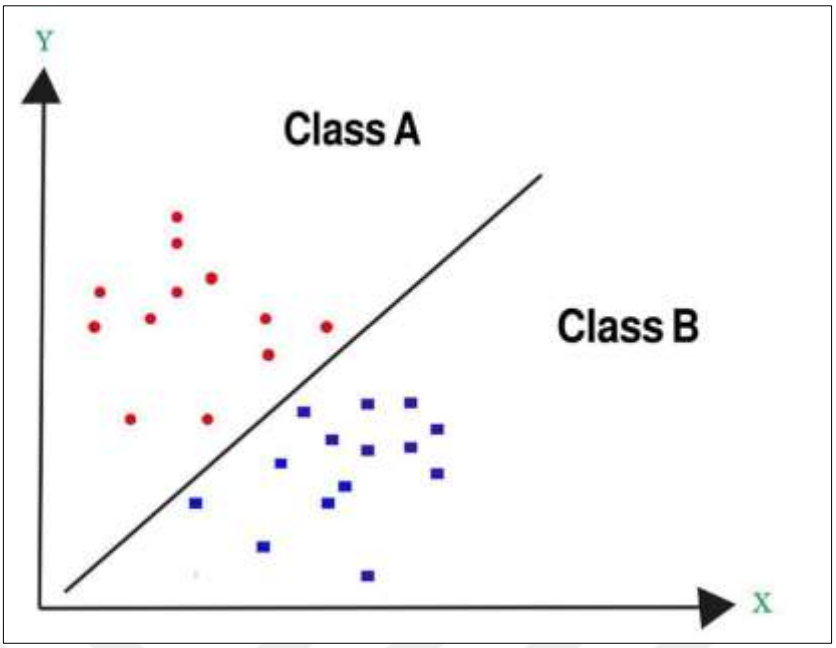

Figure 2.3: Classification Aalgorithm in Machine Learning.

2.2.2 Unsupervised Learning

There are no names in unsupervised learning, and the learning calculation centers exclusively on recognizing structure in the unlabeled input information. Self-organizing neural systems learn by distinguishing covered-up designs in unlabeled input information utilizing an unsupervised learning calculation. The capacity to memorize and organize data without providing a mistake flag to assess the potential arrangement is alluded to as unsupervised learning. In unsupervised learning, the need, of course, for the learning calculation can be invaluable since it permits mathematics to see back for designs that were not already considered. The six major categories of unsupervised learning strategies are summarized in, and the categories are various leveled learning, information collection, fundamental variable models, dimensional lessening, and outside location.

Unsupervised machine learning methods, like current supervised machine learning methods, can be more efficient because they allow the network to manage, monitor, and improve itself while keeping human officials informed and providing actionable information quickly [29].

2.2.3 The Machine Learning Process

As already expressed, the primary step is to gather the information; the moment step is to get ready and reprocess the information. At this point, the data set that will be used as input

to the model stage should be designed. Strange dates and outliers should be entered as missing values, and the row data column should be changed to the date/time format. The data is then mixed so that one of them cannot be used without other malware for training and testing, or it is scaled. Many machine learning algorithms work best when they stand out on a medium comparison scale and are close to a normal distribution. Scikit learning methods for preprocessing machine learning data include Min Max Scaler, Robust Scaler, Standard Scaler, and Normalizer. The strategy is unlikely to be influenced by the model classification and feature values used in this study. Scaling primarily refers to changing the range of values. The distribution method remains unchanged. Consider the building's proportion display cabinet to be the same as the original proportion, small and beautiful [9]. Some characteristics are essential for classification, while others are not. Using classifier input raises the level of complexity. When training the model, the features that have a positive and negative impact on accuracy have been selected, and the remaining elements that give a false impression have been excluded. Built-in strategies have the following advantages: they are exact, generalizable, and interpretable. Random forests are made up of 4 to 1200 selection trees, each of which is based on perceptual and irregularly extracted points that were randomly removed from the data set. All observations confirm that the trees are unrelated and thus less prone to over-fitting. Furthermore, each tree is an array of yes or no questions based on a single option or combination of options. The three division dates are divided into two groups in each center (usually in each direction), and the ideas generated by each group are more comparable and different from the ideas generated by the other group. As a result, each of them has a meaning that is determined by the "purity" of each group [10]. Before using a random selection of forest objects, the data analyst may delete some columns. Columns like "Stream ID," "Source IP," and "Destination IP" are self-explanatory. They are useless for analysis, add no information, and have no effect on the results. The third step is to train the chosen machine learning algorithm on various samples, which are represented as vectors containing the extracted attributes. It focuses on the training of machine learning classification algorithms, which are covered in the following section. The fourth step is to test and evaluate the model; this is the last step in testing the trained model and assessing its effectiveness. It is crucial in the development of android machine learning malware detectors. An excellent choice can help to avoid distortion and train and evaluate a well-designed model. The stage of feature extraction is a series. Static analysis can be extracted

