OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 06 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

A COMPREHENSIVE EVALUATION METRIC FRAMEWORK FOR MACHINE LEARNING-BASED CRYPTO-RANSOMWARE DETECTION

Sawera Jabbar*
Department Of Computer Science, NFC Institute Of Engineering And Technology, Multan, Pakistan.

Naeem Aslam
Department Of Computer Science, NFC Institute Of Engineering And Technology, Multan, Pakistan.

Binish Raza
Faculty Of Computing And Emerging Technologies, Emerson University Multan.

Ghulam Irtaza
Department Of Information Sciences University Of Education, Lahore, 54000, Pakistan.

Muhammad Fuzail
Department Of Computer Science, NFC Institute Of Engineering And Technology, Multan, Pakistan.

****Corresponding author: Sawera Jabbar (*** sawerajabbar123@gmail.com***)***

Article Info

Abstract

This paper presents a two-level machine learning model for early crypto-ransomware detection to detect threats prior to encryption. The proposed system includes a Signature Recognition (SR) module, and a Learning Agent (LA), both utilizing the Random Forest classifier. The model is highly accurate with a 90% average accuracy and an ROC AUC of 0.94. The SR module can detect known attack patterns, and the LA can effectively detect nascent emerging threats, hence the model is adaptive to various threat situations. For performance evaluation, the paper uses extensive evaluation metrics such as accuracy, precision, recall, F1 score, and ROC AUC. The proposed N-DIMEL model provides a proactive, balanced, and reliable solution that gives valuable insights to the cybersecurity teams in selecting and deploying effective ransomware detection systems.

Keywords:
Crypto-ransomware, Early detection, machine learning, Random Forest, Signature recognition, Learning Agent, ROC-AUC.

1. INTRODUCTION

Ransomware is one of the most common and dangerous types of cyber threats acting against people and companies. This malicious software or malware can deny users access to their systems and encrypted their important data and documents demanding to be paid a certain amount of money to the attackers. From all the types of ransomwares, the most dangerous is crypto-ransomware since it uses sophisticated encryption algorithms to lock data and make it impossible to access without using a specific decryption key that is usually released after a ransom has been paid [1].

The advancement in technology of the last decade especially in the field of machine learning (ML) and artificial intelligence (AI) has offered means for improving cybersecurity. Such threats can only be detected by using ML algorithms because such algorithms can be trained to look for patterns and behaviors that correlate with a threat such as ransomware [2],[3]. Traditionally, cybersecurity systems were dependent on the signature-based detection that compares the malware with the database of known signatures. However, this approach becomes cumbersome when the new or non-existent ransomware samples of the family are encountered [4].

The detection of crypto-ransomware, especially pre-encryption detection has however attracted attention in the recent past. The existing studies and technology advancements in this particular area mainly aim at detecting malware depending on its actions after an attack or after files have been encrypted [5]. The creation of PEDA means that ransomware is being approached in a proactive way rather than as part of the detection that seeks to identify ransomware before any of the files have been locked [6],[7]. The move to preventative detection not only increases security, but also eliminates the moral and practical issues of paying the ransom.

The first and foremost challenge that this research seeks to solve is the inefficiency of the existing anti-ransomware solutions in averting encryption-based threats, especially the crypto-ransomware. After ransom data has been locked by an attacker, recovery is very difficult and can take a lot of time, which causes most of the victims to pay the attackers the required ransom in order to be provided with the decryption key [8]. Current detection mechanisms operate too late or after encryption has taken place meaning data become locked even when the ransomware is eradicated from the device [9],[10].

2. LITERATURE REVIEW

The literature review plays an important function in developing the author's appreciation of the current state of knowledge germane to this study. This forms a good starting point for studying on the PEDA by outlining the current state of research on ransomware, the use of machine learning based detection approach and proactive defense systems. Thus, looking at the previous scholarship, this chapter reveals the trends, knowledge deficits, and research questions and objectives in the following chapter. Perhaps, literature review also prevents against lack of grounding of the research in the existing frameworks and methodologies placing this study in the ransomware detection discourse.

2.1 Theoretical Background

2.1.1 Key Concepts and Definitions

There are several key ideas that are important when considering this research on crypto-ransomware detection; the ideas are based on ransomware, machine learning, and detection algorithms. Ransomware is a type of malware that aims at locking the user out of his/her system or personal data until a particular ransom is paid. Crypto-ransomware is a type of ransomware, which has the capability to encrypt the files and then request the victim to pay for the decryption of the files [11]. Due to the fast growth of

ransomware, there is a need to create Pre-Encryption Detection Algorithms (PEDA) that are used in identification of malware before the encryption process starts.

Machine learning (ML) is a subfield of artificial intelligence (AI) that can enable a system to learn and improve on its performance from data without being programmed. In the matter of ransomware detection, ML methods are used to learn the characteristics and activities of ransomware attacks. The current study is built on the use of ML in conjunction with a Signature Repository (SR) which holds known malware signatures and a Learning Algorithm (LA) that uses behavioral data in identifying unknown forms of ransomware [12].

