


## Research Article

# Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls

**Faizan Ullah,<sup>1</sup> Qaisar Javaid,<sup>2</sup> Abdu Salam,<sup>3</sup> Masood Ahmad,<sup>3</sup> Nadeem Sarwar <sup>4</sup>, Dilawar Shah,<sup>1</sup> and Muhammad Abrar<sup>1</sup>**

<sup>1</sup>Department of Computer Science, Bacha Khan University, Charsadda, Pakistan

<sup>2</sup>Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan

<sup>3</sup>Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan

<sup>4</sup>Department of Computer Science, Bahria University Lahore Campus, Lahore, Pakistan

Correspondence should be addressed to Nadeem Sarwar; [nsarwar.bulc@bahria.edu.pk](mailto:nsarwar.bulc@bahria.edu.pk)

Received 20 April 2020; Revised 20 June 2020; Accepted 6 July 2020; Published 1 August 2020

Academic Editor: Antonio J. Peña

Copyright © 2020 Faizan Ullah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ransomware (RW) is a distinctive variety of malware that encrypts the files or locks the user's system by keeping and taking their files hostage, which leads to huge financial losses to users. In this article, we propose a new model that extracts the novel features from the RW dataset and performs classification of the RW and benign files. The proposed model can detect a large number of RW from various families at runtime and scan the network, registry activities, and file system throughout the execution. API-call series was reutilized to represent the behavior-based features of RW. The technique extracts fourteen-feature vector at runtime and analyzes it by applying online machine learning algorithms to predict the RW. To validate the effectiveness and scalability, we test 78550 recent malign and benign RW and compare with the random forest and AdaBoost, and the testing accuracy is extended at 99.56%.

## 1. Introduction

Computers are now becoming a legal part of our daily life, and the world cannot imagine life without a computer. Internet and computer applications have facilitated our daily life. The development has also brought us several threats to the computer, i.e., malware [1]. Malware is a malicious code, which is composed of two words “Mal” mean malicious and “ware” mean software. Through e-mail, this malicious software sent a link or file, and when the user clicks on the link or opens the file, their malware type viruses, ransomware (RW) and spyware, get executed [2]. Malicious software consists of codes developed by cyber attackers and designed to extensively damage the victim data. There are numerous types of malware but the most common types are spyware, virus, and scareware. The spyware is designed to spy on the users' activities. It is a hidden application that is secretly executed in the background on the victim's computer. This type of malicious software collects information

such as details of credit cards, passwords, and other sensitive information without the user's permission. A computer virus is a common type of malware which attaches itself to victims' other files. It gets downloaded or installed itself in the computer system. The computer virus spreads quickly in the computer system. It also damages the main functionality of the computer systems and corrupts or locks the victim's system and files [3]. The third type of malware is scareware, also acknowledged as RW, that comes with a high price. It is capable to lock or encrypt user data and restrict a user to get access to their data until the demanded money or ransom is paid.

RW attacked some of the largest organizations in the world. It is the main type of malware related to cybercriminals, and it is very common. The aim and objective of this malware are to collect money as a ransom. RW encrypts the files or locks the user's system by holding and taking user files' hostage that leads to financial gain [4]. In today's Internet market, RW is the most dangerous and significant

security threat and is on the top of the list. The history of RW goes back to 1980 [5]. In the last few years, such kinds of attacks are in the headlines around the world. They have resulted in increasing new families, e.g., Cryptowall 3.0 is one amongst the family of RW, which known as costly and effective RW family that had caused around \$325 million damage to the industry. Sony RW attack is also very dangerous which got huge media headlines. North Korea was behind the attack, and US government confirmed it [6–9].

*1.1. Types of Ransomware (RW).* According to the current arrival and weekly arising stories of RW, it is difficult to identify the different strains, as each of them spread differently. They generally follow similar strategies to gain the benefit of users' security weaknesses and hold data hostage [10]. There are several forms of RW in which some of them are discussed here in detail.

*1.1.1. Bad Rabbit RW.* Bad Rabbit is the type of scareware, which is on the top of the list. In Eastern Europe and Russia, RW infected different organizations. The RW spreads itself by showing itself as a fake adobe flash update on compromised websites. When this RW infects a system, the user is directed to the payment page and shows that you are infected or hacked and now you have to pay \$285 [10].

*1.1.2. Cerber RW.* It is the most dangerous and powerful RW because it also works even if you are not connected to the Internet, and even if your PC is unplugged, it still works. Cerber function is to encode the files of infected users, and then if you want to give access to your files back, you need to pay money. It attaches and sends the infected Microsoft Office document through e-mail to the victim's system. Accessing the attached file automatically encrypts the files with Rivest–Cipher (RC4) and Rivest–Shamir–Adleman (RSA) algorithms and updates or modifies them with Cerber extensions [11].