from a variety of resources and converted into executable files—the process of launching an application in the sandbox and acquiring runtime functionality. Many algorithms, such as B. decision trees or models, can be used in the training phase. The choice is primarily determined by the type of malware analysis: the focus, feature selection, and feature presentation. Allows you to test the model's performance on a new sample. This entire process corresponds to a complex task that causes a variety of specific methods. The two goals of malware detection and family identification enabled this research to achieve the following goals, all of which aim to broaden the current investigation:

a. Study the most important behavioral patterns of Android Malware that must be considered when designing Android malware detection tools.
b. To develop tools for automatic extraction of hybrid properties.
c. To create a comprehensive data set of features extracted from classified samples for training Machine learning aids.
d. Study, development, and evaluation of machine learning models to detect Malware in Android.
e. Study, development, and evaluation of machine learning models to classify the malware family for Android.
f. They are assessing the security of machine learning models against hostile attacks. To begin, it is critical to investigate the most significant behavioral patterns that they exhibit. Malware should be taken into account when developing malware detection and classification tools. Information can be enriched by analyzing samples from Android malware families. It was put to be one of the best critical malicious practices that had been carried out. This is critical. When deciding on the initial best set of highlights and approaches for malware analysis to complete this task. Furthermore, this procedure necessitates the use of automated tools capable of extracting an array of features based on specific malware analysis methods. Once the properties space has been defined in which each application's behavior is represented, trait vectors are extracted from extensive sets of samples for machine learning.

The payments, in this case, represent both excellent software and malware. It is required for the development of malware detection tools. Malware from various families would be necessary when focusing on the named family classification. To address the issue at hand, many existing machine learning models can be used. It must be the most appropriate in terms

of performance. It attempted to reduce the number of incorrectly classified cases as much as possible. While previous research has encountered this issue, this work attempts to take a step forward. This issue provides knowledge for the development of more vital models.

2.2.4 Classification Methods

2.2.4.1 K-Nearest neighbors

K-Nearest Neighbors (KNN) is a simple algorithm used in pattern recognition and data mining. It finds the k-nearest neighbors to a given data point. KNN was invented by John Vennard in 1881. K-Nearest Neighbors is often considered one of the most important discoveries in computer science and machine learning since it has broad applications in many fields of study, such as image analysis, bioinformatics, pattern recognition, and text mining. Since the invention of KNN in 1881, more than 30 variants of it have been proposed.

The algorithm works by first analyzing the dataset and finding a partition on the original data set (called "seed" or "training" data). Then for each element in the new set given as input, one may find its k-nearest neighbors to that seed. The algorithm then iteratively computes the k-nearest neighbors of each component in turn. In this way, all features of all elements are found, and their location is calculated.

K-NN could be a sluggish learning calculation, as in, not at all like numerous other measures. It does not create speculation work from preparing information beforehand. Rough implementations of K-NN, just like the one we'll code afterward, will not indeed use the preparing information till it must make a prediction. When a new test is to be classified, the whole preparing information is filtered to induce K neighbors closest to the given test based on a remove metric. The coming about the lesson of the test is calculated using the lion's share vote among the K neighbors. The votes can be weighted based on their removal to the test to maintain a strategic distance from a lesson, with the significant number of tests ruling other classes. Every time K-NN needs to make an expectation, it should run a computation on the whole dataset. This makes the calculation moderate. Modern executions of K-NN like K-D Tree or Ball Tree execution utilize intelligent preprocessing of information to provide much quicker forecasts. These executions store the preparing information on a great day [4].

