Hindawi
Security and Communication Networks
Volume 2022, Article ID 1424638, 16 pages
https://doi.org/10.1155/2022/1424638

Research Article

Digital Forensics as Advanced Ransomware Pre-Attack Detection Algorithm for Endpoint Data Protection

Jian Du,[1] Sajid Hussain Raza,[2] Mudassar Ahmad,[2] Iqbal Alam,[3] Saadat Hanif Dar,[4] and Muhammad Asif Habib[2]

[1]*Transport Information Security Center Co. Ltd, Transport Telecommunications & Information Center, Beijing, China*
[2]*Department of Computer Science, National Textile University, (NTU), Faisalabad, Pakistan*
[3]*Academic Department, Nan Yang Academy of Sciences (NASS), Beijing, China*
[4]*Department of Electrical Engineering, University of Azad Jammu & Kashmir, Muzaffarabad 13100, Pakistan*

Correspondence should be addressed to Muhammad Asif Habib; drasif@ntu.edu.pk

Received 19 November 2021; Accepted 22 May 2022; Published 6 July 2022

Academic Editor: Farhan Ullah

Ransomware is a malicious software that takes files hostage and demands ransomware to release them. It targets individuals, corporations, organizations, and public services such as hospitals and police stations. It is a growing industry that affected more than three million users from 2019 to 2020. The ransom payments totaled 25 billion-plus dollars in the year 2019. The latest version of ransomware was developed using undetectable and nonanalysis techniques. This paper represents an intelligent KNN and density-based machine learning algorithm to detect ransomware pre-attacks on an endpoint system. The data preprocessing and feature engineering techniques are augmented with the KNN algorithm for finding the solution. This helps the anti-malware developer, vendors, endpoint security provider companies, or researchers work on malware detection using advanced machine learning algorithms to develop the more effective ransomware defensive solutions to detect and prevent ransomware pre-attack execution. The proposed KNN and density-based algorithm will predict ransomware detection with higher accuracy than other machine learning algorithms. The anti-malware and anti-ransomware solution provider companies can use this algorithm to improve their existing ransomware detection solutions for endpoint users.

1. Introduction

Malware is a computer program used to access a computer system without the user's permission, usually for the benefit of a third-party known as a hacker. The malware contains malicious and harmful pieces of code, like ransomware, spyware, and other malware applications. In this paper, the only focus is on ransomware-based malware. Ransomware is a unique branch of malicious software that takes files hostage and demands ransomware to release them. It targets individuals, corporations, organizations, and public services such as hospitals and police stations. It is a growing industry that affects more than 3 million users from 2018 to 2019 [1]. The ransom payments totaled 25 billion-plus dollars [2]. The more advanced versions of ransomware contain anti-analysis techniques that are undetectable. The best guarantee against a ransomware attack is a good data backup. Ensure that any device comprising backups is safe and well protected. It needs to test the archived data from time to time. Ensure that any device containing backups is safe and well protected. Test the archived data from time to time.

A well-designed antivirus gets rid of ransomware in seconds, but ransomware designers are brilliant. They always work very hard to circumvent old-designed signature-based ransomware detection. Finally, it takes a receipt from an antivirus for a newly generated, zero-day ransomware attack to fetch your files. Even though the antivirus received a message that removed the ransomware, the files could not be recovered permanently. New generation antivirus programs complement signature-based detection and modify programs' monitoring behaviors. A few rely heavily on malicious behavior other than looking for threats, and behavior-

based malware detection, specifically looking towards ransomware behavior, is becoming more specific.

A schematic diagram of the working ransomware process is shown in Figure 1.

More than 300 ransomware types will exist until the end of 2022. However, this research discusses only a few types of ransomware that are the most harmful and dangerous for endpoint users. When examining the currently monitored ransomware types, our focus was only on the top five crypto-ransomware types/families: Cerber, CryptoWall, Locky, TeslaCrypt, and TorrentLocker.

Ransomware usually hunts files stored in public places such as desktops and documents folders in windows. A few antivirus toolkits and security packages prevent ransomware attacks by restricting unauthorized user access to these locations. Usually, some good programs, like Microsoft Word processing and spreadsheets, are already authorized. Every time they try to access it using a unique software program, you, as the user, can ask if you want to allow file access to the program. If this alert was unexpected and not generated by you, just report and block it. Using an online backup toolkit to keep your backup files up to date is undoubtedly the best possible protection against a ransomware attack. You get rid of the malware first, perhaps with the help of technical support from antivirus companies.

Once this task is complete, just restore all of your backup files in the previous state. Be aware that a ransomware attacker may try to encrypt your archive's backup files. A backup system that displays your backup files on a virtual drive can be highly vulnerable to a ransomware attack. Contact your backup provider to determine what protection the product offers against ransomware. Top five ransomware families trend in 2019 is shown in Figure 2.

1.1. Research Motivation. Machine Learning (ML) is a branch of Artificial Intelligence (AI) that studies systems' automatic learning without feeding any hard-coded program. ML is also a data analysis technique that computers use to analyze what humans and animals take from nature, like learning from their experiences. We proposed a unique and intelligent KNN and density-based adaptive clustering algorithm to detect ransomware pre-attacks on an endpoint system. Hence, Machine learning has become the most demanding topic. ML algorithms use computational methods to train information directly from given data without depending on predefined calculations as models.

With colossal data evolution every day, ML has become an essential technique for solving many information security problems. The general machine learning diagram process is shown in Figure 3. We used both supervised and unsupervised machine learning techniques to improve the detection of ransomware attacks for an endpoint.

1.1.1. Supervised Machine Learning. Supervised machine learning is used to build a data model for evidence-based prediction in uncertain situations. Supervised machine learning uses regression and classification techniques to develop a prediction model. Clustering is very helpful due to its maximum capacity to discover the previously unknown groups in the dataset. Five basic categories are the most common in cluster methods. One of them is the hierarchical method, which generates a hierarchical tree to divide into specific predefined datasets. Supervised machine learning uses regression and classification techniques to develop a prediction model.

Regression: In supervised machine learning, regression is used to predict continuous responses like changes in encryption methods such as hash or changes in the level of encryption keys. There is a relationship between independent and dependent sets of variables in regression analysis. There are some popular machine learning–based regression algorithms like linear regression, generalized linear model (GLM), support vector regression (SVR), Gaussian process regression (GPR), ensemble methods, decision trees, and neural networks.

Classification: In supervised machine learning, classification is used to predict the discrete response. If there is a set of mixed ransomware attack data, we can find the categories of good ware and ransomware, respectively, by applying classification techniques. These classified data can be categorized, tagged, and classified into different classes or groups. There are some popular machine learning-based classification algorithms like support vector machine, discriminant analysis, naive Bayes, and nearest neighbor.

1.1.2. Unsupervised Machine Learning. It deals with studying inherent structures and fetching hidden patterns of data. Unsupervised machine learning contains unlabeled data. Unsupervised machine learning uses clustering techniques for the development of data models.

Clustering: In unsupervised machine learning, clustering is a technique used for an exploratory data analysis to fetch or recognize hidden groupings or patterns in data. Market research, object recognition, and sequence analysis are the most common clustering applications. Some popular machine learning–based regression algorithms are K-Means, K-medoids, fuzzy, c-means, Hierarchical, Gaussian mixture, and hidden Markov model.

1.1.3. Research Challenges. Many researchers, professionals, and security experts have developed good ransomware analysis and detection systems. Still, they only focused on either network-based preventive systems or OS-based prevention systems. Some only focus on static analysis or dynamic solutions to analyze ransomware-affected strategies.

1.1.4. Main Contribution. This paper proposes a unique and intelligent KNN and density-based adaptive clustering algorithm to detect ransomware pre-attacks on an endpoint system. The implemented algorithm is better than the previously developed ransomware analysis and detection algorithms. The first step is to collect ransomware data of the most relevant multi-class ransomware families that only affect Windows-based operating systems in the form of exe files. The data is collected or provided by a data engineer,

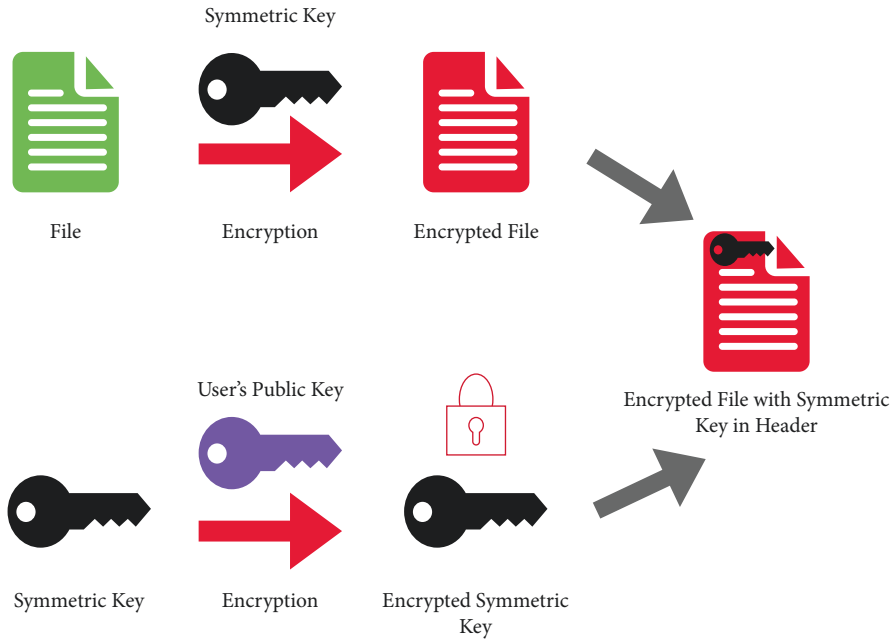


Figure 1: Ransomware generic working process.