*1.1.3. CryptoLocker RW.* Crypto RW is also a special type of malware. It works like a Trojan horse, which is also used to earn money. It encodes files on the specific system, and the users will be asked to pay to decrypt their files. Through spam emails, Ads, or fake sites or by malicious methods, these threats affect the user system. Thus, once the system is infected by Trojan, it stores the path of encoded files through several registered entries and runs when the system restarts and specific extensions are made in the computer which encrypts the records, and to find the decryption key, it creates additional files. To get the key, this dangerous family tries to convert the user to pay money. They use different kinds of techniques for users to pay the money for ransom [12].

*1.1.4. Cryptowall RW.* Ransom Cryptowall is a Trojan horse type virus that encodes files on the specific computer and asks the user to pay for file decryption. These threats typically

arrive on the affected PC through exploit kits, spam emails held through malware ads or compromised sites, or other malicious. Once the Trojan is entered into the compromised system, it makes several registry entries to store the path of the encoded files and run when the computer restarts. It encrypts the records with specific extensions on the system and creates additional files with instructions on how to find the decryption key. This danger family attempts to convince the user to pay money to get the key to unlocking their documents. It uses different techniques to convince the user to pay the money for ransom [13].

RW is a specialized form of malware that encrypts files and condenses them unreachable until the victim pays a ransom. It is an extremely serious problem, and it is quickly getting worse. The statistics gathered by the FBI's Internet Crime Complaint Center (IC3) for 2018 show Internet-enabled theft, fraud, and exploitation remain pervasive and were responsible for a staggering \$2.7 billion financial losses [14]. The FBI reports the IC3 received 351,936 complaints in 2018 and an average of more than 900 every day. There is a dramatic increase in extortion payments with tens of thousands of ransomware victims paying several hundred dollars each to recover their encrypted files. In some instances, the ransom is larger, such as South Korean web hosting company Nayana, which paid 397.6 Bitcoin (about \$1 million) in June 2017 and Hollywood Presbyterian Medical Center, which paid \$17,000 in Bitcoin in February 2016 [15].

This emerging issue needs the attention of the research community to detect and prevent the families of RW that will protect users from huge losses. In this paper, we proposed a robust solution to detect RW at runtime by monitoring network, registry activities, and file systems. We use the API-call series to represent the behavior-based features of malware. The proposed methodology extracts the 14-feature vector by using runtime analysis by applying online machine learning algorithms for the classification of malware samples in a distributed and scalable architecture.

This paper organized as follows: Section 2 has the literature about recent work on RW classification and detection. In Section 3, we present our proposed methodology in detail. Section 4 has the experiments, dataset used, time of the proposed approach, evaluation metrics, and experimental results. In Section 5, we conclude this paper and outline for future direction.

## 2. Literature Review

In this section, the existing research work done on the detection and classification of RW is analyzed. The summary of the literature on RW with findings is given in Table 1. The existing computational model for detection and classification of RW is summarized in Tables 1 and 2.

Alhawi et al. [16] presented a machine learning- (ML-) based solution for the detection of RW. The dataset was collected from VirusTotal, and both data are the malicious and benign and contain 264 records having 9 RW families and 3 types of benign. Wireshark is used to capture the data and features. T-Shark is used to extract the features. The

TABLE 1: Summary of the literature on RW detection and classification with findings.

Ref	Dataset	Techniques	Accuracy (%)	Pros	Cons
[16]	VirusTotal	Decision tree + RF + Bayes network	97.1	The well-organized flow of the research. The results are compared with 7 algorithms.	Few families of RW used in experiments.
[17]	VirusTotal	RNNs	96	Recurrent neural network used with convolutional layers.	Training an RNN is a very difficult testing.
[18]	VirusTotal	Decision tree + RF + $k$ -nearest neighbour (K-NN) + naive Bayes	97.3	Performs well on large datasets.	Decision trees are prone to overfitting.
[19]	Malware-traffic analysis .net	RF + (J48)	93	RF performs sound with both continuous variables categorical data.	RF needs much more time to train.
[4]	VirusTotal	RF + J48 + logistic regression + naive Bayes	97	Involves a small amount of training data for classification	Assumption class conditional independence.
[20]	VirusShark	RF + J48	99.5	RF can be used to solve both classifications as well as regression problems.	RF is complex and much computational resources involved.
[21]	VirusShare	RF + hidden Markov models	98.4	Strong statistical foundation.	HMM often have a large number of unstructured parameters.
[22]	VirusShark	Regularized logistic regression + SVM + naive Bayes	96.3	Give good results even semistructured and unstructured data like images, text, and trees.	Difficult to understand variable weights and individual impact.
[23]	VirusTotal and VirusShare	RF + decision tree	97.95	Random forest is usually robust to the outliers.	Need to choose the number of trees.
[24]	VirusTotal	SVM	97.48	SVM compared with ANN. SVMs give better results.	Long training time for large datasets.
[25]	VirusTotal	ANN + SVM	97.8	Store information on the whole network.	ANN requires processors with parallel processing power.
[26]	Malware-traffic analysis .net	Deep neural network, 7 layers	93.92	Creates new tasks to reduce the human intervention.	They cannot make decisions beyond what the machines have been fed.

