SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I2P113

Volume 12 Issue 2, 140-151, February 2025

Original Article

Machine Learning Based Hybrid Approach in Ransomware Recognition and Classification

M.S. Balamurugan[1], V. Rajendran[2], S. Suma Christal Mary[3]

[1,2]*Department of Electronics and Communication Engineering, Vels University, Chennai, Tamilnadu, India.*
[3]*Department of Information Technology, Panimalar Engineering College, Chennai, Tamilnadu, India.*

[3]*Corresponding Author : sumasheyalin@gmail.com*

Received: 18 December 2024 Revised: 16 January 2025 Accepted: 15 February 2025 Published: 26 February 2025

Abstract *- Cyber security is severely restricted by spyware, ransomware, along malevolent assaults, which can seriously harm networks, server rooms, websites, and mobile devices in a variety of commercial and industrial settings. Conventional anti-ransomware software finds it difficult to defend against immediately developed, highly competent attacks. As a result, contemporary techniques such as conventional and neural network-based topologies can be greatly applied to creating novel ransomware remedies. This research work employs a feature selection-based method along with implementing machine learning classification approaches in ransomware malware recognition and classification. Moreover, we developed six machine learning approaches: Adaptive Boosting, K-Nearest Neighbor, Stochastic Gradient Descent, Extra tree, Artificial Neural Network and Hybrid approaches based on preferred features for ransomware malware classification. Our investigational outcomes reveal that the proposed hybrid model outperforms conventional approaches with a detection accuracy of 99.5% in terms of measures like accuracy, precision, F1-score, Recall, Matthew's Correlation Coefficient and Kappa score.*

Keywords *- K-Nearest Neighbor (KNN), Stochastic Gradient Descent (SGD), Mathew's Correlation Coefficient (MCC).*

1. Introduction

One of the biggest cyber security risks that enterprises are currently experiencing is the quick spread of ransomware attacks. Ransomware is a technique that hackers have been using more and more frequently in recent years to extort money from victims by encrypting their data and requesting payment for a decryption key. Ransomware attacks have affected every sector of the economy, including government, education, healthcare, and finance. Understanding ransomware attacks, their propagation, and the possible repercussions of becoming a victim of one is essential, given the high stakes involved in De Groot et al. [1]. It is impossible to overestimate the significance of this field's research. The increasing threat of ransomware attacks necessitates further investigation into the issue by academics and industry professionals in order to develop practical approaches to control and avoid Zakaria et al. [2]. For instance, in 2017, the most well-known ransomware attack in online history struck MAERSK, the world's number 1 shipping company. Within minutes, 56,000 devices had their encryption completed. All gadgets linked to the MAERSK network have encryption installed on them. MAERSK Shipping ships 25% of the world's food supply also, this malware brought everything happening to an abrupt end [3]. Ransomware payments exceeded $1 billion in 2023, the biggest amount ever recorded. Despite a drop in ransomware payments in 2022, the overall upward trend from 2019 to 2023 shows that ransomware is becoming a more prevalent problem. The prevalence and ongoing expansion of ransomware present significant challenges to observing every incidence and tracking all cryptocurrency ransom payments. It is crucial to understand that these are only cautious estimations that will probably rise if more ransomware addresses are found in future. For example, the $457 million in ransoms [4] first reported for 2022 in burglary records from the previous year was raised by 24.1%. Figure 1 shows the statistics of ransomware malware from the year 2019-2023, as stated by chainalysis [4].

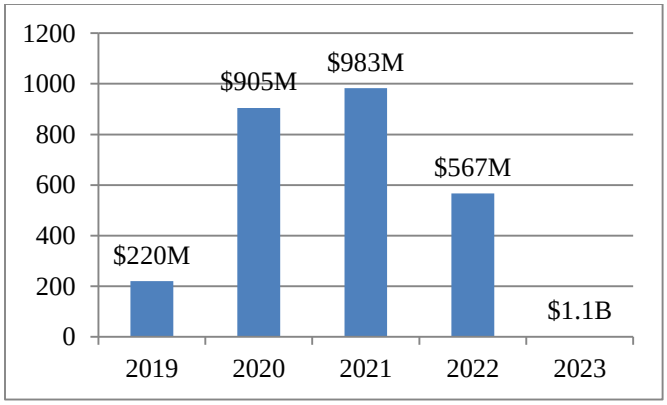

Fig. 1 Entire value received by ransomware attackers (2019-2023)

1.1. Research Gap

Cybercriminals develop spyware that is malicious and installs it on multiple computers with the intention of gaining permission or causing injury. Enterprises utilize a variety of techniques, such as anti-virus programs, log file assessment, including activity surveillance, to search for indicative signals of abnormal or fraudulent activity that could point to a known risk or offensive strategy Zaharia et al. [2]. Using signature-based malware detection techniques to find well-known threats can yield efficient outcomes.

However, attackers can easily circumvent such systems. Many studies have been done to improve hazardous file recognition, aiming to increase detection rates, lower FPR, and lower execution time. However, because of various problems in malicious software detection environments, this type of research is difficult to expand upon and advance. In this research, we examined numerous techniques for identifying malware in files that were previously made public and discussed areas in which further research is still needed.

We examined the efforts being made to standardize the measurement, description, assessment, and framework that enable malware detection, in addition to identifying components that could be useful in improving the accessibility and extensibility of research on dangerous file identification.

1.2. Objectives

The main intention of this research work is as follows

- Gathered malware-based ransomware metadata from Kaggle, an open resource website comprised of several datasets.
- Feature selection-based methods and normalization techniques were applied to standardize the features by transforming input data into preprocessed form.
- We provide a thorough examination of the classification of ransomware and suggest architecture by choosing several features for the model development process and utilizing conventional ML classifiers.
- Developed six machine-learning classification approaches suitable for detecting and classifying ransomware malware that affects the entire network system.
- Comparison has been made with existing research work on ransomware malware classification in terms of accuracy, precision, recall and other evaluation metrics to enhance the model's performance.

2. Related Work

This survey provides an overview of ransomware malware that affects personal computers, including Linux, Windows and Android, smartphones and tablets. Also, it describes several kinds of ransomware, explains how it works, provides a way to take ransom, and lists the devices commonly addressed as ransomware.

2.1. Ransomware on PC

Malware that demands a ransom from users in order to unlock encrypted files is referred to as ransomware. Your files will be encrypted as soon as ransomware infiltrates your computer's system, and you cannot use the device's standard features. Afterwards, you will receive a message from the hackers requesting payment in exchange for recovering your filesDickson et al. [5].