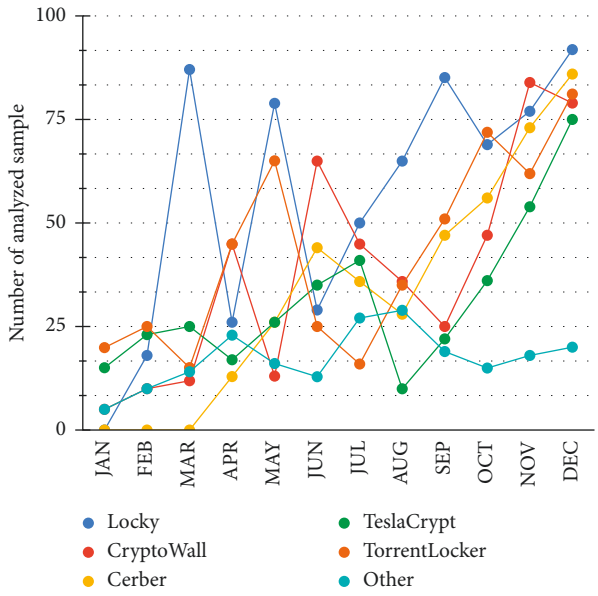


Figure 2: Top five ransomware families' trend samples in 2019.

respectively. We use a dataset of ransomware from the Github repository [3]. Then, we check for the outliers by initializing threshold values like measuring the distance of the closest data point, which is greater than its nearest cluster identifier and is identified as an outlier in our dataset. The next step is to remove outliers and clean up the ransomware dataset.

Data cleaning may generate some missing values, and we must fix them. The second step, wrangle a ransomware dataset to create new variables, identify duplicates, and filter variables. The next step is ransomware data's feature engineering. It features extraction using recursive feature elimination (RFE) and principal component analysis (PCA), which use orthogonal transformation to convert data into lower dimensional space like in between a specific number ranges to maximize the variance of the dataset. Then shuffle the extracted feature dataset to apply machine learning–based KNN and density algorithms and get trained in the dataset with the ratio of 70:30 as training and testing data. The proposed KNN and density-based algorithm predicted ransomware detection with higher accuracy than other machine learning algorithms. Finally, the anti-malware and anti-ransomware solution provider companies can use this algorithm to improve their existing ransomware detection solutions for endpoint users.

1.1.5. Use of Clustering. In clustering, the number of data objects is divided into subsets. Every individual subset is known as a cluster, so objects in a cluster are similar to each other but not different from other clusters. Separation is done using a clustering algorithm [4]. Clustering is very helpful due to its maximum capacity to discover the previously unknown groups in the dataset. Five basic categories are most common in cluster methods. One of them is the hierarchical method, which generates a hierarchical tree to divide into specific predefined datasets. These methods can also be categorized into two more types of operating modes. 1- Decomposition (top-down), 2- Concatenation (bottom-up) and contain BIRCH (CURE)-ROCK & CHEMALOEN. Second-one is the division method. The first k-partition is generated, and then the objects are shifted from one substance to other substances (partitions) to increase the substance quality with cycle positioning techniques that contain K means- CLARINS and fuzzy c-means algorithms.

These methods can also be categorized into two more types of operating modes; decomposition (top-down) and Concatenation (bottom-up), which contain BIRCH

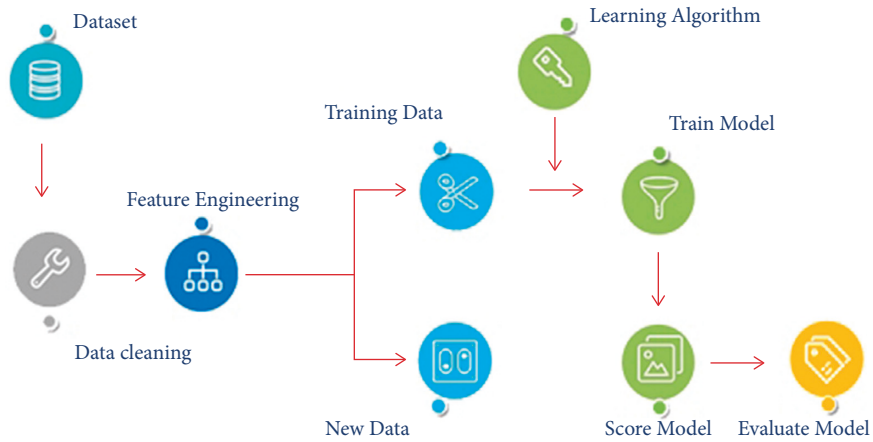


Figure 3: General machine learning process diagram.

(CURE)-ROCK & CHEMALOEN. Second mode is the division method. The first K- partition is generated. Then, the objects are shifted from one substance to another to increase the substance quality with cycle positioning techniques containing K means- CLARINS and Fuzzy C-means algorithms.

1.2. Density-Based Method. Density-Based Method (DBM) groups objects under Object Density (OD). e.g., DBSCAN describes the clusters as areas of point-to-point density. The clustering algorithm also enters data by constantly developing areas of high density. It can detect groups of random numbers in the space as clusters in a database by using noise. The Density-Based Clustering (DBC) algorithm can process clusters of any kind because the time complexity is less than $O(n^2)$ and is acceptable for processing large amounts of data. For e.g., the DBSCAN algorithm. Figure 4 shows the types of points in DBSCAN.

The density-based clustering (DBC) algorithm can process clusters of any kind because the time complexity is more minor than O(N2) and is acceptable for processing large amounts of data. For e.g., the DBSCAN algorithm. This differentiates the main points from less essential points. Still, there is a drawback to specifying two parameters; the radius of the surrounding area (RSA) and the lowest amount of points in a given surrounding area. Moreover, DBCLASD uses the sample distance to its nearest neighbor as a random variable, and the sample density is calculated from the probability distribution of this distance.

The best advantage is that the parameters do not have to be specified, and the disadvantage is that the uptime is about twice as long as the DBSCAN algorithm. The fundamental principle attracts much attention concerning the algorithm based on more than three similar algorithms. The only methods assume that neighbors surround the cluster center. Density-based clustering algorithms (DBCA) are not only widely used at any point with a higher density [5] but are also being further developed [6]. In particular, KNN and density-based methods are integrated to increase Overall Cluster Efficiency (OCE).

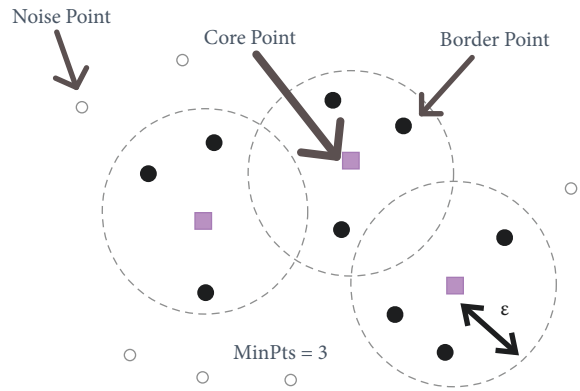


Figure 4: Types of points in DBSCAN.

Analysis of clusters is highly used in many science fields, such as social science, bio, stat, pattern recognition, info fetching, machine-learning (ML), and Artificial Intelligence (AI). For e.g., analysis of clusters is used to categorize related search documents, find genes and proteins with the same functions, and classify earthquake-prone locations.

Moreover, the analysis of clusters is used as a preprocessing step in multiple applications such as data compression and efficient search of the closest points. The already defined clustering algorithms face several challenges today due to their growing size and uneven data distribution. The clustered data are probably a challenging task complicated by the random shapes of the clusters, their varying sizes, different densities, and ambiguous separation.

Many researchers have studied clustering by density under a promising approach to addressing these challenges. They can find randomly formed clusters and handle noise without first knowing the number of clusters. The cluster is generally considered in high-density areas separate from low-density regions. This hypothesis allows groups to be characterized by nonconvex shapes and zones. The effectiveness of existing density-based clustering algorithms relies heavily on intensive user arbitration to determine density thresholds, which determines the differentiation of all density levels available in the data. This density limit is

often harder to calculate for large amounts of anonymous data. Additionally, using a global density threshold can lead to nonclassified clusters with lower densities.