experiment was carried out in WEKA version 3.8.1. The WEKA machine-learning tool splits a dataset for training and testing purposes. The training dataset contained 75,618 samples, and the test dataset contained 48,526 samples. The training and testing datasets are split as 70 percent and 30 percent, respectively. Six different machine learning algorithms were applied. Using dataset network traffic features, we got a true positive detection rate of 97.1 percent, and using a decision tree classifier, we achieved a zero false positive rate (FPR) and true positive rate (TPR) of 96.3 percent.

Rhode et al. [17] carried out a study for the detection of RW. To achieve high accuracy, the author presented a novel approach. The proposed algorithm detects RW files during the execution stage in the first 20 sec. The dataset was collected from VirusTotal and VirusShare. The dataset contains 23,145 benign and 2,286 malicious records. A preprocess was carried out to convert all alphabetic values into numerical range for presenting of RW. Recurrent Neural Networks (RNNs) are applied to predict RW. The accuracy in 5 sec is 94 percent and 10 sec is 96 percent. The minimum false negative rate (FNR) for a model was 4.5 percent and FPR was just 3 percent. The actual value of the model in 20 seconds is 93 percent. The experiment carried was out in Python version 2.7 using Keras to implement the RNN model.

Carlin et al. [20] developed a dynamical analysis with a new detecting cryptomining technique. The dataset consists of 490 samples and is collected from VirusShark. A total of 490 samples, 194 are benign and Cryptomining has 296 HTML files or malicious samples. The RF classifier is used and implemented in WEKA version 3.9. The data will have used 10-fold cross-validation. The best accuracy of RF is 99.05 percent. The FPR is 99.7 percent, and FNR is 98.6 percent.

Carlin et al. [21] emphasized the analysis of low-level opcode, both dynamic and static, to detect the malware on runtime dataset 1,000 labels samples to affect the traditional AV labels. The dataset was collected from VirusShare. The author selected the size modality and facility. 180,000 records are malware, and all records are named by message digest MD5 hash with no other metadata. Data will be preprocessing only 1,000 opcodes with a 1.0 percent margin. The dataset contains 764 benign and 18,827 malicious samples. The counterbased classifier uses RF and implements it in WEKA version 3.8. The best accuracy of the RF is 98.4 percent.

Takeuchi et al. [24] introduced RW detecting using support vector machines (SVMs). The dataset consists of 588 samples, which have 312 benign and 276 RW, and was collected from VirusTotal. The authors design different sequence of API calls into the same vector symbols. The

TABLE 2: Summary in detail of the existing literature (detection, classification, and computational models for RW).

Ref.	PC				Mobile			Data Source				Statistics				Machine learning based							Outcome				Accuracy				Time Measure		
	Network	Windows	Mac	Linux	Raspberry	Android	iOS	Cloud	Dynamic analysis	Static analysis	Homospor	Statistic	Random forest	Decision tree	Dimension reduction	Support vector machine	Clustering	Deep learning	Ensemble	Neural network	Regularization	Rule system	Regression	Binary classification	Multiclass classification	Clustering	Accuracy	True positive rate	False positive rate	True negative rate		False negative rate	
[16]				x										x										x			x					x	
[17]																																	
[18]		x							x																								
[19]		x																															
[4]																																	
[20]																																	
[21]																																	
[22]																																	
[23]																																	
[24]																																	
[25]																																	
[26]																																	
Total	0	3	0	1	0	0	0	0	4	4	0	1	5	1	0	1	0	0	0	3	0	0	1	5	0	0	5	1	5	0	0	1	

author tested and trained the data form the SVM classifier. The standard accuracy of the vector symbol is 93.52 percent, and the best accuracy of SVM is 97.48 percent.