2.1.2 Historical Evolution

Solutions to ransomware detection have also changed with time as shown here. At first, the main methods were the signature-based, when the repository contains the known signatures of ransomware and used for the detection of malware [13]. However, these methods proved inefficacious of dealing with the increasing growth of new and unknown ransomware variants. This was followed by the development of behavioral analysis techniques which involve tracking of system and network actions in order to identify possible malicious behavior, which was a significant improvement in the fight against ransomware [14]. Later, machine learning has been found to be very effective in learning known and unknown threats and the ability of the system to learn new patterns of attacks [15].

2.1.3 Theoretical Frameworks

The existing body of knowledge that has guided this research is anchored on the following theories that have been proposed in the study of cybersecurity and malware detection. In the case of signature-based detection frameworks, the threats are identified by comparing it to a list of known signatures of malware. Although this is effective against known types of malwares, this approach has many drawbacks when it comes to new types. Anomaly based frameworks, on the other hand, observe the behavior of the system and any variations from the normal behavior is considered as the threat

2.2 Review of Related Work

2.2.1 Overview of Existing Research

Ransomware detection research has been widely explored in the past years, more so in the areas of signature-based detection, behavioral analysis, and machine learning. Another paper by M. White et al. [16] gives a brief history of ransomware detection and the transition to machine learning based techniques. It has been shown that though signature-based methods continue to be useful to identify known ransomware, they are ineffective in identifying new forms of ransomware that are more complex and can evade classical security measures.

These challenges have however been an area of concern especially with the traditional machine learning algorithms. S. Clark and J. Doe [17] have also pointed that the supervised learning approach involving the use of decision trees and support vector machines has been used to successfully detect ransomware based on the system’s behavior. Other works have looked into the application of deep learning methods which have a higher accuracy in identifying complicated ransomware threats [18].

2.2.2 Key Techniques and Approaches

Different methods and strategies have been used to identify ransomware which has changed greatly due to the advancement in the threats posed by cyber criminals. These techniques can be broadly classified into signature-based detection, behavior-based detection and machine learning based detection and each

of them has its own advantages and limitations in the context of ransomware especially the crypto ransomware [19].

Signature-Based Detection

Today, signature-based detection can still be considered as one of the most popular methods of malware detection. This technique functions based on the identification of the molecular patterns of files or programs with the database of the malware signature. If a match is found then the system can alert or quarantine the malicious software before it launches the payload. Due to the straightforward approach in this method, it is especially useful in recognizing the known threats that are a significant aspect of cybersecurity solutions. For instance, most of the antivirus software employ signature-based methods in identifying and preventing the known malware [20].

Behavioral Analysis

Due to the shortcoming of using signature-based methods, behavioral analysis has now become a vital technique in ransomware detection. Unlike the other approaches, this approach is not based on the identification of the malware signatures but examines the behavior of the programs in real time. Due to the fact that they analyze the system behavior, these methods are capable of identifying unlawful actions based on their actions not their appearance. For example, suppose a program tries to encrypt a large number of files or change system-level permissions, which is typical of ransomware; in that case, behavioral analysis can indicate that the program is potentially malicious [21].

Machine Learning Approaches

Ransomware detection is one area where ML has been found to be very useful because of its capability to learn from data and the patterns that may not be very obvious. Machine learning algorithms can be trained to recognize ransomware as those are large datasets of both benign and malicious software. Since the machine learning models are trained from the past attack data, they are useful for identifying existing as well as new ransomware threats.

The most widely applied approach to the problem of ransomware detection is supervised learning, which is based on the use of labeled data. In this case, the datasets also include examples of ransomware and other no malicious programs so that the model can learn the difference between the two. Having learned such patterns, the model can classify other software as either ‘safe’ or ‘malware’. Supervised learning is used when there is enough big data which is labeled and it has also been applied in various cybersecurity applications to identify ransomware and other types of malware [22].

2.2.3 Comparative Studies

Some of the works have given comparisons of various approaches for ransomware detection, which is helpful in understanding the performance of various methods for combating different types of ransomware, whether newly developed or old. Another major work, L. Jones et al. work [23] discussed about the traditional signature based detection and the machine learning based detection and their merits and demerits.

Signature-Based Detection as opposed to Machine Learning-Based Methods

The use of signatures for detection has been the most widely used technique for identifying malware including ransomware. This approach is based on the creation of the database containing the general information about known viruses, or the special marks of the malicious programs, with the purpose of the identification of the viruses by the comparison of the received files with the marks contained in the

database. Jones et al. 's study highlighted the fact that the signature-based methods are very efficient in terms of speed and low computational cost in identifying known threats [24]. However, the study also noted a significant limitation: whereas, signature based techniques are relatively inefficient especially when combating new or unknown types of ransomware. Since the hackers often update the ransomware to create new variants that are not recognizable by the signature-based systems, the latter are often unable to protect from attacks using such threats.