Decision-tree works on the entire dataset by using all of its variables and features, but the random forest only selects elements and variables randomly. The random forest has to build multi-level decision trees from the dataset and then show the results in averages.

2. Literature Review

A literature review of published work and development on Machine Learning Algorithm based ransomware detection is discussed in this section. A few years ago, when the Harvard-trained evolutionary biologist Joseph Pop developed the first ransomware virus of its kind in 1989, it was called trojan AIDS, also known as PC Cyborg. Pop sent 20,000 introductory discs with information on AIDS infection to the World Health Organization (WHO)'s International AIDS Conference attendees. AIDS trojan horse is the first generation of ransomware malware and is relatively easy to remove. The trojan uses simple symmetric cryptography, and a tool to decrypt file names is available soon, but the trojan horse AIDS created the conditions for what lies ahead. Top 10 malware categories are shown in Figure 5.

In the last few years, ransomware turned pro. It is usually installed on a user's workstation (PC or Mac) using psychological manipulation attacks that trick users into clicking a link or opening an attachment. Once on a computer, the malware begins encrypting any data files it can find in the system itself and any networks that come under computer access. Then, suppose the user requests access to one of these files. In that case, it is blocked, and the system administrator, who is receiving a warning from the user, finds two files in the directory indicating that the file was used as a ransom and how to pay the ransom decrypted files. The techniques used by cybercriminals are constantly evolving to bypass traditional protections.

Some significant ransomware are WAnnaCry, Petya/NotPetya, Bad Rabbit, and Locky. Ransomware is a highly successful criminal business model. According to a report from Cybersecurity Ventures, the annual cost of ransomware is projected to exceed $11.5 billion by 2019. Finding and replying to copies of phishing emails to 3.8 billion people worldwide, papered to reach 6 billion by 2022, online is the next best thing to vaccinating them with ransomware. Cybersecurity Ventures predicts that cybercrime will cost the world more than $6 trillion per year by 2021, up from $3 trillion in 2019.

Ransomware is expected to worsen and cause a more significant share of total cybercrime by 2021. Employees are an essential variable and a notable potential winner in reducing the cost of ransomware damage. The report from Cybersecurity Ventures, due for release in 2018, includes the estimated cost of ransomware for the five years from 2017 to 2021 [7, 8]. Table. 1 describes the top five ransomware families trend samples in 2019 concerning crime type in a loss.

The ML techniques can work more effectively and efficiently to detect malware [? 8]. They used dynamic analytics; the reports collected by some online analytics services could not evaluate various ML classifiers. The best performing J48 classifier from the decision tree achieved 97.3% accuracy and a percentage of False Positives (FPR) of 2.4%. They proposed a framework for automated malware detection behavior using ML [9, 10]. They embed observed behavior in a vector space and applied clustering algorithms to improve previous work in this area significantly.

ML was highly needed to overcome existing limitation techniques for more effective detection of malware [11]. Authors in Ref. [4] introduced a statically and dynamically integrated classification method to overcome the limitations associated with each technique [4]. They demonstrate the importance of combining old and new malware patterns to overcome malware authors' avoidance techniques. Their accuracy was at least 5% lower when only new samples were used in all classifiers assessed. Similarly, one more detailed review of the methods and technical tools used to analyze and classify malware has been conducted [5]. They include processes for acquisition, static or dynamic analysis or feature extraction, and ML classification. The flow control method achieved an accuracy increase of 2% compared to the text-based method using ANN classification [12].

A similar technique was being used with additional filtering features to achieve 97% accuracy with random forest classifiers [13]. The ransomware classification system has approximate values of the control flow graph adjustment. The distance metrics are based on the distance between the string-based signature feature vectors. It is being done to achieve the best accuracy rates [14]. Additional studies on size reduction/screening were included in which a two-step approach was proposed to the dimensional reduction that significantly combines feature selection and extraction to reduce training feature size and classification [15].

They used the reduction function to identify the top 10 malware detection codes and reduced the controlled learning algorithm's training time by 91% without losing accuracy [16]. The feature reduction identifies nine features that differentiate malware from good software. With the random forest classifier, an accuracy of 99.60% was achieved [17]. A ransomware detection framework is based on 20 extracted file systems and registry events [18].

With the Bayesian network model it achieves an F-measure of 93.3%. Most recently, in 2017, the use of a sequential sample file system, registry, and DLL event is being fetched to achieve 97% accuracy when differentiating between crypto-ransomware and valuable software and 95.5% when distinguishing between three different ransomware groups [6, 12]. They introduce a new method (Adaptive-DP) that estimates density using the heat diffusion method and the adaptive approach to select the exact number of cluster centers.

The limitations of DP, the difficulty of choosing an appropriate density estimation method, the selection of boundary distances, and the human interpretation required to select the number of cluster centers have been improved

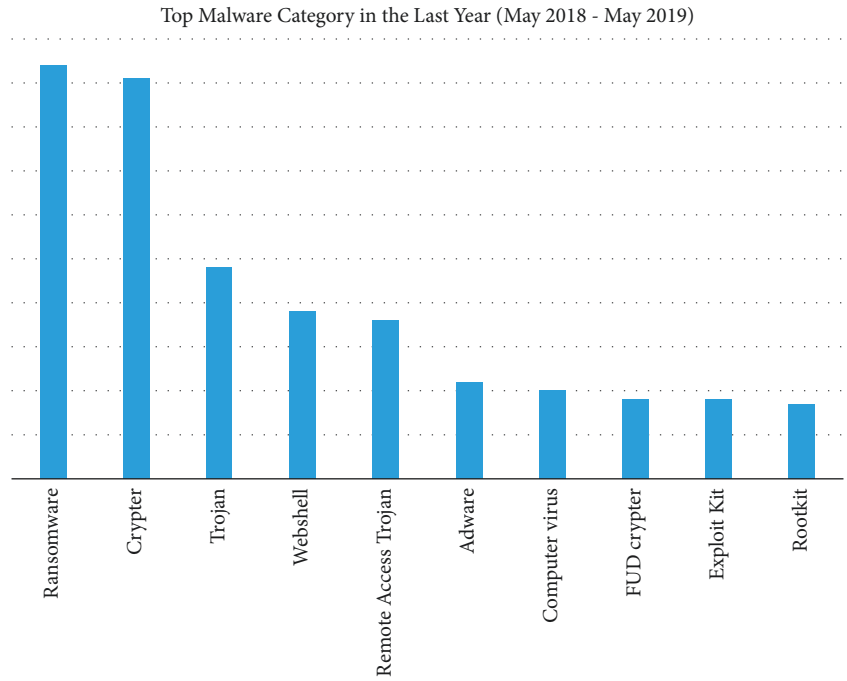


Figure 5: Top 10 malware category mentions overall [2].

Table 1: Top five ransomware families' trend samples in 2019.

Crime Type	Loss
BEC/EAC	$676,151, 185
Confidence fraud/Romance	$211,382,989
Nonpayment/Nondelivery	$141,110,441
Investment	$96,844,144
Personal data breach	$77,134,865
Identity theft	$66,815, 298
Corporate data breach	$60,942,306
Advanced fee	$57,861,324
Credit card fraud	$57,207,248
Real estate/Rental	$56,231,333
Overpayment	$53,450,830
Employment	$38,883,616
Phishing/Vishing/Smishing/Pharming	$29,703,421
Other	$23,853, 704
Lottery/Sweepstakes	$16,835,001
Extortion	$15,302,792
Tech support	$14,810,080
Misrepresentation	$14,580,907
Harassment/Threats of violence	$12,569,185
Government impersonation	$12,467,380
Civil matter	$5,766,550
IPR/Copyright and counterfeit	$5,536,912
Malware/Scareware/Virus	$5,003,434
Ransomware	$2,344,365
Denial of service/TDoS	$1,466,195
Charity	$1,405,460
Health care related	$925,849
Re-shipping	$809,746
Gambling	$598,853
Crimes against children	$46,411
Hacktivist	$20,147
Terrorism	$18,926
No lead value	$0

in Adaptive-DP [19, 20]. They have developed a RE framework for detecting ransomware infected data recovery by using machine learning and features generation engines [21].

This RE framework can analyze malware files at a multi-level by using binary codes and different libraries. The author used a portable executable parser with object code dump Linux-based tools to decode binary-level assembly instructions and DLL, respectively [19]. Their experiments show the performance of detecting ransomware samples accurately from 76% to 97% using eight machine learning techniques where seven results out of eight showed 90% accuracy for detection rate [22]. This study is purely based on the static level of analysis to differentiate between ASM and DLL levels as compared to regular binaries [3].

The research gap was founded after careful study of the literature. We evaluate shallow and deep networks for ransomware detection and classification. To characterize and differentiate the ransomware from different types of other ransomware families, some researchers use API. Some previous researchers used nonmachine learning techniques to achieve more accurate results in detecting ransomware. Unfortunately, over time, due to the more advanced use of machine learning techniques used by the hackers to develop the ransomware, the endpoint use required more accurate solutions against ransomware.