### 3. Methodology

In this section, the new methodology is discussed. The main objective of the new methodology is the detection of the RW family at runtime. The dataset used in this paper is collected from a virus's total website [27]. VirusTotal is an online provision that examines the files and uniform resource locations (URLs) to help in the detection of worms, viruses, and other kinds of malicious gratified using website scanners and antivirus engines. The dataset is used to identify benign and malware from the data. The proposed methodological model has different phases as shown in Figure 1.

First, the selected dataset is processed. The second phase is used to extract useful features from the preprocessed dataset using API calls. In the third phase, the dataset is divided into testing and training subsets. Finally, for the classification purpose, three diverse machine-learning algorithms, i.e., modified decision tree, random forest, and AdaBoost, are used.

**3.1. Data Sets.** The dataset is collected from the VirusTotal. It consists of 78550 samples; among them, there are 35369 malware and 43191 benign.

The dataset has a total of 18 features, and we select 14 features that are most relevant for the classification of a file in malware or benign. For the accuracy and improvement of the result, the 10-fold cross-validation technique is applied to the data [27].

**3.2. Feature Extraction.** In this step, we extract 14 features from the dataset. The detail of these features given in Table 3. The file names and MD5 hash features are dropped from the dataset. The last feature will be used as a class label, i.e., benign or malware.

**3.3. Training and Testing.** After extracting all vector's features, we utilized the feature vectors with class labels to train the model. Then, the trained classifiers can calculate the labels of new instances in the form of feature vectors. Later, the performance of the proposed model is calculated. In this research, we utilized three different machine-learning algorithms, namely, decision tree, random forest, and AdaBoost.

**3.4. Classification.** During classification, the dataset is split into training and test datasets. This process has a key role in the field of RW detection and ML. The set training is used to train the model, and the test set is used to validate the model results.

**3.4.1. Modified Decision Tree.** Algorithm 1 is used to split a huge collection of records into continuously smaller subsets of records by applying a sequence of simple decision rules.

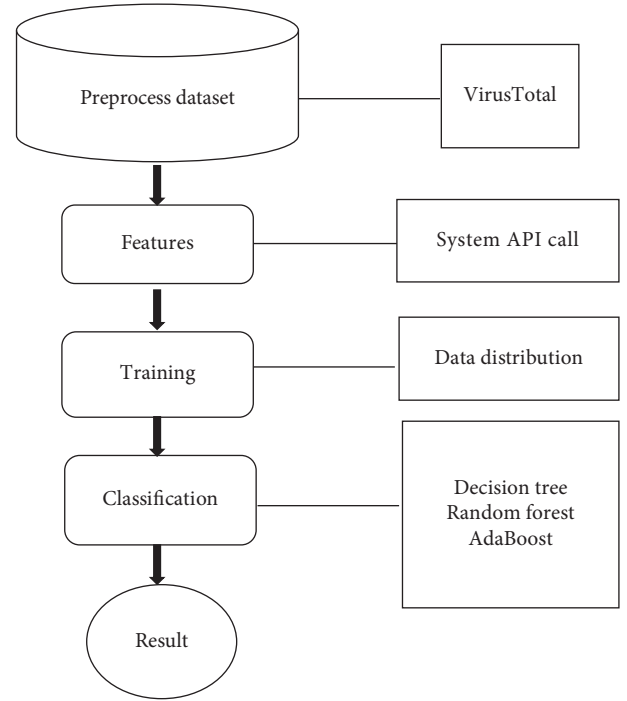


FIGURE 1: Model representation.

The algorithm 1 splits the feature space into subsets where each subset consists of a homogeneous group of samples [28]. The outcome is a tree with leaf nodes and decision nodes. The topmost decision node in a tree, which corresponds to the best predictor, is called the root node. Decision trees can handle both categorical variables and numerical data [29].

The decision tree uses the information gain theory to select the best partitioning attribute from the dataset. The info ( $X$ ) is calculated using (1):

$$\text{info}(X) = - \sum_{i=1}^n P(x_i) \log_2 P(x_i). \quad (1)$$

The key advantage of the decision tree is its' easy implementation. Decision trees and the underlying principle that they work on are easy to interpret and understand as compared with other complex machine-learning algorithms.

**3.4.2. Random Forest.** Algorithm 2 is a combination of different decision trees, each with the unique nodes, but utilizing diverse data that leads to different leaves Figure 2. It combines the decisions of multiple decision trees to find the best answer, which denotes the average decision trees [4]. Random forest is a flexible, easy to practice machine-learning algorithm that generally generates, even without hyperparameter tuning, an improved result. It can be used for both regression and classification problems [30].