2.2. Ransomware on Edge Devices

The impacts of ransomware malware on edge computing devices such as gadgets, mobile phones, tablets and Personal computers were explained by Goteng et al. [6]. During the year 2020, more than 50 billion edge devices were interconnected to the network; thereby, various applications on edge devices were severely attacked by ransomware.

2.3. Recognition of Ransoms

Nowadays, increasing amounts of illegal activities involve using various types of ransomware. Here, we mentioned how this ransomware is detected in the network environment Alrimy et al. [7].

- Watch out for known file extensions.
- Watch out for enhancement in renaming files.
- Developing network sharing.
- Creating anti-ransomware agents.

2.4. Ransom Prevention

Ransoms are prevented by following these steps.

- Make frequent backups of your files and maintain an up-to-date off-site backup. Beyond ransomware, backups can shield your data from other threats.
- To ensure that only you can restore the backed-up data, ensure it is encrypted Tailor et al. [8].
- When opening unsolicited attachments, proceed with extreme caution. iii. Avoid granting yourself greater login authority than is required. Never log in as an administrator longer than is necessary [8].
- Refrain from using your administrator login to browse, open documents, or perform other routine work tasks [8].

The features are chosen via N-grams to extract features N-grams; hence, it displays improved bias among malware families Zhang et al. [9]. After this, such selected features are fed into ML algorithms and undergo classification as malware and benign. Zimba et al. [10] virus-free incursions present a novel attack vector that would be highly appealing to a hacker since it does not need a third-party mule and does not require the user to take any action. The accessibility backdoor, a malware-free infection vector that allows for system-level access to be obtained without logging during pre-authentication over an RDP session, was introduced. Alraizza et al. [11] surveyed how machine learning algorithms are suitable for detecting ransomware and how to prevent such kind of malware, which damages devices, files and various

software. Around 20 articles were reviewed by Alzahrani et al. [12] regarding the ransomware overview and how deep learning algorithms help find ransomware in Android applications. Moreover, our survey explains the ransomware detection in which the Zero-day attack found by Seong et al. [13] to predict and classify ransomware in such a way that the CF-NCF technique to vectorize the data and Zuhair et al. [14] using multi-tier streaming analytical method. Few more authors, the occurrence of ransomware in Android applications proposed Kirubavathi et al. [15] on 331 permissions on Android, Multi-factor feature filtration approach Bibi et al. [16] Application Programming Interface based ransomware detection by Alsoghyer et al. [17] on 2959 ransom samples, unknown malware detected Ashu Sharma [18] using RF classifier with 97.5% accuracy, analyzed android malware via probable feature selection techniques Deepa et al. [19], SVM with SMOTE technique on imbalanced sample data Almomani et al. [20] as the accuracy of 88.75% G-mean around 98%, respectively. A Two-phase detection approach was used, namely creating a Markov model for extracting features. Then classification was done by Random Forest with 97% Hwang et al. [21], and 95.6% Garcia et al. [22] detection rate, respectively.

Finding ransomware samples using single metadata in addition to multiple datasets comprises ten kinds of ransomware groups implemented on CICAndMal2017 by F. Noorbehbahani et al. [23]. Among various models, the Random forest classification model attained maximum effectiveness in detecting ransomware.

Table 1 compares and evaluates the taxonomy of ransomware detection using traditional methods.

Table 1. Taxonomy of ransomware detection using traditional methods

Researcher	Year	Dataset	Scope	Techniques Applied	Accuracy
F. Noorbehbahani et al. [23]	2019	CICAndMal2017	Single ransom and multiple ransom families	Random Forest	92%
Zhang et al. [9]	2019	Real dataset	Chosen N-grams features via TF-TDF	Random Forest	91.43%
Deepa et al. [19]	2015	Android dataset	Finding malware Android files	AdaBoosting	88.75%
Hwang et al. [21]	2020	Ransomware samples	Markov model to capture ransom features	Two-stage detection approach	97.3%
Zuhair et al. [14]	2020	Static features	0-day ransomware detection	Streaming analytics method	97%
Seong et al. [13]	2019	Ransomware samples	CF-NCF for feature selection	KNN	96.6%
Garcia et al. [22]	2016	Malware images	10-fold cross-validation	Random Forest	95.2%
Tobiyama et al. [24]	2016	Malware images	Extracting features using RNN	CNN	96%
Abien et al. [25]	2019	Malimg dataset	Classification of malware	GRU, along with SVM	84.2%
Almomani et al. [20]	2021	Various open resources dataset	Android ransomware detection	SVM, along with SMOTE	G-mean score as 97.5%
Bibi et al. [16]	2019	Android malware CICAndMal2017 dataset	19 features used via majority voting procedure	LSTM model for detecting android based ransomware	97%
Subash et al. [27]	2019	Samples of ransomware	Preprocessing done by feature reverse engineering process	AdaBoost with RF	97%
Kirubavathi et al. [15]	2023	331 permission dataset	Detection of Android ransomware based on behaviours	Decision tree	98.1%
Proposed model	2024	GitHub 2020 ransomware	Normalization, Feature selection, machine learning models	Hybrid (RF+DT)	99.5%

We noted that from the above table, few authors employed deep learning techniques in detecting ransomware. Tobiyama et al. [24] developed a Support vector machine Abien Fred et al. [25] based on hyperplanes. Here, deep based Recurrent Neural Network was used to extract features from malware images and finally fed into CNN to train the parameters for further classification. The main drawbacks of using such a deep-based model are that it increases computational complexity while training the selected features and enhances processing time. Due to their high false positive rate, existing malware detection techniques, such as statistical-based prevention techniques, cannot stop the emerging ransomware. Hence, we also used eccentric machine learning algorithms to reduce complexity and processing time.

3. Research Methodology

In this section, the authors discussed the overall workflow implemented for finding ransomware malware and two-class categorization (class 0 and class 1). Furthermore, pre-processed has done, how the features have been chosen to predict the specific malware that affects the networks and how to prevent victim's files and devices using conventional techniques executed under the prediction phase. Finally, evaluation was done among developed algorithms based on performance metrics; in that way, model performance was also evaluated.

3.1. Dataset Description

The authors have gathered ransomware data samples from open resource websites. The Github link is mentioned below. https://github.com/muditmathur2020/RansomwareDetection/tree/master Here, the metadata comprises 1,38,047 data samples by which samples are split into ransomware (96,724) samples as Class 0, whereas legitimate (41,323) samples are considered as Class 1. Moreover, the distribution of overall data samples based on two classes is depicted in Figure 2.