Based on the literature review, we analyze that many researchers, professionals, and security experts have developed good ransomware analysis and prevention systems. Still, they only focused on either network-based prevention systems or only on OS-based prevention systems [23]. Some only focus on static analysis or dynamic solutions to analyze ransomware-affected strategies. That is why we proposed a

unique and intelligent KNN and density-based Adaptive clustering algorithm to detect ransomware pre-attacks on an endpoint system. So this suggested algorithm shows better results than previously developed ransomware analysis and detection algorithms. There are various applications of machine learning and deep learning in different applied domains of artificial intelligence [24, 25].

2.1. Problem Statement. There is a need for possible detection solutions against zero-day attacks and random file system detection. There is also a need to use a more advanced and more accurate machine learning detection technique like KNN and a density-based adaptive clustering algorithm for improving ransomware pre-attacks detection. Proposed solution: Our proposed KNN and density-based machine learning algorithm efficiently detects and analyzes ransomware pre-attack on an endpoint's system. This proposed solution also offers the detection of zero-day attacks as well.

2.1.1. Objective. To detect zero-day attacks and random file system attacks by using portable executables (.exe). To provide an intelligent KNN and density-based adaptive clustering algorithm for detecting ransomware pre-attacks on an endpoint's system.

2.1.2. Aim of Our Research. This project aims to provide a KNN and density-based adaptive clustering algorithm to detect ransomware pre-attacks on an endpoint's system. This proposed solution also offers the detection of zero-day attacks as well.

2.1.3. Research Questions. Q.1. How to detect and analyze ransomware using KNN and density-based algorithms? Q.2. How to perform detection of zero-day and random file name attacks using KNN and density-based algorithms?

2.1.4. Scope of Research. The scope of the suggested algorithms is very high for the detection and analysis of ransomware. To get live ransomware detection and zero-day attack updates, one can embed or deploy these algorithms in the Microsoft BI dashboard. It also has a limitation in scope because these algorithms are only trained for analyzing and detecting portable executable files (.exe format).

Table 2 shows a comparison of methods for the analysis and protection of ransomware attacks used by many researchers in literature.

There is a need for possible detection solutions against zero-day attacks and random file system detection. There is also a need to use a more advanced and more accurate machine learning detection technique like KNN and a density-based Adaptive clustering algorithm for improving ransomware pre-attacks detection to detect zero-day attacks and random file system attacks by using portable ex-executable in.exe format and to offer an intelligent KNN and density-based Adaptive clustering algorithm for detecting ransomware pre-attacks on an endpoint's system. To get live ransomware detection and zero-day attack updates, one can embed or deploy these algorithms in the Microsoft BI dashboard. It also has a limitation in scope because these algorithms are only trained for analyzing and detecting portable executable files in.exe format.

3. Research Methodology

Many windows based ransomware detection techniques and algorithms are available. These detection algorithms are working very effectively. But with the continued growth of zero-day attacks and the new generation of ransomware families' attacks on endpoint systems, there is a need to improve the malware detection techniques and algorithms. In this research, for the detection of ransomware in supervised ML, we used KNN and density-based algorithm and random forest with outstanding accuracy results. On the other hand, to detect zero-day attacks that lie under unsupervised ML, we implemented k-means and DBSCAN clustering algorithms to detect zero-day attacks. The methodology used in this research is shown in Figure 6.

For this purpose, we need to fetch the number of clusters by using the elbow method to identify the best optimal value of *k*. DBSCAN randomly placed k-centroids, and there is no need to specify the number of clusters. We used the Python programming language under the Jupyter platform in this research. These steps and their workings are briefly defined below step by step, respectively, in Figure 6. The first step is to collect ransomware data of the most relevant multi-class ransomware families that only affect Windows-based operating systems in the form of.exe files. The data collection is done by third-party sources, as mentioned in the below sections, provided by the data engineer in CSV file format. We converted the CSV file into.xlsx format for ease of implementation in Python. Then, we check for the outliers by initializing threshold values like measuring the distance of the closest data point, which is greater than its nearest cluster identifier and is identified as an outlier in our dataset. The next step is to remove outliers and clean up the ransomware dataset.

Data cleaning may generate some missing values, and we have to fix them. Afterward, wrangling a ransomware dataset to create new variables, duplicate values are identified, and the variables are filtered. The next step is ransomware data's feature engineering. It features extraction using RFE and PCA, which use orthogonal transformation for the conversion of data into lower dimensional space like in between specific number range to maximize the variance of the dataset. Then shuffle the extracted feature dataset to apply machine learning-based KNN and density algorithm and get trained in the dataset with 70:30 as training and testing data. For zero-day attack detection under unsupervised ML, k-means will be used.

Data collection is the most critical step in solving controlled machine learning problems. Machine learning models often give outstanding results when asked to recall objects from their training, but sometimes they show inferior results when taken out of their comfort zone.

Table 2: Comparison of State-of-the-art Methods for analysis and protection of Ransomware Attack.

Technique/Methodology	Today Attacks Protection	Min. File Loss	Min. False Positive Rate	Inputs are Enough	Min. Latency and Max. Performance	Invul.	Offer Counter Measure
Cloud-based sandbox environmental method	Unknown	Unknown	Unknown	No	No	Yes	Yes
Cloud-based detection method	Unknown	Unknown	Unknown	No	No	Yes	No
Monitoring abnormal registry and file system activities	Unknown	No	Unknown	Yes	No	No	Yes
Monitoring process and IO events	Yes	No	No	No	Yes	No	No
Machine learned behavior-based method	Yes	Yes	No	Yes	No	No	No
Software defined networking-based method	Unknown	Yes	Yes	Yes	Unknown	No	Yes
Connection monitor and connection breaker method	Yes	Yes	Yes	Yes	No	Unknown	No
A large scale automated approach	Yes	Yes	Yes	Yes	Unknown	No	No
Automated dynamic analysis method	Yes	Yes	Yes	Yes	Unknown	Yes	No

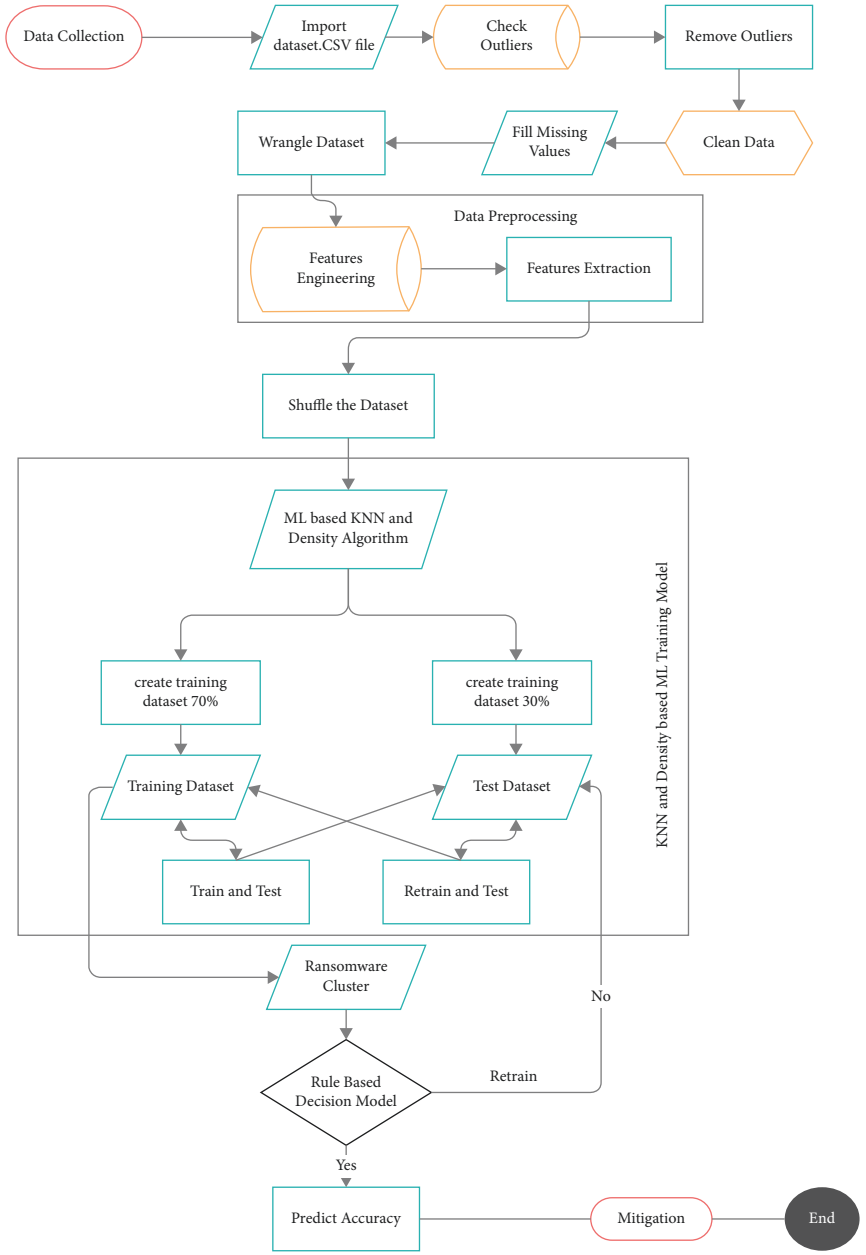


Figure 6: Advanced ransomware pre-attack detection algorithm for endpoint data protection.