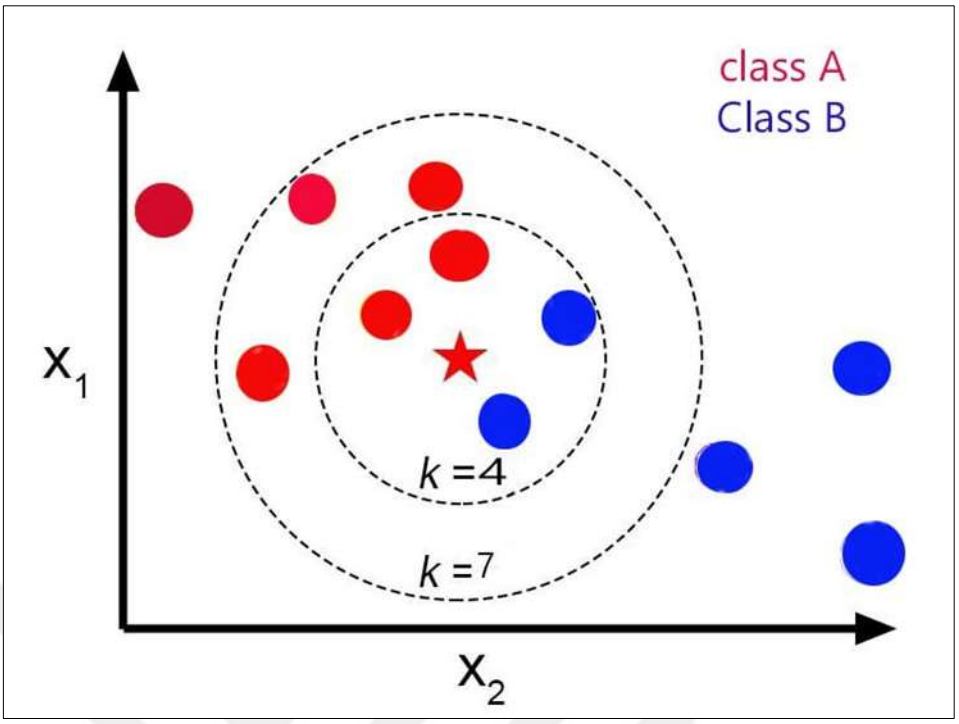

Figure 2.4: K-Nearest Neighbours for Two Classes and Various Number of Neighbours.

2.2.4.2 Random forest

Random forests are supervised learning algorithms. It can be utilized both for classification and backslide. It is also the most adaptable and straightforward calculation. A woodland is made up of trees. It is said that the more trees a forest has, the more vigorous it is. Arbitrary timberlands generate decision trees based on arbitrarily chosen information tests, collect expectations from each tree, and vote on the best arrangement to give an excellent beautiful pointer of the highlighted importance.

Random timberlands contain a variety of applications, including proposal motors, image classification, and include selection. It can be used to categorize trustworthy credit candidates, detect fraudulent behavior, and predict disasters. It is the foundation of the Boruta calculation, which selects imperative highlights in a dataset, as shown in Figure 2.5. It operates in four steps:

a. Select irregular tests from a dataset.
b. Construct a choice tree for each test and get a forecast result from each choice tree.
c. Perform a vote for each expected result.
d. Select the expected result with the foremost votes as the ultimate expectation [20].

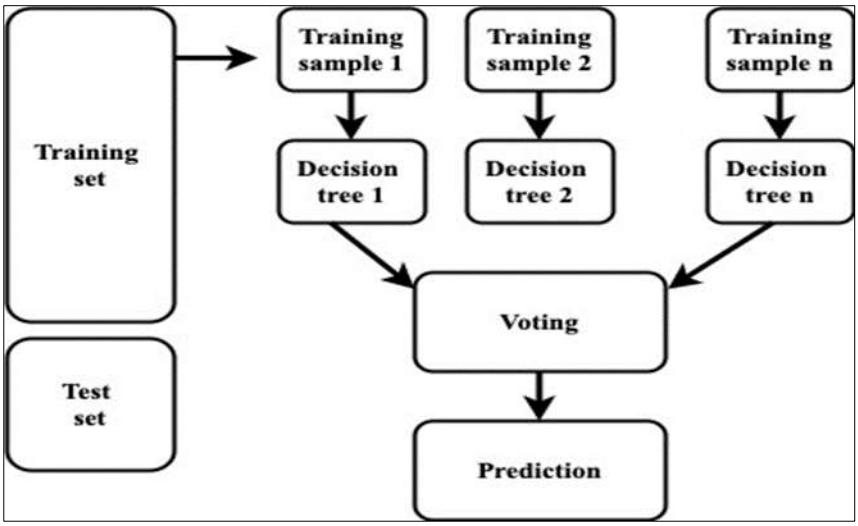

Figure 2.5: Random Forest Model [20].

2.2.4.3 Advantages of random forest model

a. Random Forest is regarded as a profoundly exact and powerful strategy due to the large number of choice trees involved in the process.
b. It is not affected by the overfitting problem.
c. The best reason is that it takes the average of all forecasts, canceling out any biases. The calculation applies to both classification and relapse problems.
d. Random timberlands can also deal with lost values. There are two approaches to cooperation with this: using intermediate values to replace persistent factors and computing the proximity-weighted normal of lost values.
e. The relative highlight significance should be obtained to helps us choose the essential contributing highlights for the classifier [20].

2.2.4.4 The disadvantage of the random forest model

a. Random forests are moderate in producing expectations since they have different choice trees. At whatever point it makes an expectation, all the trees within the woodland ought to make an expectation for the same given input and, after that, perform voting on it. This entire handle is time-consuming.
b. The demonstration is more difficult to interpret than a decision tree, where you can easily choose by following the path within the tree [20].

2.2.5 Logistic Regression

Logistic regression may be a classical factual learning strategy separated into twofold calculated relapse and multivariate calculated relapse. The calculated degeneration alludes to the binomial calculated relapse as appropriate for the two category assignments. The calculated deterioration gauges the parameters of the show through administered learning. The framework shape of the twofold frame is:

$$h\_\theta\ (x) =\ \ 1/(1 +\ e\^(-\theta TX)\ ) \qquad (2.1)$$

Where h_θ (X) ∈ (0,1) m×1 is the output matrix, θ∈Rm×1 is the demonstrate coefficient vector, and X∈R_(m×n) is the highlight lattice of m tests. For each input test x, the likelihood that the test is judged as one is h_θ (x), and the likelihood that the test is considered as is 1 - h_θ(x) [26].