**3.4.3. AdaBoost.** AdaBoost stands for adaptive boosting and combines weak classifiers into a strong classifier. Adaptive boosting is the first practical learning technique for building

TABLE 3: Most relevant features for classification with description.

S. no	Attributes/features	Description
1	Debug size	Debug is detecting and takes away errors from the computer system. Debug stands for the size of the debug directory table. Typically, Microsoft-executable files have a debug directory. Therefore, many benign applications may have a positive value for debug size.
2	DebugRVA (debug relative virtual address)	An RVA in the portable executable (PE) header, which has a value of zero, indicates the field has not used all tables, and structure fields must be united on their ordinary limits, with the possible exception of the debug information.
3	Major image version	It is the file version. This record is user-definable and not connected to the task of the application. Many benign programs have more varieties and a larger image version set. Malware distributes a 0 value.
4	MajorOSVersion (major operating system version)	It is the major operating system required to run .exe files.
5	ExportRVA (export relative virtual address)	RVA (relative virtual address) exports ordinals for table entry. The location is virtual to the commencement of the image base. The export address table holds the location of exported data, entry points, and absolutes. An ordinal value is used to index the export address table.
6	Export size	Present the size of the export records. Only DLLs, not runtime applications, have export tables. So, the vote of this feature may be positive for clean files, which contain many DLLs and 0 for virus files.
7	IatRVA	This means the relative-virtual address of the import-address table. The value of this feature is read chunks of 4096 bytes and cleanest files and 0 or a very large value for virus files.
8	Major linker version	The major version linker produced the file to the PE header in the major linker version, and the resources size malware will be sometimes 0 in the section of PE header. Malware sometimes has 0 resources.
9	Minor linker version	The minor version linker produced the file.
10	Number of sections	The amount of virtual memory to standby for the initial thread's stack.
11	Size of stack reserve	The amount of virtual memory to reserve for the initial thread's stack.
12	All characteristics	It is a set of flags indicating under which circumstances a dynamic-link library (DLL) initialization function
13	Resource size	It symbolizes the dimensions of the resource section. Some malware records may have no resources. Benign files may have higher resources.
14	Machine	Defines the architecture type of the computer. The program can be run only on a system that monitors this type.

**Input:**

Training samples = series of API calls

C:

c:

large:

attribute\_list:

test\_attribute:

**Output:**

Vector feature

**Function**

1. Create a node N

2. **If**  $N = c$  **Then**

3. *return*(n)

4. **Else**

5.  $C = n$

6. **End if**

7. **If** attribute\_list = 0 **Then**

8. *return*(n)

9. **Else**

10.  $C = large$

11. **End if**

12. *test\_attribute* = large

13. **For**  $a_i$  **To** *test\_attribute*

14. *Sample\_set* = N is portitioned.

15. **If** *test\_attribute* =  $a_i$

16.  $n = test\ condition$

17. **End if**

18. **End For Loop**

**End Function**

ALGORITHM 1: Modified decision tree algorithm.

```

Input:
  Dataset = malicious and benign files
  C:
  c:
  N:
  attribute_list:
  test_attribute:
  k

Output:
  vector feature

Function selection of features (RW dataset)
1. For  $i = 0$  to  $N$  do
2.   Replace Sample_set =  $c$ 
3.   Create node  $N$ 
4.   Call tree( $N$ )
5. End for
6. Createtree( $n$ )
7. IF  $N = \text{attribute\_list}$  then
8.   Return( $N$ )
9. Else
10.  Select from vector feature
11.  Select vector feature  $F$ .
12.  For  $i = 0$  to  $k$  do
13.    Set sample  $N$  to  $C$ , where  $C$  is features = match vectors calltree ( $N$ )
14.  End for
15. End if
End Function

```

ALGORITHM 2: Random forest [30].

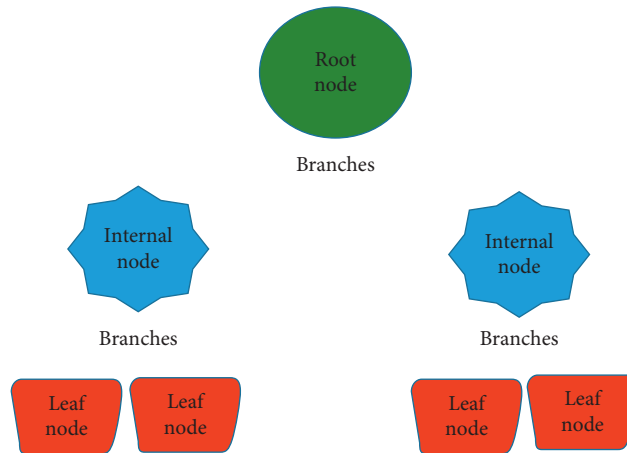


FIGURE 2: Strong classifier (random forest).

a strong classifier by the combination of weaker one [31]. A tree just has one node, and two leaves are called decision stump Figure 3.