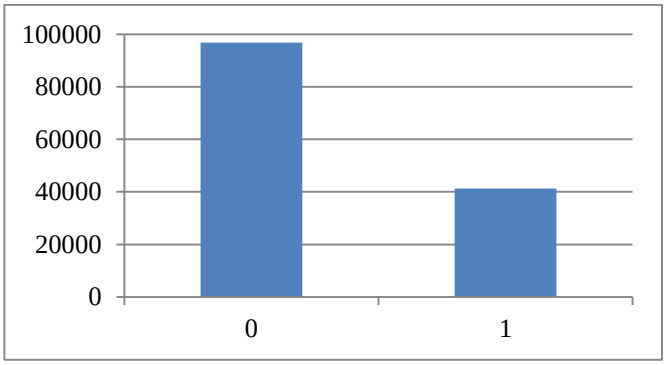

Fig. 2 Distribution of ransomware data

3.2. Data Preparation/ Pre-Processing

The initial phase of processing the data indicates Data pre-processing. The goal of pre-processing data is to convert input data samples in unprocessed format into one that will be simpler and more useful for subsequent processing stages. Applying the Min-Max approach, researchers standardize data in the initial phases. Since every value in the experimental information has a similar scale. For instance, all data samples lie between 0 and 1 where normalization can speed up the procedure for training. To do this, we carry out (1) in the manner described in Tohari Ahmad [26], where Xnorm represents the normalized result and X is the original value prior to normalization. The maximum and minimum values of each characteristic are shown Equation (1) herein Xmax and Xmin, correspondingly.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

3.2.1. Feature Selection

Finding the most suitable feature within metadata represents the goal of feature selection. The input data can be categorized by applied machine-based algorithms to the collection of class ransomware and legitimate features and criteria. Feature domains for machine learning or pattern recognition applications have expanded between hundreds to dozens of possible parameters or features. A number of strategies have been created to address the issue of eliminating superfluous and insignificant factors. Variable reduction through choosing features enhances efficiency, lowers processing prerequisites, lessens the impacts of the dimensionality curse, and aids in dataset understanding. By fixing the threshold value as 0.01, the data samples are notified as ransomware and legitimate. If the sample value is less than the threshold value, it is considered legitimate; otherwise, it is considered ransomware malware. The selected features with a standardized format, in addition to a few rows of normalized training data, are listed in Table 2.

Table 2. Selected features with a normalized value

S. No	Selected features	Normalized value depends on the threshold
1.	Machine	-0.3599
2.	Size of Optional Header	-0.3593
3.	Characteristics	-0.5101
4.	Major Linker version	-1.6307
5.	Minor Linker version	4.4161
6.	Size of code	0.14265
7.	Size of Initialized data	-0.01965
8.	Size of UniInitialized data	-0.006401
9.	Address on the Entry point	-0.04296
10.	Base of code	-0.00956
11.	Export Nb	-0.09351
12.	Resources Nb	-0.1201
13.	Resources Mean Entropy	1.1078
14.	Resources Min Entropy	1.3785
15.	Resources Max Entropy	1.5411
16.	Resources Mean size	-0.0071
17.	Resources Min size	-0.0030
18.	Resources Max size	-0.00286
19.	Load Configuration size	-0.01866
20.	Version Information size	0.5356

Hereby, 56 features are selected for finding ransomware malware that provides huge harm to the devices of the computer system. After selecting features based on standardization, the set size and label size of both training and testing samples are (1,10,437, 56), (27,610, 56), (1,104,437, 0) and (22610, 1).

3.3. Training: Testing Split

After feature selection, the samples are split into two phases, namely training and testing, with a ratio of 80:20 shown in Figure 3. Here, 80% of data samples are trained to predict ransomware that affects devices and files, whereas 20 data samples are validated to evaluate the ML performance in detecting ransomware and classifying ransomware as legitimate.

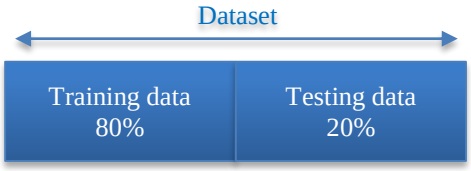

Fig. 3 Splitting of dataset

3.4. Proposed Architecture for Detecting Ransomware

The architecture of ransomware detection was introduced by Subash et al. [27]. ASM and Dynamic link libraries are utilized for the classification of ransomware. Five stages are involved in the suggested ransomware identification process, as shown in Figure 4.

- Data Acquisition
- Data Preprocessing
- Feature Selection
- Creating models,
- Performing validation

The Python programming language was used to retrieve the dataset and preprocess it. The dataset was subjected to feature selection techniques. Python programming was used to normalize the dataset using a standard scalar function. 20% of the dataset was allocated to the testing set and 70% to the training set. Six Machine learning algorithms, along with hybrid models, operated on the training set, which is a ransomware dataset. The testing dataset is the one that we utilize to evaluate our model's accuracy.

3.5. Building ML Classifiers

Machine learning, at its core, uses algorithms that are programmed to take and process incoming data and forecast values within a given range. Sorting data according to a predefined class or label is known as classification. Thus, supervised learning is the category into which algorithms for solving classification problems are divided. As per this study, the effects of our proposed framework on the effectiveness of classification are evaluated using six different classification algorithms.To validate the effectiveness of the machine learning classification model for detecting ransomware, the authors executed metadata around 1 38,047 malware samples to confirm malevolent nodes and also advised the cloud server to obstruct such nodes. Various conventional classification models like AdaBoost, KNN, SGD, ANN, Extra trees and Hybrid models were executed to identify the ransomware that affects devices, files and records.