2.2.6 Decision Tree

The decision tree is one of the predictive modeling methods used in statistics, data mining, and mapping. Variables hold a set of values in decision trees called classification trees; Where the leaves are represented in the form of a tree to do what they mean in the branches. Decision trees that target numeric variables with real numbers are called regression trees (relative to linear regression). Decision making Continuous decision-making in data mining operations related to data management. A decision tree requires supervised learning, and supervised learning is the giving of a set of tests; Each test includes a set of attributes and a rating result. This means that the classification result is known. The decision tree calculation tries to fathom the issue by employing a tree representation. Each internal hub of the tree corresponds to an attribute or feature, and each leaf node corresponds to a class label. The level of understanding of the decision tree algorithm is elementary compared to other classification algorithms [31]. The decision tree is divided into three types of nodes, starting with the Robot Node or sometimes called the Input Node, where each row of data enters separately. Then the Feature Node where this data is broken down by features such as color. Then the leaf node, which is the node that has no children and usually represents the final class. Note that feature nodes take order to be up or down according to their ability to divide data with minimal confusion. For example, if we classify shapes into squares and triangles

using a decision tree. A feature that separates the data so that it combines the most significant number of triangles in one section and the largest number of squares in another province without mixing the squares with the triangles will be preferred so that the data is pure. This process is measured using arithmetic values such as entropy, Gini, and Gain, whose equations will be explained below

2.2.7 Trait Determination Measures

If the dataset contains N traits, deciding which property to place at the root or different levels of the tree as inside hubs may be a challenging step. The problem cannot be solved by arbitrarily selecting any corner to be the root. In the absence of opportunity, a self-assured approach may give us awful results with moo precision. Analysts worked on and planned a few arrangements to understand this property determination issue better. They proposed using a few criteria like entropy, information gain, Gini index, Gain Ratio, reduction in variance, chi-square. These criteria will calculate values for each quality. The values are sorted, and qualities are set within the tree by taking after the arrangement, i.e., the rate with a tall discount (in case of data pick up) is charged at the root. While utilizing data pick up as a measure, traits have been accepted to be categorical, and for the Gini file, properties are acknowledged to be nonstop [33].

2.2.7.1 Entropy

Entropy is used to gauge the degree of impurity. A skewed distribution has low entropy, while a distribution where events have equal probability has greater entropy. Entropy helps us build a suitable decision tree for choosing the best distributor [1].

2.2.7.2 Information gain

Information gain is the fundamental step where a computer program design tries to encourage data around the target. Program engineers utilize contrasting sources and devices to activate more data. Making a decision tree is all-around finding a property that returns the first fundamental data select up with the least sum of Entropy [8].

2.2.7.3 Gini index

Understanding the Gini index required a critical sum of work utilized to assess different perspectives of the dataset. It is calculated by subtracting the entirety of each lesson's squared probabilities from one. It favors more significant allocations that are basic to actualize, though information inclines toward littler disseminations with outright values. The Gini index is utilized as well to degree pollution lessening [22].

$$ini = 1 - \sum\_(i = 1)\text{^}c \text{▒} 〖(Pi)〗 \text{^}2 \quad (2.2)$$

Where Pi indicates the probability of a component being classified for a particular class. Gain ratio: The gain ratio could be a ratio of information gain to inherent data to decrease a predisposition towards multi-valued properties by considering the number and measure of branches when choosing a feature. Reduction invariance: Reduction invariance could be a calculation utilized for continuous target variables (relapse issues). In this strategy, the most excellent component is chosen utilizing the normal condition of alter. As the measure for part the populace, the portion with the minor alter is chosen:

$$\text{Variance} = (\Sigma(X - \bar{X})\text{^}2)/n \quad (2.3)$$

After finding the esteem of the mean $\bar{x}$, it is subtracted from each component x and it is squared to urge the squared contrast. Once the squared contrasts of eac.

2.2.7.4 Reduction invariance

Reduction invariance could be a calculation utilized for continuous target variables (relapse issues). In this strategy, the most excellent component is chosen utilizing the normal condition of alter. As the measure for part the populace, the portion with the minor alter is chosen:

$$\text{Variance}=(\Sigma(X-\bar{X})\text{^}2)/n \quad (2.3)$$

After finding the esteem of the mean $\bar{x}$, it is subtracted from each component x and it is squared to urge the squared contrast. Once the squared contrasts of each component are added, it is divided by the full number of components within the populace. The result is the variance of the dataset.