$h(x)$  is a weak classifier. This is equivalent to saying that  $(h)$  is computed as a weighted majority vote of the weak hypothesis  $(h)$ , where each hypothesis is assigned weight  $F(x)$ . The weak classifier learns by considering one simple feature and  $h(x)$  is the most useful feature for the classification selection Figure 4.

## 4. Experiments and Results

In this phase, the experimental environment, experiments, and results are discussed. The datasets are statistically analyzed to understand the data. Then, different classification techniques were applied to classify the malicious and benign files, and finally, the performance evaluation measures were used to assess the performance of the classifiers.

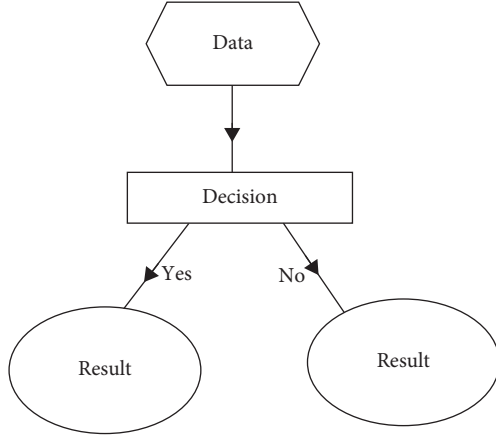


FIGURE 3: AdaBoost flow of data.

$$\left. \begin{array}{l} h_1(x) \in \{-1, +1\} \\ h_2(x) \in \{-1, +1\} \\ \vdots \\ h_T(x) \in \{-1, +1\} \end{array} \right\} F(x) = \left( \sum_{t=1}^T a_t h_t(x) \right)$$

FIGURE 4: Weak classifier (AdaBoost).

TABLE 4: Confusion matrix for evaluation measures.

Predication classes				
Predication classes	True	True	False	Total rows
	False	TP	FP	TP + FP
Total column		FN	TN	FN + TN
		TP + FN	FP + TN	N

**4.1. Datasets.** The dataset used in this study consists of 78550 samples, where 35369 samples are malware and 43191 samples are benign. RW is the type of malware, and benign is good ware. The dataset is nearly balanced; therefore, it does not need the balancing techniques.

**4.2. Experimental Environment.** All the experiments for this study are conducted on the core i5 machine with 2.4 GHz CPU and 8 GB of memory. The decision tree, random forest, and AdaBoost were implemented in Python due to its simplicity and scalability.

**4.2.1. Evaluation Matrices.** In this study, different evaluation measures are used to relate the performance of the classifiers. These include accuracy, sensitivity, specificity, and *f1*. All these measures are grounded on the confusion matrix given in Table 4.

Accurateness is the utmost intuitive performance measurement. It is a relation of correctly predicted observation concluded over total observation. The accuracy of the model is calculated using (2). Sensitivity statistic (recall) is a proportion of correctly predicted positive observation and overall positive observations in the actual class, and it is calculated using (3). The negative class

prediction power of the classifier is called specificity, which can be calculated using (4). Finally, the *f1* measure is calculated using (5) which is the Harmonic mean of the sensitivity and specificity:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

$$\text{sensitivity} = \frac{TP}{TP + FN}, \quad (3)$$

$$\text{specificity} = \frac{TN}{TN + FP}, \quad (4)$$

$$f1 = 2 \left( \frac{\text{sensitivity} * \text{specificity}}{\text{sensitivity} + \text{specificity}} \right). \quad (5)$$

**(1) Accuracy-Based Analysis.** Table 5 presents the accuracy-based result of the experiments. According to the reported results, the performance of the decision tree is promising with the highest accuracy of 99.34%. Random forest is very close to the performance of decision tree having 99.24% accuracy. It is clear that the AdaBoost has less performance as compared with that of the decision tree and random forest. The AdaBoost has the lowest accuracy of 98.37%.

**(2) Sensitivity-Based Analysis.** The sensitivity-based comparison of 10-fold cross-validation is performed the best as shown in Table 6. The experiments show that the sensitivity of the decision tree is higher.

**(3) Specificity-Based Analysis.** Table 7 represents the specificity-based comparison of the different classifiers. The experiments show that the specificity of decision tree has a higher accuracy and the value is 99.62% because the feature of specificity is higher.

Specificity (Precision) is a proportion of correctly classified positive observation over total predicted positive observation.