ML algorithms require significant amounts of data to function. When they deal with millions and billions of pictures or text records, it is much more challenging to identify what causes an algorithm to perform poorly. Therefore, capturing a massive amount of information is not enough; by treating it with an ML model, the competitive results can be expected

Classifying ransomware into particular families is a challenging task due to the massive number of ransomware and multiple ransomware families. The ransom families' data used in this research experiment were identified on the ransomware Tracker website [21], the online ransomware data providing resource mainly used by the Internet-Service-Providers (ISPs), Computer Emergency Response Teams (CERT), and Law Enforcement-Agencies (LEA). When examining the currently being monitored ransomware families, our focus was only on the five-crypto families like Cerber, CryptoWall, Locky, TeslaCrypt, and TorrentLocker, including the available historical examples to confirm the variant history and malware changes in displayed code [26].

The ransomware tracking website (RTW) lists popular hosts for all ransomware families' distributions, payments, administrations, and management. The md5 value for all collected samples is uploaded to the VirusTotal Intelligence platform [36] to retrieve malware details rapidly. The download list is checked to ensure that only the Portable Executable PE (.exe) file's format is downloaded, with some other forms like DLL (Dynamic Link Library) being removed. This allows accurate comparisons with good software samples with a similar PE (.exe) format.

The dataset used to demonstrate the given algorithm is a making-dataset of malware images: visualizations and automatic classification documents [19]. This dataset contains 25 clusters of ransomware with several different family variants and 56 columns with a 65535 number of rows [6]. The deep convolutional neural network is used to detect malicious infections in IoT network through color image visualization [27].

3.1. Dataset Arrangement. The way datasets are organized plays a crucial role in controlled (supervised) classification. As we can see in Figure 7, there are 25 ransomware (malware) families, and each family has a different number of samples. Our classification method does not require malware debugging or code execution to related works. Besides, the texture of the image used for classification offers a lot more sustainable property in ambiguous technologies, especially for encryption. Finally, we apply our algorithm to a larger dataset that consists of 25 families in the corpus of ransomware malware than 9458 ransomware. After using a newly suggested algorithm, the final results show that our method offers better accuracy at lower computational costs. They encrypt various files on the victim's hard drives before requesting a ransom to get the files decrypted. Security-related media and some antivirus vendors quickly brandished this "new" type of virii.

3.1.1. Import Datasetcpsec. After Collecting and summarizing ransomware data, we convert them into two dataset

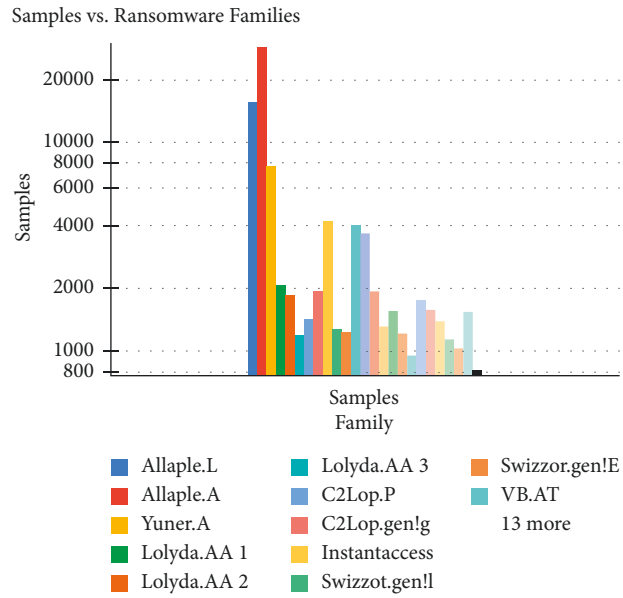


Figure 7: Ransomware (malware) dataset of 25 families [20].

formats of comma-separated file (CSV) and Microsoft Excel file format.xls or.xlsx. We require the dataset in three stages for the training of the machine learning model. One of them is training, the second is validation, and the third one is testing.

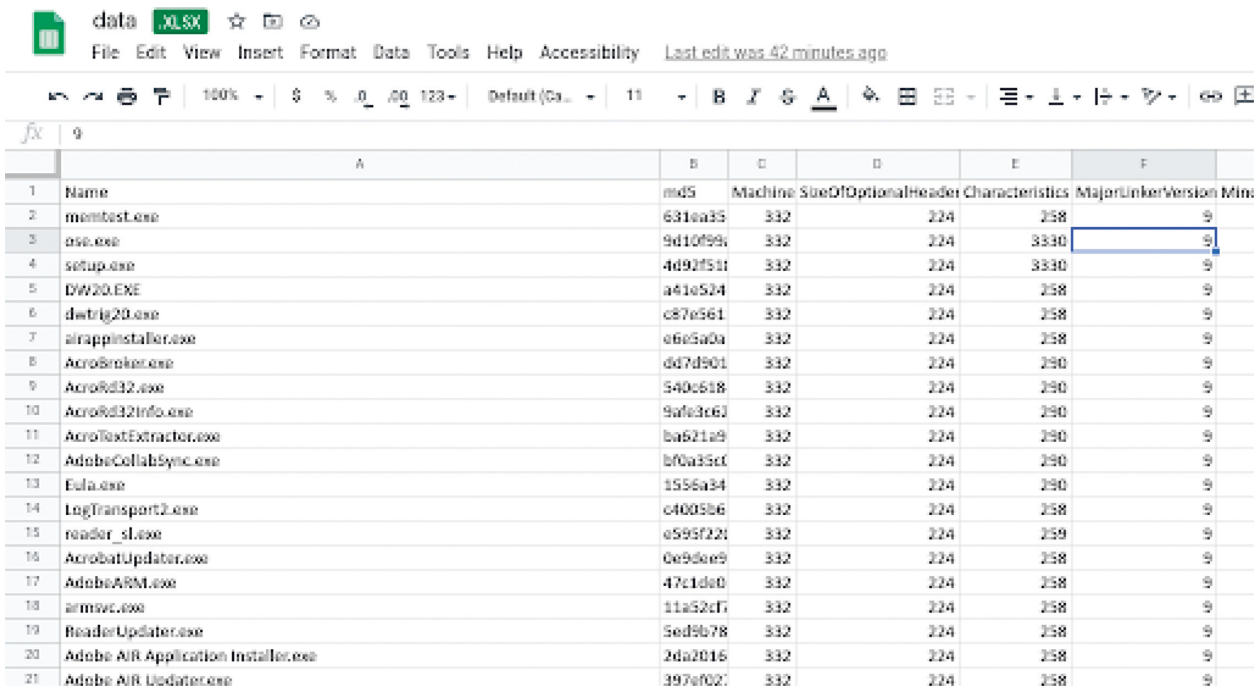
3.1.2. Excel File Data Reading. We need to import data from the excel file to pandas first. For this purpose, first, import the pandas' library using the commands. Then, we use the read_excel panda's method to read data from an excel file as shown in Figure 8.

The simplest way to call this method is to enter a filename. If the sheet name is not specified, the first sheet in the index is read.

Now, the read_excel method reads data from the excel file to the pandas DataFrame object. By default, pandas store data in DataFrames (df). Then, we store the DataFrame in a variable named df. Pandas have a built-in DataFrame.head () method, which we can use to return a few rows of our DataFrame quickly. If no arguments are given, the first five lines are displayed by default. We can verify that this aggregation exists by checking the number of rows in the aggregated DataFrame by calling the "shape" method, which displays the number of rows & columns.

The data.xls dataset file contains 65535 records and 57 columns. It can be helpful when counting the number of columns and rows compared with source records of datasets. Another tail method is also beneficial for seeing the last rows of a given dataset.

3.1.3. Training a Model. Data collection and modeling training is an iterative process, so we may need to retake the decisions taken while collecting data. The method includes data preprocessing, model training, and parameter setting.



	A	B	C	D	E	F	
1	Name	md5	Machine	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	Min
2	memtest.exe	631ea35	332	224	258	9	
3	ose.exe	9d10f99	332	224	3330	9	
4	setup.exe	4d92f51	332	224	3330	9	
5	DW20.EXE	a41e524	332	224	258	9	
6	dwtrig20.exe	c87e561	332	224	258	9	
7	airappinstaller.exe	e6e5a0a	332	224	258	9	
8	AcroBroker.exe	dd7d901	332	224	290	9	
9	AcroRd32.exe	540c618	332	224	290	9	
10	AcroRd32Info.exe	9afe3c6	332	224	290	9	
11	AcroTextExtractor.exe	ba621a9	332	224	290	9	
12	AdobeCollabSync.exe	bf0a33c	332	224	290	9	
13	Eula.exe	1556a34	332	224	290	9	
14	LogTransport2.exe	c4005b6	332	224	258	9	
15	reader_sl.exe	e595f22	332	224	259	9	
16	AcrobatUpdater.exe	0e9dee9	332	224	258	9	
17	AdobeARM.exe	47c1de0	332	224	258	9	
18	armsvc.exe	11a52cf	332	224	258	9	
19	ReaderUpdater.exe	5ed5b78	332	224	258	9	
20	Adobe AIR Application Installer.exe	2da2016	332	224	258	9	
21	Adobe AIR Updater.exe	397ef02	332	224	258	9	

Figure 8: Ransomware dataset in Microsoft excel sheet.