Chi-square: Chi-square could be an abbreviated shape for Chi-squared Modified Interaction Locator. It is one of the most setup tree classification strategies. It finds the essential centrality of the refinements between sub-nodes and parent centers. The entire squares of standardized contrasts between observed and anticipated frequencies of the target factors ought to be measured [28].

$$X^2 = \sum((O-E)^2)/E \quad (2.4) \qquad (2.4)$$

Where:

X^2 is Chi square obtained is observed score E is expected score

2.2.8 SVM

A support vector machine (SVM) is a type of classification calculation in supervised machine learning. SVMs were first presented within the 1960s and, after that, refined within the 1990s. They are, as of now, getting to be an excessively predominant sense of their capacity to realize brilliant things. SVMs are executed especially when compared to other machine learning calculations. A standard machine learning calculation attempts to find a boundary that allotments the data so that the misclassification goof is minimized within the case of straightly detailed data in two estimations. A number of limitations that precisely partitioned the data center can be seen. The data is accurately classified by the two dashed lines and one firm line. A few boundaries that precisely isolated the data center can be seen. The data is accurately classified by the two dashed lines and on. SVM differs from other classification algorithms because it selects the decision boundary that maximizes the distance from the closest information focuses of all classes. An SVM not only finds a decision boundary but also finds the best decision boundary [2]. The ideal decision boundary is the one with the most significant advantage from the closest focuses of all classes. Support vectors are the closest focuses from the decision boundary that maximize the distance between the decision boundary and the directions. With support vector machines, the decision boundary is referred to as the most extreme edge classifier or the greatest edge hyperplane. Finding the back vectors, calculating the edge between the choice boundary and the support vectors, and maximizing it all require complex arithmetic. The arithmetic details in this tutorial won't be taken into consideration while how SVM and Part SVM are implemented in Python using the Scikit-Learn library will be studied [16].

2.2.8.1 Ensemble

To improve the performance of the learning algorithms, ensemble methods have been used where multiple classification models vote for the Y label, and the result with the higher voting score has been collected. The ensembles that contained combinations of three and five algorithms have been experimented with. Costumes have been used with models like SVM, Decision Trees, Random trees, Random Forests, and Logistic Regression [18].

3. PROPOSED METHODOLOGY

This paper proposes a two-stage classification model for studying application collusion: The first stage is a hybrid classifier, which includes the KMeans clustering algorithm and a series of linear vector machines (SVM). Stage 1 Benign and Malicious Applications-KMeans: this algorithm divides the data set into several groups. Linear SVM training for each group and using the parameters obtained from the linear SVM to create a parameter vector. The second stage of the classification model uses parameter vectors and light discriminant functions to detect collusion in applications. This model can see complex application pairs as well as single malicious applications. The model is less computationally intensive and easy to implement. This article proposes an opcode-based method to analyze Android malware. Different machine learning algorithms have been used to classify malicious Android applications. From the results obtained, this method can organize malware more effectively and accurately, with an accuracy rate of 99.5% and a TPR of 0.995. Machine learning algorithms have been used to browse static analysis of Android malware quickly. Machine learning algorithms play an essential task in the research and classification of Android malware. They use an opcode-based approach to analyze Android malware. To carry out this work, many data have been collected. The two principal parts of this research are the identification of behavioral features, which can be used for the best classification accuracy, and ransomware classification.

3.1 DATA PREPARATION

The data used in this study from Canadian Institute for Cybersecurity android malware and it contains five strings of malware attacks: adware, benign software, ransomware, SMS malware, and Scareware. This study goals to develop a model that can be used to categorize new future data into these five categories. It should be noted that each attack type may have a different amount of raw data to balance the data during the training process. During the testing phase, the 3,000 rows of data are used for each type of malware attack will be checked. The research starts with the use of machine learning classification to identify future attacks. When the analysis is not performed without running on the system, malware analysis is called static analysis. Otherwise, it is called dynamic analysis. Static analysis is a fast low-generation false positive rate (FPR), but the fixed functions used are sensitive to obfuscation

techniques and cannot detect undetected malware. However, these methods are time-consuming processes for learning and collecting redundant information because many attributes will lead to a decrease in inaccuracy. The false-positive rate (FPR) is high, which makes floor placement impractical. This requires more research to develop methods that require less computation and reduce redundancies. The learning process of the classification model can be summarized as the following steps: data extraction and collection, data analysis and processing, modeling, testing, and evaluation. All these steps will be clarified within the taking after segments.

3.2 RESEARCH QUESTIONS

The questions that this research will answer are: can the machine learning algorithms and tools be used to build malware attacks classification models capable of classifying different malware families' attacks? What is the effectiveness and accuracy of these models, and are they trustable to predict the attacks in the future?

3.3 CONTRIBUTION

This research worked on a large dataset through the process of picking and using favorable ones and incorporating machine learning algorithms classifying the five mentioned in this thesis document. Then, to build a model that can pinpoint android malware and categorize it into different types, binary classification was used. Two classes, namely ransomware and malware, which are the types of attacks that occur were used to represent the malicious software. In this study, different machine learning models have been built to evaluate the performance of machine learning-based Android malware prediction models.

4. RESULTS

The terms used in the research has been explained below to understand the results clearly:

a. Accuracy: The extent of the overall number of correctly classed apps, whether as benign or malicious.