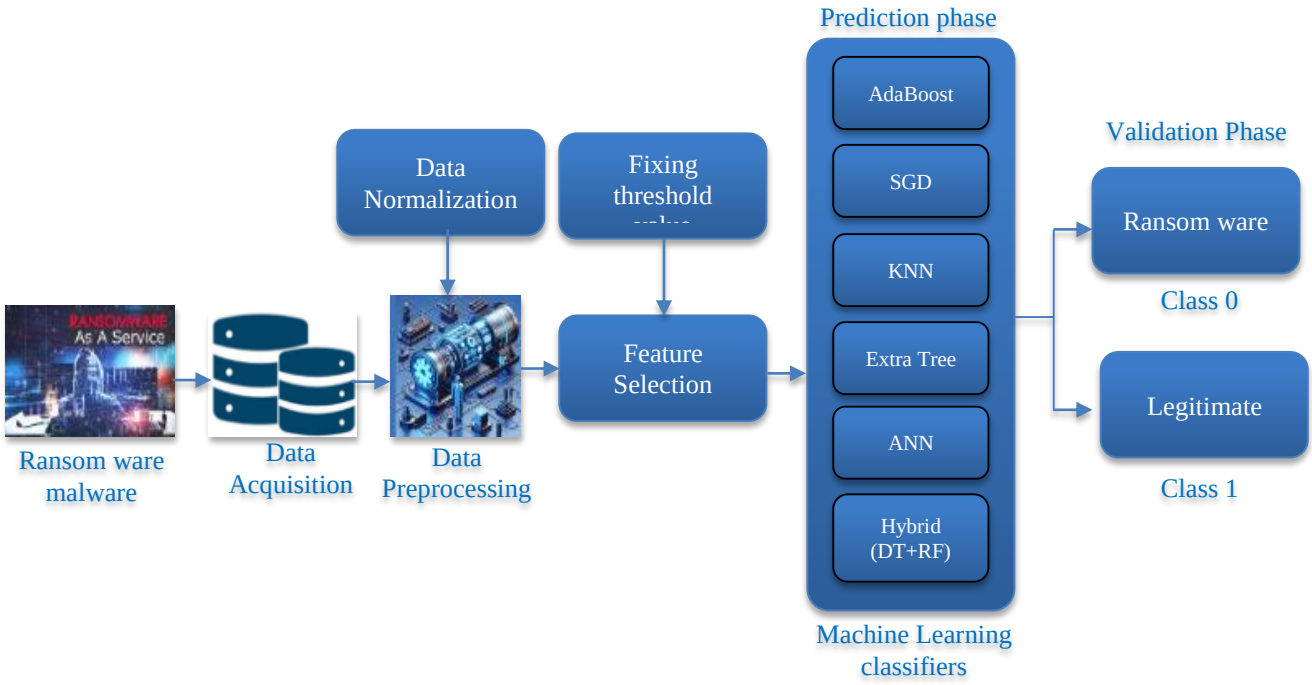

Fig. 4 Framework for detecting ransomware

3.5.1. AdaBoost Model

Boosting is a popular method for transforming an ineffective learnerinto an effective learner to acquire classifiers. By improving the forecasting capabilities of the weak classification algorithm, the boosting technique aims to create a very advantageous classifier by starting with a weak classifier. The outcomes of numerous weak classifiers are equalized in order to prepare this forecast. Adaptive Boosting, or AdaBoost, is a well-liked boosting technique that targets classification issues by constructing a powerful classifier from a large number of weak classifiers Khan et al. [28]. The process involves creating a prototype using the training set of data, followed by constructing a second model that aims to address the shortcomings of the first model.

Models are added until either the maximum number of models or the training set is determined.AdaBoost is a widely used technique for improving decision tree output on binary classification tasks. The AdaBoost algorithm was selected because it helps to improve the efficiency of machine learning techniques. When it is done with weaker learners, it is usually excellent. One-level decision trees are the traditional algorithm used with AdaBoost. The trees are called decision stumps because they are small and only store one choice for a class. Weak models are generated using the weighted training ransomware data and concatenated one after the other. The process keeps going until either a certain number of poor learners are generated or the training dataset can no longer be improved upon. Hereby, the week models are evaluated using the error rate given in Equation (2)

$$\varepsilon_t = P_{r\epsilon D_t}[h_t(X_i) \neq Y_i] \quad (2)$$

KNN

Based on the learning data closest to the item, the k-NN classifies data samples to predict ransomware. The goal of this approach is to categorize ransomware malware using training samples along with its features. Comparable to the clustering technique, which groups new data according to its distance from existing data or its nearest neighbor, it is extremely straightforward and easy to use. It must first ascertain the significance of the surrounding k (neighbor) prior to calculating the distance among the data and its nearest neighbor.

Next, the Euclidean formula (5) is applied to determine the separation between two points, i.e., the training and testing locations. The Euclidean distance is represented by the formula Eucldist (x, y) and is shown as Equation (3), where 'a'represents starting data, 'b' denotes the subsequent data of a feature, also 'n' represents the number of features chosen after performing the feature selection phase.

$$Eucl_{dist}(x,y) = x_0 + \sum_{i=0}^{n}(a_i + b_i)^2 \quad (3)$$

3.5.2. ML-based Extra Tree

Extra Trees classifier is a straightforward ensemble training method based on selection trees. Using preprocessed features from metadata, a huge number of unpruned selection trees are created for the Extra Trees computation Fadare et al. [29]. The Extra Trees computations are similar to random forest computation and will randomly test the key features at every split purpose of the selection tree. Using randomized selections of highlights, Extra Trees creates a large number of trees and associated hubs, much like Random Forest. In Extra Trees, inconsistency comes from the randomized bits of everything being equal rather than from upgrading data. Extra trees function by increasing propensity and reducing change at the same time. The computation of Extra Trees selects an intersection point arbitrarily.

3.5.3. Stochastic Gradient Descent

This model is an effective continuous classification training technique. It is necessary to substitute a less accurate gradient approximation for the actual item. Stochastic gradient descent provides a gradient to each step in the learning process, allowing one to estimate the gradient of the cost function.

Various adjustments were made to the parameters in order to accommodate the anticipated fluctuations. The model parameters were modified each time additional training data was added. When working with large datasets, stochastic gradient descent produces significantly better results than the traditional method.

The basic form of SGD is mentioned in Equation (4):

$$\theta^{(t+1)} = \theta^{(t)} - \alpha_t \nabla l_i(\theta^t) \quad (4)$$

't' indicates the number of iterations. These illustrate the training package's sizes to adjust the parameter's value.

ANN

The collected features were classified using an artificial neural network (ANN)Dreiseitlet al. [30] by getting input signals represented as numbers (x1, x2, ….xn) or received output from preceding layers converted into two classes, namely 0 and 1. The weighted sum can be evaluated using Equation (5) based on the weight (w1, w2, w3….wj) which defines hyperplane

$$\sum_j w_j * x_j + b \quad (5)$$

Here, the authors assigned 30 nodes by which data samples are considered for every node, two hidden layers are created where 50 nodes are allocated to each hidden layer. Finally, the output layer comprises only two nodes namely class 0 as ransomware and class 1 as legitimate. The investigation's ANN's architecture is depicted in Figure 5.