3.1.4. Data Preprocessing. In any machine learning process, the preprocessing of data is the step where the information is changed or encoded to bring it into a state such that machines can quickly analyze it. On the other hand, an algorithm can now easily interpret the data features. Cleaning data: is the most crucial step in any ML project. There are multiple and different statistical analyses and data visualization techniques in spreadsheet data can be used to examine a dataset to identify data cleaning operations that need to be performed. The primary focus of data cleaning is to identify, correct errors, and check redundant data to develop reliable datasets. These steps improve the quality of analytical training data and enable the best decision-making.

3.1.5. Fill Missing Values. In standard practice, data values are often missing from your dataset. This could have occurred during data collection or due to data validation rules but must be considered regardless of missing values.

MinMaxScaler: We have applied Min-Max-Scaler directly to the ransomware dataset for the normalization of input variables. We have used the default configuration and the scaling values, which ranged from 0 to 1. The MinMaxScaler instance was defined by applying standard hyper-parameters, then calling the fit_transform () to pass it to the ransomware dataset to update and upgrade it.

From sklearn. Preprocessing import MinMaxScaler = transforms the features by scaling each element to a limited range. Each feature estimator scales individually and translates so that it falls within the scope of a given training set, between [0, 1]. The transformation can be written as:

$$\begin{aligned} X\text{std} &= X - X \cdot \text{Min axis} \\ &= 0X \cdot \text{Max axis} \\ &= 0 - X \cdot \text{Min axis} \\ &= 0, \end{aligned} \tag{1}$$

$$X\_\text{scaled} = X\_\text{std} * (\text{Max} - \text{Min}) + \text{Min}, \tag{2}$$

where, Min, Max = feature_range.

Missing data handling by elimination rows: The elimination of rows is simple and may be the most effective technique. It fails if many objects lose value. If most of the features have missing values, these features themselves can be eliminated.

Evaluate missing values: If only a fair % of the weight is missing, we can also use a simple-interpolation method (SIM) to fill this value.

Duplicate values in the dataset: Datasets can contain data objects that are duplicates of one another. This can happen if you say the same person inserts the same values multiple times.

The data frame.nunique () is a Pandas function that returns a series with the number of individual values over the requested axis. If the axis value is 0, all numbers of unique observations above the index axis have been found. Setting the axis value to 1 finds the number of individual comments from the column axis. Excluded NaN values can be provided as a feature from the number of unique numbers.

3.1.6. Wrangle Dataset. Wrangling dataset is a significant part of any data science project. Data wrangling is the process by which a data scientist transforms "raw data" to

make it more fruitful for analysis, and this improves the quality of data to be analyzed. There are several preprocessing techniques to cover the entire process.

Zero-value: Count zero values, and decide what to do with these values.

Data-exploration: Search data types of features, unique values, and data descriptions.

Feature engineering and reshaping: Converts the raw data into more useful formats. Examples of engineering functions are one-time coding, aggregation, and clustering.

3.1.7. Exploratory Data Analysis. There is only one column and only one missing value. This research project loads the row with the lost value and sees how to handle them. Let us use pandas' df to check and confirm that our data matches the original information. The output of the describe() function summarizes the statistical data of all numerical columns. Types of statistical data are shown in Figure 9.

3.1.8. Feature Engineering and Feature Extraction. A feature is a property that can be individually measured or a characteristic of an observed phenomenon. A dataset can be seen as a collection of data objects, often referred to as a set of data, points, vectors, patterns, events, samples, observations, or things. Data objects are structured by several characteristics that capture the basic properties of an object.Features are commonly known as variables, fields, attributes, and dimensions. Various types of features can occur while handling data.

3.1.9. Principal Component Analysis. PCA is an unsupervised algorithm that generates linear combinations of original features. PCA is classified in the order of explained variance. PCA is used to reduce the dimensions of an extensive dataset, as we used 57 features in our ransomware dataset. The first principal component, PC1, describes the most significant deviation in your dataset, PC2 describes the second largest variation, etc.

Refer to Figure 10; principal component analysis of a data matrix extracts the dominant patterns in the matrix in terms of a complementary set of score and loading plots. It is the responsibility of the data analyst to formulate the scientific issue at hand in terms of PC projections, PLS regressions, etc. Ask yourself or the investigator why the data matrix was collected and for what purpose the experiments and measurements were made. Specify before the analysis what kinds of patterns you would expect and what you would find exciting. In the initial study, look for outliers and strong groupings in the plots, indicating that the data matrix perhaps should be "polished" or whether disjoint modeling is the proper course. Use the resulting principal components to guide your continued investigation or chemical experimentation, not as an end.Hence, we reduce the dimensions by limiting the number of principal components that must be retained based on the accumulated variance. We choose only critical components as necessary to keep achieving a cumulative described variant of 88.

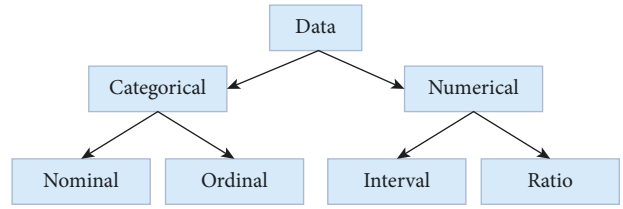


Figure 9: Types of statistical data.

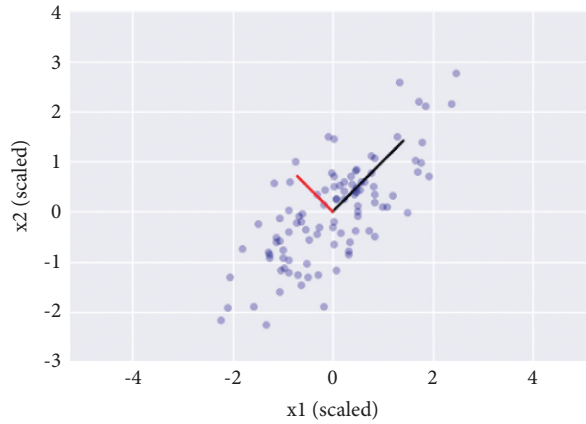


Figure 10: Principal component analysis.

3.1.10. Shuffle Data. In machine learning, we need to shuffle our learning data. The major problem we are facing is that, let us suppose, we have to train 70% of the data. But the given data frames contain the string, char, and integer values. During the testing of data, we have to test deals starting with either from A-R and then test S-Z values or start from 1–25 and then test the 25–100 values, respectively. This representative test does not look nice, or it is too necessary to perform this way. The main limitation is that we cannot train and test the same data, but given our current algorithm, there is nothing wrong with training and testing the same data values. There are multiple methods available for shuffling dataset values, but we have tried the given technique for data shuffling.

1st 25% - train 2nd 25% - train 3rd 25% - train 4th 25% - test Then, 1st 25% - test 2nd 25% - train 3rd 25% - train 4th 25% - train Then: 1st 25% - train 2nd 25% - test 3rd 25% - train 4th 25% - train Finally, 1st 25% - train 2nd 25% - train 3rd 25% - test 4th 25% - train.

We have done training and testing of all available data and then take the average of all values instead of simply shuffling the rows. For this purpose, we have created a Randomizing() function.

Another method we have used is the keyword "frac" argument which specifies the fragment of rows that should be returned to the random sample. "Frac = 1" means that all the rows should be returned in the random order. By using reset_index function with the parameter drop=true, the index can be reset with shuffling the data frame. Drop = True specifies the prevention of reset_index from generating an old column that contains an old index.

3.1.11. ML Algorithms. Many ML algorithms are used for the accuracy, classification, and prediction of ransomware over good ware. For this research project, Random-forest, DBSCAN, and KNN algorithms are being utilized to detect ransomware and zero-day attacks.

Random forest is a multi-level estimator that matches multiple decision tree classifiers in different subsets of the dataset and uses averages to improve forecast accuracy and control for over-fitting. The sample size is controlled with the max_samples parameter when bootstrap = True by default, and else the entire dataset is used to build all trees.

Random forest is not only an effective classifier but also useful for column selection. In this research project, we create a large, carefully constructed tree-set to predict target classes and use the usage statistics for each column to find the most informative subsets of columns. We made a large set (65535) of many flat trees (tree levels), and each tree was trained on a fraction (10 columns) of the total number of columns. The most predictable column is the column with the highest score.