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{4.1}$$

b. The F-score: The F1-score, also known as the accuracy of a model on a dataset, is a measure of its accuracy. It is utilized to assess binary classification frameworks, which categorize cases as ‘positive' or ‘negative.' The F-score may be a way of combining the model's precision and recall, and it is defined as the concordant cruel of the model's accuracy and recall. The F-score is commonly used for evaluating data recovery frameworks such as look motors, as well as many types of machine learning models, particularly in standard dialect preparation [34]. Formula One Score:

$$F_1 = \frac{2}{\frac{1}{recall} \times \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{4.2}$$

c. Precision: is the division of simple, cheerful illustrations among the cases classified as positive by the show. In other words, the number of genuine positives divided by the number of false positives equals the number of actual positives [34].

$$Precision = \frac{tp}{tp+fp} \tag{4.3}$$

d. Recall: The division of cases classified as positive among the total number of positive illustrations, also known as affectability, is the division of patients classified as positive among the absolute number of positive specimens, the number of true positives is isolated by the number of true positives additionally false negatives [34].

$$recall = \frac{tp}{tp+fn} \tag{4.4}$$

TP: The number of true positives classified by the model.

FN: The number of false negatives classified by the model.

FP: The number of false positives classified by the model.

4.1 BINARY CLASSIFICATION

Binary classification results: as the name implies, binary classification has only two classes, 0 OR 1. The data has been classified only to Benign and Ransomware Charge classes, as shown in Table 4.1.

Table 4.1: The Accuracy, F1, Precision and Recall Score According to The Binary Classification.

Model\Measure	Accuracy	F1_score	Precision score	Recall score
Binary Classification	91.27 %	77.44 %	86.45 %	73.35 %

What can be deduced from the previous table is that the model, in general, can predict a highly correct class for the malware. This is expected in the Binary Classification because there are only two classes to differentiate them during learning and prediction. This is evident from the result of the accuracy, which is calculated by measuring the number of correct predictions relative to all predictions. In more detail, the model is able to predict the True Positive TP very well through the Precision value of approximately 87 percent of cases, which represents the number of times the model predicts malware like ransomware, and it was actually ransomware, for instance. However, it seems that the model can do better in predicting a False Negative, which is a value opposite to the precision, it is called the recall, meaning that the model, with a percentage of 74 percent, predicts malware to be ransomware, but it is in fact Benign. This is a lower value than the previous values, and therefore the F-score decreased to 87 percent because it represents the balance between Precision and Recall.

4.2 MULTI-CLASS CLASSIFICATION

The model's output is shown below, where’s the paramount accuracy they are above 60 %. As previously stated, extracting the features allows us to remove the fundamental characteristic of the classifier and discard the rest that is not. However, unlike the binary classification, the results in the multi-class classification were not accurate enough, as shown in Table 4.2.

Table 4.1: The Accuracy, F1, Precision and Recall Score According to The Multi-Class Classification

Model\Measure	Accuracy	F1_score	Precision score	Recall score
KNN	61.89 %	55.86 %	59.86 %	53.50 %
Random Forest	70.78 %	65.05 %	70.92 %	61.83 %
Logistic Regression	58.07 %	30.56 %	52.18 %	35.31 %
Decision Tree	67.97 %	62.57 %	62.44 %	62.70 %
SVM	57.13 %	42.45 %	56.53 %	45.74 %

From Table 4.2, it is noted that models become a more difficult task in differentiating between multiple classes. In fact, and by looking at the results of most models, it can be concluded that the predictions are more like guesswork with accuracy rates close to 50%, with the exception of the random forest model, which is able to give better predictions because it passes over the data more than once during more than one Decision Tree, and therefore it is more capable of the prediction because it learns patterns in the data better. In conclusion, the table shows that more data preparation operations and a change in the prediction methodology are needed to pursuit.

As previously explained, the voting group method has been used, in which the data is rated with the highest voting score, and the accuracy is improved once more to reach 71.67 %, as shown below in Table 4.3.

Table 4.2: The Accuracy, F1, Precision and Recall Score After Ensemble

Model\Measure	Accuracy	F1_score	Precision score	Recall score
After Ensemble	71.67 %	65.37 %	71.48 %	62.12 %

The final Table 4-3 gives a final result after the data preparation and ensemble process, and it is clear that it gets better results than just guessing in the predictions of the multi-class classification. There remains the need to improve the accuracy results, especially the Recall score, which was previously explained. This is what leads us to the discussion section and future work and what can be done to obtain better results.

4.3 TECHNOLOGIES AND ENVIRONMENTS

a. Python version 3.8.8.is the most popular AI and machine learning programming language because it is simple and self-explanatory, allowing analysts and engineers to focus on algorithms rather than excellent programming skills. To avoid errors, old versions, and release issues, the most recent Python version available has been used.
b. Anaconda includes a download package manager to assist in obtaining all required libraries such as Panda and Sklearn, where models such as KNN and SVM have been trained. These libraries and models occasionally require working and installing environments in specific versions; Anaconda has solved this problem and assisted developers in creating specific work environments as needed.
c. Pandas: is a data control and examination program library written in the Python programming language. It provides data structures and operations for controlling numerical tables and time arrangement in particular.
d. NumPy: it is a Python library that provides a multidimensional array question, various inferred objects (such as masked arrays and matrices), and a collection of schedules for quick operations on arrays, counting numerically, consistently, shape control, sorting, selecting, I/O, discrete Fourier changes, fundamental straight variable based math, essential factual operations, arbitrary reenactment, and much more.
e. Scikit-learn: It's a free machine learning library for the Python programming language. It emphasizes various classification, relapse, and clustering calculations using back vector machines, arbitrary timberlands, slope boosting, k-means, and DBSCAN, and is planned to interoperate with the Python numerical and logical libraries NumPy and SciPy.