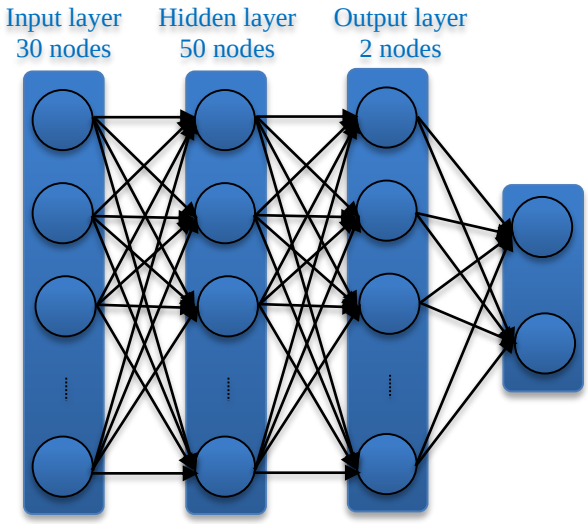

Fig. 5 Framework for ANN

The implementation of the ANN model using Python language is mentioned below:

```
ann_model = Sequential()
ann_model.add(Dense(64,
input_dim=X_train_normalized.shape [1], activation='relu'))
ann_model.add(Dense(32, activation='relu'))
ann_model.add(Dense(1, activation='sigmoid'))
ann_model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy'])
ann_model.fit(X_train_normalized, y_train, epochs=10,
batch_size=32, verbose=1)
```

Hybrid Model

Here, the hybrid model comprises the integration of random forest and decision tree models to enhance the performance in the prediction of ransomware. The model having a place for supervised machine learning type indicates a decision tree. By looking for a single component at each hub, it tries to divide the available ransomware data into smaller subgroups. The computation keeps dividing the dataset into smaller and smaller chunks until each observation in a given subset belongs to a single class-in this case, "legitimate" or "ransomware". The information gained, or the Gini record is used to complete the test directed at each split. An enhancement of ID3, C4.5, uses an extra percentage augmentation to data growth.

The process of bagging technique is further upon by Random Forest. Each classifier in bagging is built independently by utilizing a bootstrap test of the data. A decision at the center splits in a standard decision tree classifier based on each element the classifier assigns. Regardless, Random Forest generates the optimal border at every center in a decision tree by randomly selecting an appropriate amount of spotlights. In addition to helping random forest models scale effectively in situations when there are several highlights per inclusion vector, this randomized feature selection additionally assists the models in weakening the association (connection) among the component credits.

4. Validation

During the validation phase, the implemented machine learning algorithms are evaluated to predict ransomware malware, a kind of malicious software that demands payment to the attacker in the form of a ransom in order to release data or prevent access to a computer system. The attacker typically encrypts the data in order to achieve this.

5. Metrics Evaluation

To evaluate the performance of six machine learning classifiers, the authors calculated metrics such as accuracy, precision, recall, F1-score, Cohen's kappa score and Mathew's correlation coefficient.

For appraisal reasons, we used the Overall Accuracy (OA), False Positive, False Negative, True Positive, True Negative, Recall, Precision, F1-Score, Cohen Kappa, Accuracy and MCC. False Negatives (FN): the number of malicious samples classified as benign. True Negatives (TN): the number of benign samples classified as benign. False Positive (F.P) implies wrongly classifier favorable as malware. True Negative (T.N) means correctly classifying benign as benign.

1. Accuracy: Accuracy can be defined as the ratio of truly predicted samples as ransomware from the entire input data samples. It can be formulated as Equation (6)

$$\text{Accuracy} = \frac{\text{Number of samples truely predicted as ransomware}}{\text{Overall input data samples (TP+FP+TN+FN)}} \quad (6)$$

2. Precision: The ratio of correctly predicted positive to total expected positive data samples is known as precision. Percentage of positive identifications that was, in fact, accurate. Precision can be formulated by Equation (7)

$$\text{Precision} = \frac{\text{Truely predicted as positive}}{\text{True Positive+False Positive}} \quad (7)$$

3. Recall: A calculation's recall is its ability to find every positively predicted sample from overall samples. A True Positive rate is equal to a recall. The percentage of real findings that were accurately detected. Recall can be evaluated using Equation (8)

$$Recall = \frac{Truely\ predicted\ as\ positive}{True\ Positive+False\ Negative} \quad (8)$$

4. F1-score-F1-score predicted the harmonic mean of both precision and recall value. Since the characterization measure's accuracy and recall are two opposing metrics, using F1-Score may appropriately regulate the precision and review. Additionally, the more similar one of the F1-

Score theoretical worth approaches, the higher the classifier's performance (9).

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

5. Cohen's Kappa score: The statistical metric that assesses the degree of concurrence among two raters who independently categorize data samples into ransomware or legitimate. The range of the kappa score lies between -1 to +1. It can be evaluated using Equation (10)

$$k = \frac{2*(TP*TN - FN*FP)}{(TP+FP)*(FP+TN)*(TP+FN)*(FN+TN)} \quad (10)$$

6. Mathew's correlation coefficient: The best single-value classification metric for reducing an error or confusion matrix is Matthew's correlation coefficient (MCC). This can be evaluated using the formula Equation (11)

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)*(FP+TN)*(TP+FN)*(FN+TN)}} \quad (11)$$

7. The model is surveyed through evaluation measures like precision, recall, F1-score, Accuracy, Cohen kappa score, and MCC.

6. Experimental Outcomes

6.1. Experiment Setup

The machines used in this experiment have the identical setup, and we have implemented the experiment using the Python programming language: Computer specifications: Intel Core-i7 2.40 GHz (4 CPUs), 8 GB RAM, 64-bit Microsoft Windows 10.

6.2. Comparison of Accuracy among Different Models

The plotted graph shows the accuracy measures for all machine learning algorithms. Here, Adaboost achieved accuracy for AdaBoost as 98.84, KNN reached 99.09%, the Extra tree model attained 99.5%, SGD was 98.13%, ANN 99.07%, and the hybrid model integrated with Random Forest and Decision tree attained accuracy as 99.51% depicted in Figure 6. Among all six classification models, the hybrid and extra tree models attained maximum accuracy in detecting ransomware that harms devices and files.

6.3. Assessment of F1 Score among Conventional Approaches

The plotted graph shows the F1-score measures for all machine learning algorithms. Here, Adaboost achieved an F1-score of 98.08%, KNN reached 98.51%, the Extra tree model attained 99.18%, SGD was 96.91%, ANN was 98.47% and the hybrid model integrated with Random Forest, and the Decision tree attained F1-measure as 99.19%. Among all six classification models, the hybrid model attained the maximum score in detecting ransomware that harms devices and files, illustrated in Figure 7.