Random-forest algorithm implementation. To discuss why we implement the random forest algorithm, let me tell you the following benefits of the random forest algorithm. Random forest and random forest classifiers are both algorithms that are often used to perform classification or regression problems. The random forest classifier can handle the missing values. If we have to use many trees in a forest, random-forest classifiers help avoid model overfitting problems. Random-forest creation pseudocode: 1. Select k features randomly from the "m" number of total features, where k << m. 2. Calculate the "d" number of nodes by using the best split-point concerning "k" features. 3. Use the best split of the given nodes into daughter node. 4. Repeat Steps 1–3 until you reach the "I" number of nodes. 5. Repeat Steps 1–4 by building a forest to generate "n" number of trees.

Random-forest prediction pseudocode: The trained random forest algorithm will be used to perform random-forest prediction 1-import test features and implements the rules of randomly generated decision trees to perform prediction as output and save it. 2-Calculate votes for all predicted outputs. A random forest algorithm will select the 3-Final prediction based on maximum votes. To make predictions using a trained random forest algorithm, we have to run the test feature through the rules of individual random trees.

Scikit-learn Python-based library calculates the importance of each node by using Gini importance for each decision tree to consider two child nodes only like a binary tree, as follows:

$$\begin{aligned} \text{Gini} &= 1 - i \\ &= Cpi2. \end{aligned} \tag{3}$$

The Gini formula uses class and probability to calculate the gini of each tuple on each node and find which branch is more likely to occur. In this formula, pi is used for the relative frequency of the class we are looking at in the dataset, and *c* is used for the number of classes in it. We also calculate entropy to determine how many nodes branch in a decision tree.

$$\begin{aligned} \text{Entropy} &= i \\ &= 1C - pi * \log 2(pi). \end{aligned} \tag{4}$$

Entropy is used for calculating the probability of a specific outcome to decide to check how to adjust nodes as a branch. Compared to the Gini index, entropy is more complex in the mathematical calculation because of its logarithmic function. Hence, we may say that random forest is the most practical algorithm with multiple types of datasets regarding regression or classification data. Random-forest is very easy to use and very fast to train data and find more accurate representations of decision trees.

(1) Training Dataset. At this stage, ML algorithms are trained to build the model. The model tries to learn the dataset and its multiple properties, which raises overfitting and underfitting problems.

Split training and testing dataset: Now, a dataset is used to test the hypotheses of our model. It remains unused and invisible until decisions are made about models and hyperparameters. After that, the model is applied to the test data to accurately measure how the model is performed when applied to real-world data. The training set defines a subset of the given dataset used for the ML model training. On the other hand, a subgroup is part of a dataset used for testing an ML model. The machine learning model uses a series of test-set to predict outcomes. It is a common practice that the dataset is divided into 70:30 ratios — 80:20 ratio of total datasets. You can take 70% or 80% of the model's training data and leave the remaining 30% or 20% for testing data as shown in Figure 11.

We need to use the python library; the first row divides the array from the dataset into random subsets of train and test datasets. The second line contains four variables:

X_train - training data features.

X_test - testing data features.

Y_train - training data's dependent variables.

Y_test - the independent variable used for test data.

The train_test_split () function has four parameters, the first and second used for dataset arrays. The test_size function shows the test-set size. The test size can be 0.5 or 0.3, or 0.2.

These values define the distributed ratio between the training and test set. In the end, random_state parameters always fix input for a ransom generator to get the same output.

Collecting statistical information of data Pandas' python library offers several very convenient statistical display methods for our datasets. We can use the technique "describe" to get a statistical summary of a dataset. The method described shows the information given below for each column.

Total value or count Mean (concerning all features of the dataset).

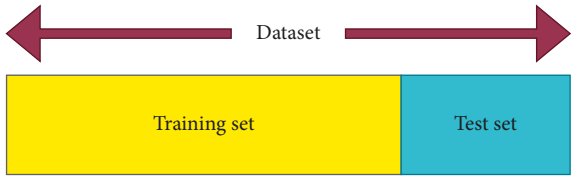


Figure 11: Dataset division into 70% training set and 30% test set.

Standard deviation (concerning all features of the dataset).

Min-max values (concerning all features of the dataset) 75%, 50%, 25% of quantity (concerning all features of the dataset).

This information is only calculated for numeric values. Prediction accuracy for ransomware detection. To check prediction accuracy, we apply data analysis techniques.

Analysis of data: there are two experimental stages: (1) the training stage and (2) the testing stage. All the machine learning algorithms described in the above section were trained and tested in the ransomware dataset [3]. The dataset is divided as follows: 70% for the training stage and 30% for the testing stage. The variables considered in the experiment are as follows:

Accuracy test (invisible data accuracy estimate): Accuracy is an indicator for the evaluation of classification models. Accuracy is the fraction of the predictions that our model gets right. Now we may define accuracy as follows:

$$\text{Accuracy} = \text{number of correct predictions total number of predictions.} \tag{5}$$

For binary classification, accuracy can also be calculated using true [T], false [F], positive [P], and negative [N] as follows:

$$\text{Accuracy} = TP + TNTP + TN + FP + FN. \tag{6}$$

(2) F1-score (harmonic mean for recall and precision, as given below).

The F1 score is a harmonious mean between recall & precision. The range for the F1 score is [0–1]. High-precision but lower recall gives you good accuracy but misses many instances that are not easy to classify. The higher the F1 score, the betterness of our model concerning performing. Mathematically, we may express as

$$F = 2PPVTPRPPV + TPR. \tag{7}$$

(3) Recall (True-positive-rate, as given below) True-positive-rate can be calculated as TP/(FN + TP). A TPR corresponds to the proportion of positive data points (PDP) correctly considered positive for all positive data points.

$$\begin{aligned} TPR &= \text{True PositiveTrue Positive} \\ &= \text{False Negative.} \end{aligned} \tag{8}$$

(4) Precision (Positive-predictive-value, as given below). Precision can be calculated by dividing the number of true-positive results by the true-positive results predicted by the classifier.

$$\begin{aligned} PPV &= \text{True PositiveTrue Positive} \\ &= \text{False Negative.} \end{aligned} \tag{9}$$

The results, accuracy, and fetching of the F1 classification measure are calculated with the Classification_report () function using the python library sklearn metrics library.

4. KNN and Density

K-Nearest Neighbors and Density-Based Algorithm (KNN) is a branch of supervised machine learning. KNN is very user-friendly for implementation. KNN also can perform complex classification tasks efficiently. The KNN is a non-parametric technique used for data classification & regression. The source contains the next k nearest training data for learning from feature space. The results depend on whether the classification is used in KNN or regression. It is a lazy learning algorithm because there is no specific training phase. In KNN, all training data are used when a new data point or instance is classified. KNN is a nonparametric training algorithm that does not require a dany population being analyzed by meeting specific parameters or assumptions; that is why it is considered a handy feature as most real-world data do not follow only theoretical assumptions like linear separation and vice versa. In this research, KNN is implemented with Python's popular library scikit-learn and uses some of its utilities. Figure 12 shows the closest points with the shortest distance.

The basic purpose of the KNN algorithm is to provide the easiest supervised machine learning algorithms. KNN measures the distance from the new data point to all other training data points. Any type of distance can be applied, like Euclidean, Manhattan, and so on. Then choose the closest data point, K, and K must be an integer. At last, KNN allocates the data points to the class with the most K points. Let us apply the KNN algorithm using a simple example. The task is to choose the class of new data points by considering X in the Pink color and the green color as a class. The coordinates of the data points are $x = 45$, $y = 50$.

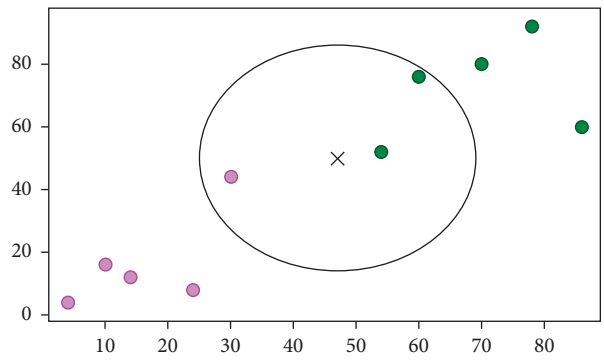


Figure 12: Closest points with the shortest distance.

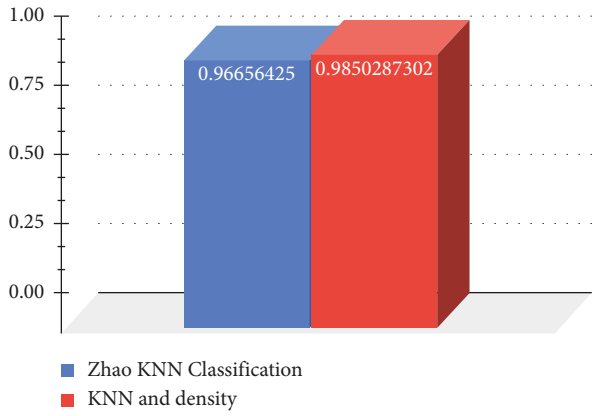


Figure 13: Zhao KNN classification vs. our KNN and density classification results.

Table 3: Classification report.