5. DISCUSSION

5.1 BINARY CLASS CLASSIFICATION

Due to the limited number of classes -only two-, the accuracy was outstanding. The model knew the characteristics of these two types intensely during training and was able to differentiate between them well, and for this accuracy was (91.2%) where's this is not the case in real life because the malware has many forms that need to be classified. In this study, only five different types of malwares have been initiated.

5.2 MULTI-CLASS CLASSIFICATION

In the multi-class classification, the accuracy has been dropped very noticeably, and unlike the binary type, the accuracy here is not good. The reasons behind this decrease may be due to the number of the classes' classification, and it is possible that the characteristics and features of each class were similar during data training, so the model will eventually make wrong classifications during the test. Also, the large number of these characteristics may affect the quality of training. That's why models that have been trained for classification such as KNN, random forest, decision tree, and SYM hasn't provided us with a good result, and to overcome the previous multi-class classification misclassification, two methods were used to increase the accuracy:

a. Better data preparation: The cleaner, interconnected, and well-arranged the input data is, the better the model will learn from it, and thus the model results will be improved. Based on this basis and to enhance the results of the multi-class classification, Data scaling has been introduced. This converts features with large numbers into smaller numbers and is more readable to handle. And it became pure from the results that the model prefers this.
b. Feature selection**:** It helps to select the characteristics that affect the classification result and leaves the rest outside the model training process. This will increase the performance of the model and learn only the essential things and focus on them.
c. Voting ensemble: Where each of the trained models predicted the test data, and then the class that got the highest vote was selected. This will also help raise accuracy because it is a process that involves all classifiers - with their different ways of training - together to vote on the expected class. And therefore, automatically, the accuracy will increase, and

the error rate will decrease from the previous time, and this is what happened, as the accuracy became higher after that.

6. CONCLUSION AND FUTURE WORK

To summarize, many malware attacks occur on Android and other devices every day because of online downloads, uploads, and different types of data transactions. As a result, traditional methods or static software cannot be used to combat attacks such as Adware, Benign, Ransomware, SMS malware, and Scareware. This will provide benefit both for the attacks and their users. Instead, dynamic Artificial Intelligence and Machine Learning-based solutions introduce a better process for analyzing and classifying malware attacks, as well as predicting a new attack mission without being specifically and precisely programmed. In this study, machine learning classification algorithms have been used to build a model for predicting new attacks. The primary types of machine learning are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. While supervised learning refers to algorithms that use labeled databases, unsupervised learning refers to methods that can work on datasets that include no class label. In this study, previously labeled records were used to characterize malware attacks. Unsupervised learning, on the other hand, can also be an interesting research direction for future work.

The significant steps in the Machine Learning process are data extraction and gathering, data analysis and reprocessing model training, testing, and evaluation. Starting with data collection and progressing to the next step, which was preparing data for input into the training mode by removing extraneous data and outliers, filling in missing values, shuffle the data, scaling the data, and extracting data features are all examples of data processing. Several classification models, including SVM, Decision Tree, Logistic Regression, Random Forest, and K-nearest neighbors, demonstrated binary and multi-classification throughout the training phase. The study findings started quite well for binary classification; however, this is not the situation in the actual world, so other classes have been added. To improve classification accuracy, data features, extraction, and ensemble voting are required in multi-classification. In the study, the terms that have been used to evaluate the experiments have been explained so that readers could better understand the study findings, such as F1 Score, Precision, Recall, and others. The researcher is aware of impending duties and challenges. The risk of attacks mutating and interfering with one another has been demonstrated, as well as how it changes across time, place, and culture. Shortly, the big data technology would be needed to manage millions of data transactions per day while also allowing the engineers to

adjust and build the models online. The attacks and the attack evolve have been classified, and for example, the social engineering attack must keep up because these attacks take different forms, overlap, and mutate over time. Furthermore, there is a massive amount of data transmitted over the internet, and this enormous amount of data can't be collected to deal with in an excel sheet or a single dataset. As a result, given the importance of big data, extensive data methodologies and technologies should be included in such future projects. Technologies like The Apache Hadoop databases that allow for the distributed processing of large data sets across clusters of computers, and Apache Spark that provides messaging and streaming systems for big data. These technologies will aid in the online and frequent training and updating of the used models.

REFERENCES

[1] Brown, D. R. L. “Formally assessing cryptographic entropy,” IACR Cryptology ePrint Archive, Pp. 659,2011.