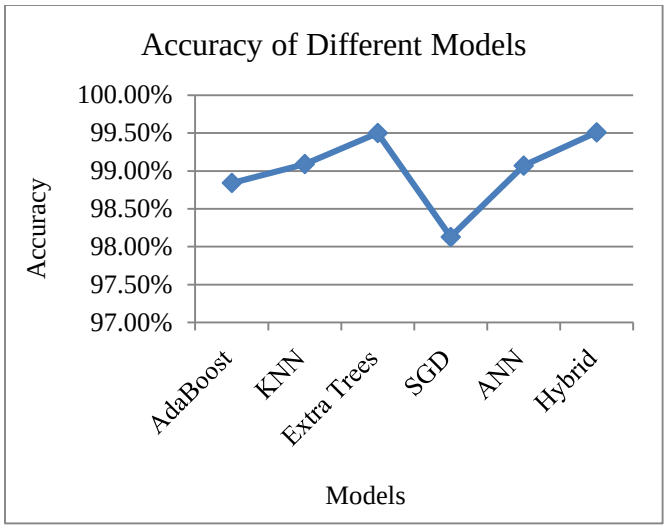

Fig. 6 Assessment of ML classifiers in terms of accuracy

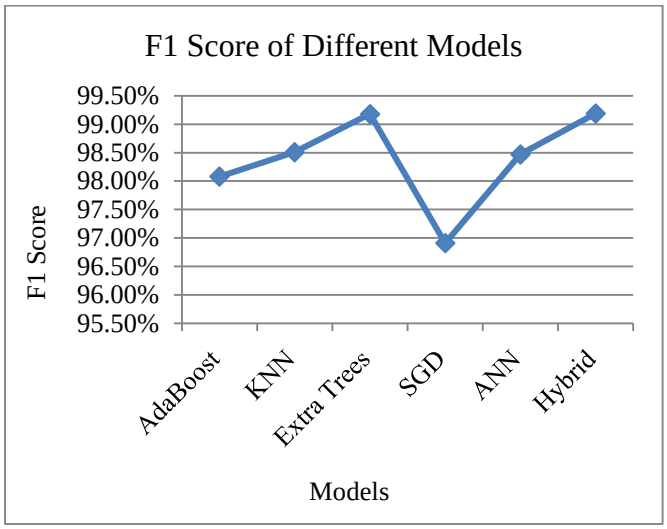

Fig. 7 F1-score comparison among various ML models

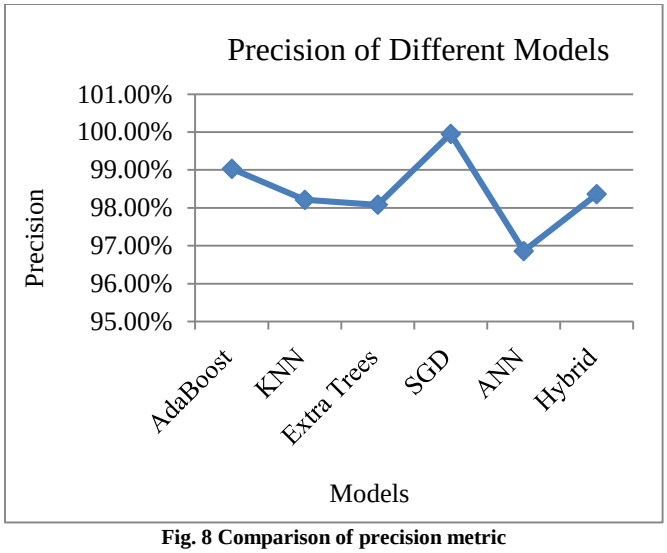

Fig. 8 Comparison of precision metric

6.4. Assessment of Precision

The precision metrics for each machine learning algorithm are displayed on the plotted graph. In this case, the hybrid model integrating Random Forest and Decision tree obtained the highest precision of 99.03%, whereas Adaboost attained 98.21%, KNN attained 98.08%, Extra tree model acquired 99.95%, SGD attained 96.86%, and ANN attained 98.36% depicted as Figure 8.

Out of the six classification models, the hybrid model achieved the highest score for identifying ransomware malware, which hacks various files and edge devices.

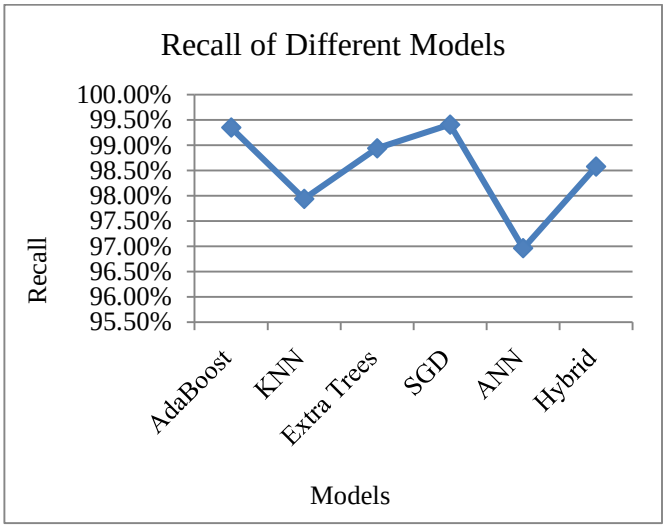

Fig. 9 Recall metric versus machine learning models

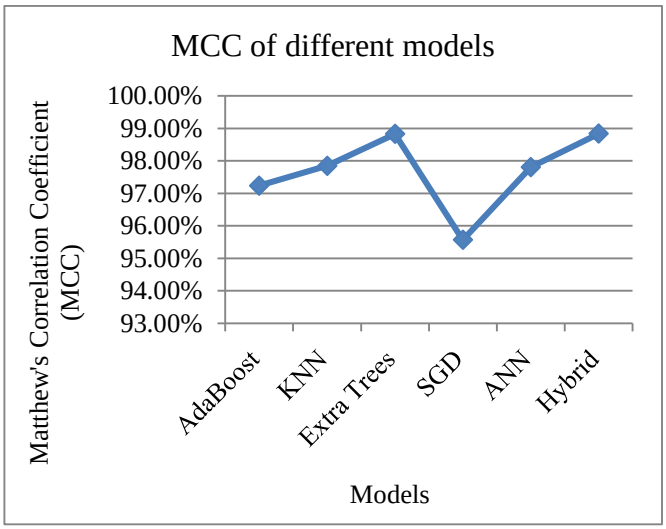

Fig. 10 MCC evaluation on different ML models

6.5. Evaluation of Recall Metric

The recall metrics for each machine learning algorithm are displayed on the plotted graph. In this case, the hybrid model integrating Random Forest and Decision tree obtained the highest recall of 99.35%, whereas Adaboost attained 97.94%, KNN attained 98.94%, Extra tree model acquired 99.41%, SGD attained 96.96%, and ANN attained 98.58%. Among those models, the hybrid model achieved the highest recall score for identifying ransomware, the attackers who provide harm to edge devices shown in Figure 9.