	Precision	Recall	F1-Score	Support
0	0.99	0.98	0.98	4805
1	0.99	0.99	0.99	8302
Accuracy			0.99	13107
Macro avg.	0.99	0.98	0.99	13107
Weighted avg.	0.99	0.99	0.99	13107
Final accuracy: 0.9846060814051052				

Let us suppose the k value is 3. KNN first measures the distance of point X from all other data points, then adopts the three closest points with the smallest distance concerning point X, as shown below. The three nearest points are surrounded. Now KNN will allocate a new point to a class that has a majority of the three closest points. As shown in Figure 13, there are two of the three most relative dots in the green color's class, and one belongs to the Blue color's class. As a result, the new data points are allocated to the green color's class.

5. Results

After successfully implementing KNN and density, we got the following fruitful results, as shown in Table. 3. The classification report shows the average performance as 0.99 w.r.t f1 sources, precision, support, and recall. This report also shows the accuracy by values, near about 99%. By using hyper-parameters-tuning, we have improved the performance by about 15% as compared to previously calculated adoptive clustering KNN and density-based algorithm's accuracy, as described in previous papers [26, 28]. Zhao also used KNN classification and random forest techniques with some additional features to achieve more accuracy of 97%.

Table 4: ROC score.

Precision	Recall
Best Leaf_Size	5
Best p	7
Best n_neighbors	1

The KNN and density algorithm recognize the classifiers of different shapes, sizes, densities, internal variants, and a few other features and are useful to remove noise and outliers automatically. The effectiveness of this algorithm is tested on multi-dimensional ransomware data. The key advantage of this algorithm is that it performs the classification process w.r.t K nearest neighbor. This algorithm is more suitable for incremental processes because this model is simple and can be efficiently working for large amounts of data processing. The Receiver Operating Characteristics (ROC) score shows the best results, as described in Table 3 that is, 0.9846060814051052.

According to these results, using the Grid Search method, the best leaf size = 5, where $p = 1$ shows the optimal distance technique used is Manhattan, and the best value for the K nearest neighbor is number 7, as described in Table. 4. The overall accuracy after testing the ROC score is about 0.98962542, which is about 99% as shown in Figure 13.

6. Conclusion

This research demonstrates the analysis and implementation of windows-based ransomware pre-attack detection using PE (.exe) files format. Many windows based on ransomware detection techniques and algorithms are available. These detection algorithms are working very effectively. But with the continued growth of zero-day attacks and the new generation of ransomware families' attacks on endpoint systems, there is a need to improve the malware detection techniques and algorithms. In this research, for the detection of ransomware in supervised ML, we used KNN and density-based algorithm and random forest with outstanding accuracy results of 98% and 99%, respectively. On the other hand, to detect zero-day attacks that lie under unsupervised ML, we implemented k-means and DBSCAN clustering algorithms to detect zero-day attacks. For this purpose, we need to fetch the number of clusters by using the elbow method to identify the best optimal value of k. DBSCAN randomly placed k-centroids, and there is no need to specify the number of clusters. Still, after analysis and implementation, the results show that K-means clustering leads to more efficient results than DBSCAN for many ransomware datasets. In unsupervised ML and zero-day attack detection,

k-means show better detection results. In supervised ML and ransomware detection, the random forest algorithm shows more accuracy in results than KNN and density-based algorithms. There are still more improvements required in ML algorithms to detect ransomware and zero-day attacks for multi-dimensional features.

7. Future Work

In the future, there are many opportunities for data scientists and researchers to get more improvement in this research concerning increasing the ransomware detection accuracy by using more multi-dimensional data features. Results can be improved by using different advanced ML algorithms with the latest ransomware datasets to gain highly accurate detection.

Data Availability

No datasets have been used or refereed in thise article. However, preliminary data can be found at Malwarebytes (2017): Cybercrime Tactics and Techniques, Report: https://www.malwarebytes.com/resources/webinars/cybercrime-tactics-techniques-q1-2017.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Moore, "Detecting ransomware with honeypot techniques," in *Proceedings of the 2016 Cybersecurity and Cyberforensics Conference (CCC)*, pp. 77–81, IEEE, Amman, Jordan, August 2016.

[2] A. Kharraz and E. Kirda, "Redemption: real-time protection against ransomware at end-hosts," in *International Symposium on Research in Attacks, Intrusions, and Defenses,* Springer Cham, Switzerland., Europe, 2017.

[3] L. Karthikeyan, G. Jacob, and B. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, no. 4, Pittsburgh, PA, USA, July 2011.

[4] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 646–656, 2013.

[5] E. Gandotra, D. Bansal, and S. Sofat, "Tools techniques for malware analysis and classification," *International Journal of Next-Generation Computing*, vol. 7, no. 3, 2016.

[6] A. Azmoodeh, A. Dehghantanha, M. Conti, and K.-K. R. Choo, "Detecting crypto-ransomware in IoT networks based on energy consumption footprint," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1141–1152, 2018.

[7] M. A. Habib, M. Ahmad, S. Jabbar, S. H. Ahmed, and J. J. P. C. Rodrigues, "Speeding up the internet of things: Leaiot: A lightweight encryption algorithm toward low-latency communication for the internet of things," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 31–37, 2018.

[8] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123–147, 2019.

[9] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *Journal of Computer Security*, vol. 19, no. 4, pp. 639–668, 2011.

[10] S. M. Muzammal, M. A. Shah, H. A. Khattak, and S. G. S. S. K. Jabbar, "Counter measuring Conceivable security threats on Smart Healthcare devices," *IEEE Access*, vol. 6, Article ID 20722, 2018.

[11] J. Landage and M. P. Wankhade, "Malware and malware detection techniques: a survey," *International Journal of Engineering Research and Technology*, vol. 2, no. 12, pp. 2278–0181, 2013.

[12] M. Conti, T. Dargahi, and A. Dehghantanha, "Cyber threat intelligence: challenges and opportunities," in *Cyber Threat Intelligence*, pp. 1–6, Springer Cham, Switzerland., Europe, 2018.

[13] Z. Zhao, J. Wang, and J. Bai, "Malware detection method based on the control-flow construct feature of software," *IET Information Security*, vol. 8, no. 1, pp. 18–24, 2014.

[14] S. Cesare, Y. Xiang, and W. Zhou, "Control flow-based malware VariantDetection," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 4, pp. 307–317, 2014.

[15] C. T. Lin, N. J. Wang, H. Xiao, and C. Eckert, "Feature selection and extraction for malware classification," *Journal of Information Science and Engineering*, vol. 31, no. 3, pp. 965–992, 2015.

[16] J. B. Park, K. S. Han, T. G. Kim, and E. G. Im, "A study on selecting key Opcodes for malware classification and its Usefulness," *Journal of KIISE*, vol. 42, no. 5, pp. 558–565, 2015.

[17] M. M. Ahmadian and H. R. Shahriari, "2entFOX: a framework for high survivable ransomware detection," in *Proceedings of the 2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pp. 79–84, IEEE, Tehran, Iran, September 2016.

[18] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know abnormal, find evil: frequent pattern mining for ransomware threat hunting and intelligence," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, 2017.

[19] S. Ruan, R. Mehmood, A. Daud, H. Dawood, and J. S. Alowibdi, "An adaptive method for clustering by fast search-and-find of density peaks: adaptive-dp," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 119–127, Perth, Australia, April 2017.

[20] M. Ahmad and S. A. F. G. Jabbar, "A sustainable solution to support data security in high Bandwidth Healthcare Remote locations by using TCP CUBIC Mechanism," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 249–259, 2020.

[21] B. Jethva, "A New Ransomware Detection Scheme Based on Tracking File Signature and File Entropy," (Doctoral Dissertation), B.Eng, Gujarat Technological University, Gujarat, India, 2019.

[22] M. M. Hasan and M. M. Rahman, "RansHunt: a support vector machines based ransomware analysis framework with integrated feature set," in *Proceedings of the 2017 20th International Conference of Computer and Information Technology (ICCIT)*, pp. 1–7, IEEE, Dhaka, Bangladesh, December 2017.

[23] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, pp. 1–7, Pittsburgh, PA, USA, July 2011.

[24] J. Liu, Z. Liu, C. Sun, and J. Zhuang, "A data transmission approach based on ant colony optimization and threshold proxy re-encryption in wsns," *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 23–31, 2022.

[25] X. Zhang and G. Wang, "Stud pose detection based on photometric stereo and lightweight YOLOv4," *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 32–37, 2022.

[26] M. A. Mirza, M. Ahmad, M. A. Habib, N. Mahmood, C. M. N. Faisal, and U. Ahmad, "CDCSS: cluster-based distributed cooperative spectrum sensing model against primary user emulation (PUE) cyber attacks," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5082–5098, 2018.

[27] F. Ullah, H. Naeem, S. Jabbar et al., "Cyber security threats detection in internet of things using deep learning approach," *IEEE Access*, vol. 7, Article ID 124379, 2019.

[28] B. Shi, L. Han, and H. Yan, "Adaptive clustering algorithm based on kNN and density," *Pattern Recognition Letters*, vol. 104, pp. 37–44, 2018.