[2] Chamasemani, F. F. and Singh, Y. P. “Multi-class Support Vector Machine (SVM) Classifiers -- An Application in Hypothyroid Detection and Classification,” in 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications, pp. 351–356,2011.

[3] Choudhary, S. and Sharma, A. “Malware Detection amp; Classification using Machine Learning,” in 2020 International Conference on Emerging Trends in Communication, Control, and Computing (ICONC3), pp. 1–4. 2020.

[4] Chumachenko, K. “Machine Learning Methods for Malware Detection and Classification” , 2017.

[5] Daku, H., Zavarsky, P. and Malik, Y. “Behavioral-Based Classification and Identification of Ransomware Variants Using Machine Learning,” in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp.1560–1564,2018.

[6] Fan, M. et al. “Frequent Subgraph Based Familial Classification of Android Malware,” in 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 24–35,2016.

[7] Galal, H.S., Mahdy, Y.B. and Atiea, M.A. “Behavior-based features model for malware detection.” J Comput Virol Hack Tech 12, PP.59–67 ,2016.

[8] Grus, J. “first principles with python.” Data science from scratch: O'Reilly Media,2019.

[9] Hale, J. “Scale Standardize or Normalize with Scikit-Learn. Accessed.”,2019.

[10] IFRS, I. “Insurance contracts. IFRS Standards Project Summary.”2017.

[11] Imtiaz, S. I. et al. “DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network,” Future Generation Computer Systems, pp. 844–856,2021.

[12] Joshi, S., Upadhyay, H., Lagos, L., Akkipeddi, N.S. and Guerra, V., "Machine learning approach for malware detection using random forest classifier on process list data structure." In Proceedings of the 2nd International Conference on Information System and Data Mining ,pp. 98-102,2018.

[13] Khan, J., Abbas, H. and Al-Muhtadi, J. "Survey on Mobile User's Data Privacy Threats and Defense Mechanisms", Procedia Computer Science, 56, pp. 376–383,2015.

[14] Kim, H. M. et al. "Andro-Simnet: Android Malware Family Classification using Social Network Analysis", in 2018 16th Annual Conference on Privacy, Security and Trust (PST), pp. 1–8,2018.

[15] Liu, J., Han, D. and Yang, Y. "Pest Identification Based on Multiple Classifier System", in 2018 37th Chinese Control Conference (CCC), pp. 9535–9539,2018.

[16] Malik, U. "Implementing SVM and Kernel SVM with Python's Scikit-Learn,"2018.

[17] Marlene,G. "Global Android devices with potentially harmful apps (PHAs),"2018.

[18] Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I. and de Mendonça, A., "Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity, and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees, and random forests." BMC research notes, pp.1-14,2011.

[19] Mitchell, T.M. "Machine learning."1997.

[20] Navlani, A. "Understanding random forests classifiers in Python.,"2018.

[21] Ndagi, J. Y. and Alhassan, J. "Machine Learning Classification Algorithms for Adware in Android Devices: A Comparative Evaluation and Analysis", in 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), pp. 1–6,2019.

[22] Nembrini, S., König, I. R. and Wright, M. N. "The revival of the Gini importance", Bioinformatics, pp. 3711–3718,2018.

[23] Sathya, R. and Abraham, A. "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification", International Journal of Advanced Research in Artificial Intelligence, 2013.

[24] Shijo, P. V and Salim, A. “Integrated Static and Dynamic Analysis for Malware Detection”, Procedia Computer Science, pp. 804–811,2015.

[25] Singh, L. and Hofmann, M. “Dynamic behavior analysis of android applications for malware detection", in 2017 International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 1–7,2017.

[26] Suhuan, L. and Xiaojun, H. “Android Malware Detection Based on Logistic Regression and XGBoost”, in 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pp. 528–532,2019.

[27] Tyler ,M. and Marie,.“Stop Badware Project.” Reference: http://www.stopbadware.org. Available: https://www.av-test.org/en/statistics/malware/.2017.

[28] Ugoni, A., and Walker, B. F. “The Chi-square test: an introduction.” COMSIG review, PP.61–64,1995.

[29] Usama, M. et al. “Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges”, IEEE Access, pp. 65579–65615,2019.

[30] Utku, A. and Doğru, İ. A. “Malware detection system based on machine learning methods for Android operating systems”, in 2017 25th Signal Processing and Communications Applications Conference (SIU), pp. 1-4,2017.

[31] Utku, A., Doğru, İ. A. and Akcayol, M.. “Decision tree based android malware detection system”, in 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1–4,2018.

[32] Varma, P. R. K., Raj, K. P. and Raju, K. V. S. “Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms”, in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 294–299,2017.

[33] Wellmann, J. F. and Regenauer-Lieb, K. “Uncertainties have a meaning: Information entropy as a quality measure for 3-D geological models”, Tectonophysics, 526–529, pp. 207–216,2012.

[34] Wood, T. (2019). “F-score.” Retrieved February 17, 2021.

[35] Zhang, H. et al. “Classification of ransomware families with machine learning based on N-gram of opcodes”, Future Generation Computer Systems, pp. 211–221,2019.