6.6. Assessment of MCC Score

The plotted graph shows the accuracy measures for all machine learning algorithms. Here, Adaboost achieved an MCC score for AdaBoost at 97.24%, KNN reached 97.85%, the Extra tree model attained 98.83%, SGD at 95.57%, ANN at 97.81%, and the hybrid model integrated with Random Forest and Decision tree attained accuracy as 98.84%. Among all six classification models, the hybrid and extra tree models attained maximum MCC in ransomware detection, as shown in Figure 10.

6.7. Cohen's Kappa Score

The plotted graph shows the accuracy measures for all machine learning algorithms. Here, Adaboost achieved a kappa score for AdaBoost as 97.24, KNN reached 97.85%, Extra tree model attained 98.83%, SGD as 95.57%, ANN as 97.81% and hybrid model integrated with Random Forest and Decision tree attained accuracy as 98.84%. Among all six classification models, the hybrid and extra tree models attained the maximum Cohen's kappa score in the detection of ransomware, portrayed in Figure 11.

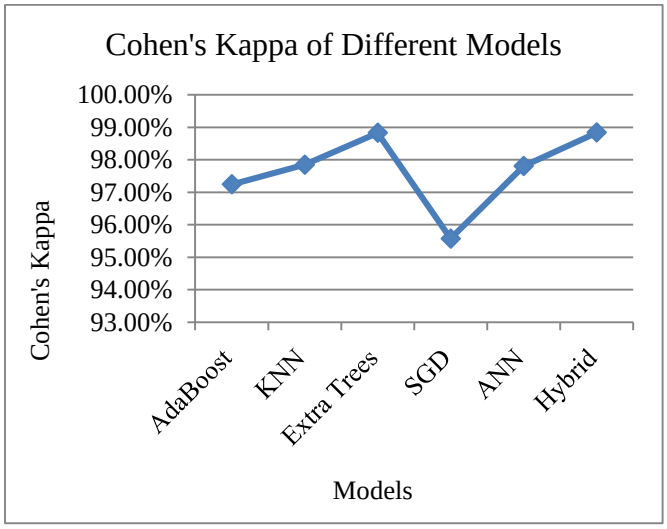

Fig. 11 ML models Vs Cohen's kappa score

6.8. Hybrid (Random Forest + Decision Tree)

During validation, the performances of conventional models are analyzed to detect ransomware malware based on execution/processing time, losses, accuracy, and the number of epochs iterated with features. As a result, the hybrid model processed the ransomware feature in less execution time and with the least losses at 2ms and 0.02, respectively.

Moreover, the attained accuracy for ransomware detection and classification by this model 99.53%described in Table 3.

Table 3. Evaluation of machine based hybrid model in terms of execution time, loss and accuracy

Epochs	Execution Time (ms)	Loss	Accuracy in (%)
1	3	0.053	98.44
2	2	0.0359	98.85
3	3	0.0321	98.95
4	2	0.0298	98.97
5	3	0.0284	99.03
6	2	0.0285	99.06
7	3	0.0260	99.10
8	2	0.0249	99.13
9	3	0.0241	99.2
10	2	0.0234	99.53

6.8.1. Assessment of Conventional Model Versus Metrics Value for Ransomware Detection

Table 4 shows overall comparison among various machine learning-based algorithms for ransomware prediction and classification in terms of accuracy, precision, recall, F1-score, MCC and Cohen's kappa score. All six metrics are considered for evaluating the performance of machine-based classification models in recognition of ransomware, depicted in Figure 12.

Table 4. Overall assessment of ransomware detection

Implemented Algorithms	Accuracy	Precision	Recall	F1-Score	MCC	Cohen's Kappa
AdaBoost	0.9884	0.9821	0.9721	0.9808	0.9724	0.9724
KNN	0.9909	0.9808	0.9894	0.9851	0.9785	0.9785
Extra Tree	0.9950	0.9895	0.9941	0.9918	0.9883	0.9883
SGD	0.9813	0.9686	0.9686	0.9691	0.9557	0.9557
ANN	0.9907	0.9836	0.9858	0.9847	0.9781	0.9781
Hybrid(RF+DT)	0.9951	0.9903	0.9935	0.9919	0.9884	0.9884

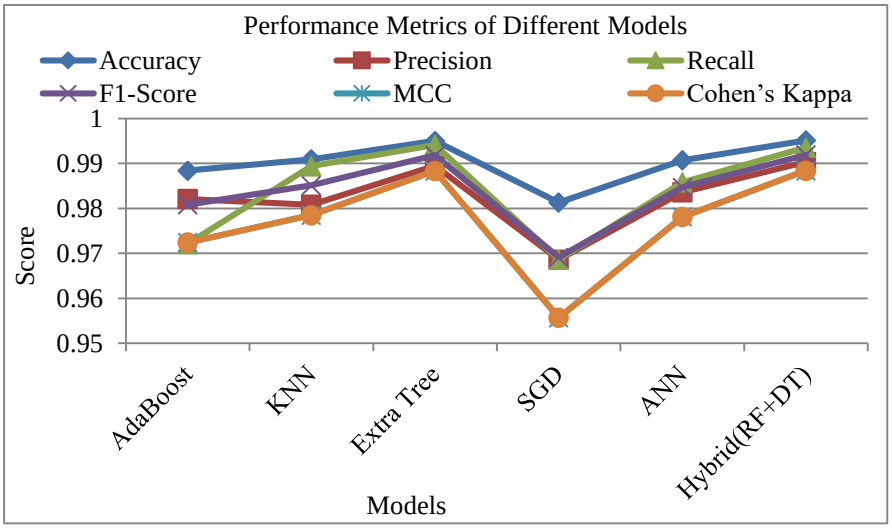

Fig. 12 Performance comparison metrics on Ransomware detection

7. Conclusion

An enhanced effectiveness for detection is crucial since ransomware is becoming more and more prevalent. The purpose of this work is to attempt to help accomplish this goal. In order to lower the false positive rate in detection, many machine learning algorithms were used to gather, analyze, and evaluate ransomware samples gathered from the GitHub link. The dataset was trained and tested using machine learning models in order to improve the dataset's ability also to identify the ransomware family of viruses.In summary, the authors implemented six machine learning models on ransomware data samples using Python programming language. The malware samples were trained using machine learning algorithms such as AdaBoost, ANN, SGD, Extra-tree, KNN, and hybrid (Random Forest + Decision tree). Such ML models are evaluated by measuring accuracy, precision, recall, F1-score, kappa score and MCC. Among these classification models, the Hybrid classifier attained maximum accuracy as 99.5% accuracy, 99.03% precision, 99.35% recall, 99.19% F1-score, 98.84% kappa score and 98.84% MCC.