**(4) *f1* Measure Based Analysis.** Table 8 represents the *f*-measure based comparison of performing. The experiments show that the *f1* value of Decision Tree is higher accuracy value is 99.55%.

**4.3. Performance Comparison with State-of-the-Art Techniques.** By comparing the performance using different classifiers used on the dataset, it is clear that the proposed technique availed a higher accuracy as matched to the already developed models. The results in Table 9 show that the modified decision tree has the highest accuracy of 99.56%. AdaBoost has the lowest accuracy of 98.37%. random Forest has an average accuracy of 99.38%.

It also clearly shows that the proposed technique availed a higher accuracy as matched to the already developed models. Table 10 presents the results of the contrast of the suggested algorithm with other multiple methods.



TABLE 5: Accuracy-based comparison of classifiers.

S. no.	Techniques	Accuracy (%)
1	Decision tree	99.34
2	Random forest	99.24
3	AdaBoost	98.37

TABLE 6: Sensitivity-based comparison of classifiers.

S. no.	Techniques	Accuracy (%)
1	Decision tree	99.56
2	Random forest	99.50
3	AdaBoost	98.11

TABLE 7: Specificity-based comparison of classifiers.

S. no.	Techniques	Accuracy (%)
1	Decision tree	99.62
2	Random forest	99.51
3	AdaBoost	98.57

TABLE 8: F-Measure based comparison of classifiers.

S. no.	Techniques	Accuracy(%)
1	Decision tree	99.55
2	Random forest	99.50
3	AdaBoost	98.33

TABLE 9: Proposed detection scheme comparison.

Technique	$S_n$ (%)	$S_p$ (%)	f1 (%)	$A$ (%)
Decision tree	99.49	99.62	99.55	99.56
Random forest	99.50	99.51	99.50	99.38
AdaBoost	98.11	98.57	98.33	98.37

TABLE 10: Comparison of the performance results with existing research.

S. no	Ref.	Techniques	Accuracy (%)
1	[18]	Decision tree + RF + K – nearest neighbor + naive Bayes	97.3
2	[19]	Random forest (RE) + decision tree	93
3	[20]	Random forest (RE) + decision tree	98.5
4	[23]	Random forest (RF) + decision tree	97.95
5	Proposed	Modified decision tree	99.56

## 5. Conclusion and Future Direction

In this research, the RW detection at runtime scheme is developed which uses a preprocessed dataset that comprises benign and RW files. Benign is good ware, and RW is a special type of malware that keeps the data encrypted until a ransom is paid to the attacker. In the experiment, three different algorithms, namely, decision tree, random forest, and AdaBoost, are used to detect the RW and benign

files. The modified decision tree, among the three algorithms, performed well in terms of accuracy, sensitivity, specificity, and f1-measure. Our experimental outcomes demonstrate that the presented malware classification's testing and training accuracy is reached at 99.56%. Researchers stated some facts about sheltered device from attack and established some parameters to save data from the attack in the future, because RW is Trojan-type attack and malware, and so anomaly-based IDS may be used in the future for detecting abnormal behaviors of the network. Data mining techniques are used for detecting the activity of attack.

## Data Availability

The data used to support the findings of the study are available upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest associated with the publication of this article.

## Authors' Contributions

Faizan Ullah and Qaisar Javaid conceptualized the study; Dilawar Shah was involved in the formal analysis; Abdu Salam was responsible for the methodologies and resources and wrote the original draft; Masood Ahmad and Muhammad Abrar were responsible for the software; Qaisar Javaid supervised the study; Nadeem Sarwar wrote, reviewed, and edited the article.

## References

- [1] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions," *Computers & Security*, vol. 74, pp. 144–166, 2018.
- [2] J. Oberheide, E. Cooke, and F. Jahanian, "CloudAV: N-version antivirus in the network cloud," in *Proceedings of the USENIX Security Symposium*, pp. 91–106, San Jose, CA, USA, July 2008.
- [3] M. Baykara and B. Sekin, "A novel approach to ransomware: designing a safe zone system," in *Proceedings of the 2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1–5, Antalya, Turkey, March 2018.
- [4] S. Mehnaz, A. Mudgerikar, and E. Bertino, "Rwguard: a real-time detection system against cryptographic ransomware," *Research in Attacks, Intrusions, and Defenses*, in *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 114–136, Heraklion, Greece, September 2018.
- [5] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "{UNVEIL}: a large-scale, automated approach to detecting ransomware," in *Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 757–772, Austin, TX, USA, August 2016.
- [6] A. Ajjan, "Ransomware: next-generation fake antivirus," A SophosLabs Technical Paper, 2013.
- [7] K. Savage, P. Coogan, and H. Lau, "The evolution of ransomware," Symantec, Mountain View, CA, USA, 2015.