References

[1] J. De Groot, A History of Ransomware Attack: The Biggest and Worst Ransomware Attack of All Time, 2017. [Online]. Available: https://www.digitalguardian.com/blog/history-ransomware-attacks-biggest-and-worst-ransomware-attacks-all-time

[2] Wira Zanoramy A. Zakaria et al., "The Rise of Ransomware," *Proceedings of the International Conference on Software and e-Business*, Hong Kong, pp. 66-70, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] Maersk Ransomware Attack, TechForce, 2017. [Online]. Available: https://techforce.co.uk/blog/2019/maersk-ransomware-attack

[4] Ransomware Payments Exceed $1 Billion in 2023, Hitting Record High After 2022 Decline, Chainalysis, 2024. [Online]. Available: https://www.chainalysis.com/blog/ransomware-2024/

[5] Ben Dickson, The IoT Ransomware Threat is More Serious than you Think, IoT Security Foundation, 2019. [Online]. Available: https://www.iotsecurityfoundation.org/the-iot-ransomware-threat-is-more-serious-than-you-think/

[6] Goteng Kuwunidi Job et al., "Impacts of Ransomware Attacks on Edge Computing Devices: Challenges and Research Opportunities," *International Journal of Engineering Research & Technology*, vol. 10, no. 4, pp. 665-670, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid, "Ransomware Threat Success Factors, Taxonomy, and Countermeasures: A Survey and Research Directions," *Computers & Security*, vol. 74, pp. 144-166, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jinal P. Tailor, and Ashish D. Patel, "A Comprehensive Survey: Ransomware Attacks Prevention, Monitoring and Damage Control," *International Journal of Research and Scientific Innovation*, vol. 4, no. 6s, pp. 116-121, 2017. [Google Scholar] [Publisher Link]

[9] Hanqi Zhang et al., "Classification of Ransomware Families with Machine Learning Based on *N*-Gram of Opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211-221, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Aaron Zimba, "Malware-Free Intrusion: A Novel Approach to Ransomware Infection Vectors," *International Journal of Computer Science & Information Security*, vol. 15, no. 2, pp. 317-325, 2017. [Google Scholar] [Publisher Link]

[11] Amjad Alraizza, and Abdulmohsen Algarni, "Ransomware Detection using Machine Learning: A Survey," *Big Data and Cognitive Computing*, vol. 7, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Nisreen Alzahrani, and Daniyal Alghazzawi, "A Review on Android Ransomware Detection using Deep Learning Techniques," *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, pp. 330-335, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Seong Il Bae, Gyu Bin Lee, and Eul Gyu Im, "Ransomware Detection using Machine Learning Algorithms," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Hiba Zuhair, Ali Selamat, and Ondrej Krejcar, "A Multi-Tier Streaming Analytics Model of 0-Day Ransomware Detection Using Machine Learning," *Applied Sciences*, vol. 10, no. 9, pp. 1-23, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] G. Kirubavathi, S. Sreevarsan, and P. Varadhan, "Behavioural Based Detection of Android Ransomware Using Machine Learning Techniques," *Research Article*, pp. 1-34, 2023. [CrossRef] [Publisher Link]

[16] Iram Bibi et al., "An Effective Android Ransomware Detection through Multi-Factor Feature Filtration and Recurrent Neural Network," *UK/China Emerging Technologies*, Glasgow, UK, pp. 1-4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Samah Alsoghyer, and Iman Almomani, "Ransomware Detection System for Android Applications," *Electronics*, vol. 8, no. 8, pp. 1-36, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ashu Sharma, and Sanjay K. Sahay, "An Effective Approach for Classification of Advanced Malware with High Accuracy," *International Journal of Security and Its Applications*, vol. 10, no. 4, pp. 249-266, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[19] K. Deepa, G. Radhamani, and P. Vinod, "Investigation of Feature Selection Methods for Android Malware Analysis," *Procedia Computer Science*, vol. 46, pp. 841-848, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[20] Iman Almomani et al., "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data," *IEEE Access*, vol. 9, pp. 57674-57691, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Jinsoo Hwang et al., "Two-Stage Ransomware Detection using Dynamic Analysis and Machine Learning Techniques," *Wireless Personal Communications*, vol. 112, pp. 2597-2609, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Felan Carlo C. Garcia, and Felix P. Muga II, "Random Forest for Malware Classification," *arXiv*, pp. 1-4, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[23] Fakhroddin Noorbehbahani, Farzaneh Rasouli, and Mohammad Saberi, "Analysis of Machine Learning Techniques for Ransomware Detection," *16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology*, Mashhad, Iran, pp. 128-133, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Shun Tobiyama et al., "Malware Detection with Deep Neural Network using Process Behavior," *IEEE 40th Annual Computer Software and Applications Conference*, Atlanta, GA, USA, pp. 577-582, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[25] Abien Fred Agarap, "Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification," *arXiv*, pp. 1-5, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[26] Tohari Ahmad, and Mohammad Nasrul Aziz, "Data Preprocessing and Feature Selection for Machine Learning Intrusion Detection Systems," *ICIC Express Letters*, vol. 13, no. 2, pp. 93-101, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Subash Poudyal, Kul Prasad Subedi, and Dipankar Dasgupta, "A Framework for Analyzing Ransomware using Machine Learning," *IEEE Symposium Series on Computational Intelligence*, Bangalore, India, pp. 1692-1699, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[28] Muhammad Salman Khan et al., "Fractal Based Adaptive Boosting Algorithm For Cognitive Detection Of Computer Malware," *IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing*, Palo Alto, CA, USA, pp. 50-59, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[29] Fadare Oluwaseun Gbenga, Adetunmbi Adebayo Olusola, and Oyinloye Oghenerukevwe Elohor, "Towards Optimization of Malware Detection using Extra-Tree and Random Forest Feature Selections on Ensemble Classifiers," *International Journal of Recent Technology and Engineering*, vol. 9, no. 6, pp. 223-232, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Stephan Dreiseitl, and Lucila Ohno-Machado, "Logistic Regression and Artificial Neural Network Classification Models: A Methodology Review," *Journal of Biomedical Informatics*, vol. 35, no. 6, pp. 352-359, 2002. [CrossRef] [Google Scholar] [Publisher Link]