- [8] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the gordian knot: a look under the hood of ransomware attacks," in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 3–24, Milan, Italy, July 2015.
- [9] A. Kashyap, A. Horbury, and A. Catacutan, "Internet security threat report 2014," 2017, [http://www.symantec.com/content/en/us/enterprise/other\\_resources/bistr\\_main\\_report\\_v19\\_21291018.en-us.pdf](http://www.symantec.com/content/en/us/enterprise/other_resources/bistr_main_report_v19_21291018.en-us.pdf).
- [10] C. Steffen, B. R. ransomware Attacks Russian, "Featured in This Issue, "Should jump box servers be consigned to history?" *Network Security*, vol. 2017, no. 11, pp. 5–6, 2017.
- [11] V. T. Nguyen, A. S. Namin, and T. Dang, "MalViz: an interactive visualization tool for tracing malware," in *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 376–379, Amsterdam, Netherland, July 2018.
- [12] D. Gonzalez and T. Hayajneh, "Detection and prevention of crypto-ransomware," in *Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pp. 472–478, New York, NY, USA, October 2017.
- [13] A. Tandon and A. Nayyar, "A comprehensive survey on ransomware attack: a growing havoc cyberthreat," in *Data Management, Analytics and Innovation*, pp. 403–420, Springer, Singapore, 2019.
- [14] G. Norris, A. Brookes, and D. Dowell, "The psychology of internet fraud victimisation: a systematic review," *Journal of Police and Criminal Psychology*, vol. 34, no. 3, pp. 231–245, 2019.
- [15] A. Zimba and M. Chishimba, "On the economic impact of crypto-ransomware attacks: the state of the art on enterprise systems," *European Journal for Security Research*, vol. 4, no. 1, pp. 3–31, 2019.
- [16] O. M. Alhawi, J. Baldwin, and A. Dehghantanha, "Leveraging machine learning techniques for windows ransomware network traffic detection," in *Cyber Threat Intelligence, Advances in Information Security*, vol. 70, pp. 93–106, Springer, Cham, Switzerland, 2018.
- [17] M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," *Computers & Security*, vol. 77, pp. 578–594, 2018.
- [18] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211–221, 2019.
- [19] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using SDN," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pp. 1–6, Verona, Italy, November 2018.
- [20] D. Carlin, P. O'kane, S. Sezer, and J. Burgess, "Detecting cryptomining using dynamic analysis," in *Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pp. 1–6, Belfast, UK, August 2018.
- [21] D. Carlin, A. Cowan, P. O'Kane, and S. Sezer, "The effects of traditional anti-virus labels on malware detection using dynamic runtime opcodes," *IEEE Access*, vol. 5, pp. 17742–17752, 2017.
- [22] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: benefits, limitations and use for detection," 2016, <https://arxiv.org/abs/1609.03020>.
- [23] S. Poudyal, K. P. Subedi, and D. Dasgupta, "A framework for analyzing ransomware using machine learning," in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1692–1699, Bangalore, India, November 2018.
- [24] Y. Takeuchi, K. Sakai, and S. Fukumoto, "Detecting ransomware using support vector machines," in *Proceedings of the 47th International Conference on Parallel Processing Companion*, pp. 1–6, Eugene, OR, USA, August 2018.
- [25] B. B. Rad, M. K. H. Nejad, and M. Shahpasand, "Malware classification and detection using artificial neural network," *Journal of Engineering Science and Technology*, vol. 13, pp. 14–23, 2018.
- [26] A. Tseng, Y. Chen, Y. Kao, and T. Lin, "Deep learning for ransomware detection," *IEICE Technical Report*, vol. 116, pp. 87–92, 2016.
- [27] V. Total, "VirusTotal-free online virus, malware and url scanner," 2012, <https://www.virustotal.com/en>.
- [28] D. Granström and J. Abrahamsson, "Loan default prediction using supervised machine learning algorithms," Master Thesis, Kth-Royal Institute of Technology, Stockholm, Sweden, 2019, <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1319711&dsid=-2508>.
- [29] L. Junjie, Z. Weichi, Y. Rong, and Z. Haochuan, "A method for option recognition based on the decision tree," *Wireless Internet Technology*, vol. 15, pp. 113–115, 2018.
- [30] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems*, pp. 2951–2959, Lake Tahoe, Nevada, December 2012.
- [31] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the Thirteenth International Conference Machine Learning*, pp. 148–156, Bari, Italy, July 1996.