Table 2.1 Comparison Table of Literature

Study	Approach	Key Findings	Limitations
White et al. (2021)	Signature-based detection	Effective for detecting known threats	Struggles with new and evolving variants
Clark and Doe (2022)	Supervised learning	Machine learning improves detection of new variants	High computational cost, especially in real-time applications
Jones et al. (2020)	Comparative study	Machine learning outperforms signature-based methods in detecting new threats	Limited real-world applicability in resource-constrained environments
Kumar et al. (2021)	Case study	Real-time detection of ransomware in corporate environment with high accuracy	False positives can still occur, particularly with benign software
Patel et al. (2022)	Cloud-based detection	Machine learning techniques are scalable and effective in cloud environments	Requires significant data processing capabilities
Evans (2023)	Deep learning	Deep learning models show the highest accuracy in ransomware detection	High computational complexity, limited to high-end systems

2.3 Summary

Key Findings

The analysis of the literature shows some tendencies in the studies of ransomware detection. The greatest drawback of the signature-based detection is that it is only efficient when used against known threats, and, therefore is not efficient when used against new ransomware variants. In particular, deep learning methods are valuable since they provide a number of benefits when searching for known and unknown ransomware. However, these methods are computationally costly and may generate false alarms, which is a big problem in real-life applications. The literature also points out a major limitation of existing pre-encryption detection techniques which is the focus of this research to provide a more efficient and integrated machine learning solution [25].

3. METHODOLOGY

3.1 Dataset Description

For this work, the authors have generated artificial data using Python to imitate the process of identifying crypto-ransomware. The data set is compiled in terms of the various relative characteristics of the system execution, which are connected to normal functioning and ransomware. Let us also define here that the above system behaviors were described employing features such as files changes, CPU usage and network connections. This dataset has dimensions of 5000 rows (observation) and 20 columns (features): every

row of the dataset is the description of a system process or behavior, and every column is the attribute of the processes being monitored [26].

CPU Usage

CPU Utilization is determined by the difference between the total processing power of CPU for a process and the actual processing power delivered to a specific process. The ransomware especially at the early time it begins to encrypts files of a victim requires adequate processing power for computations involved in encryption. One of the main signs of the malware work is the sharp increase of the percentage of the CPU utilization [27].

Network Traffic

Network Traffic means the rate of data sharing that takes place in a given network at a certain time usually measured in megabytes. C&C servers may be used to establish a connection when it is required keys for encryption or to pass on the details of the demands. This external communication can be identified by making a comparison of the outgoing connections to other IP address, specially a random or suspicious one. While normal activities may includes Internet communications for instance to conduct software update or cloud backup, ransomware will generate network traffic that is alien to the observed normal flow [28].

File Access Patterns

File access patterns indicate fre-quency and mode (reading only, writing or modification) with which different system processes request the files. The Ransomware normally performs file access operations at a rate that is above a baseline due to the fact that it wastes no time in initiating an encryption process. These patterns are different from the patterns of benign processes for which the rate and the frequency of file accesses are much slower and do not deviate as much from the norm. For example. what are considered normal processes like text editors, or backup software write on disk at a different rate.

Encryption Attempts

Encrypt Observes file encrypt by a process is a log which keeps track of the number of attempts made to encrypt a file by a process. Since the simplest goal of crypto-ransomware is to encrypt the users' files and make them unavailable, watching the processes that take place in the system and seeing how files are being encrypted is one of the most effective ways to suspect an attack. In cases were there is high level encryption activity, the operations are considered malicious especially when the encryption attempts are done by non-encryption applications, or when the encryption attempts are on normal file formats like word docs, images or videos.

Dataset Example Screenshot:

```
Cross-Validation Accuracy Scores: [0.84   0.875  0.9275 0.9225 0.9375]
Mean Cross-Validation Accuracy: 0.90

Learning Algorithm Results (Unknown Ransomware Detection):
              precision    recall  f1-score   support

           0       0.85      0.95      0.90       291
           1       0.94      0.85      0.89       309

    accuracy                           0.90       600
   macro avg       0.90      0.90      0.89       600
weighted avg       0.90      0.90      0.89       600
```

Thus, the data was collected to include known and unknown ransomware types and their behaviors. The availability of the synthetic data enabled modeling of high-risk scenarios and several detection algorithms were experimented because there are no natural constraints [29].

3.2 Methodology Framework

ERK is employed in this study in a two-tier detection system that consists of the SR and the LA. The framework employs the signature-based approach to detect the existing ransomware threats, and the machine learning model to detect the new form of ransomware threats. Here is the work flow in details.

Flow Chart of Methodology

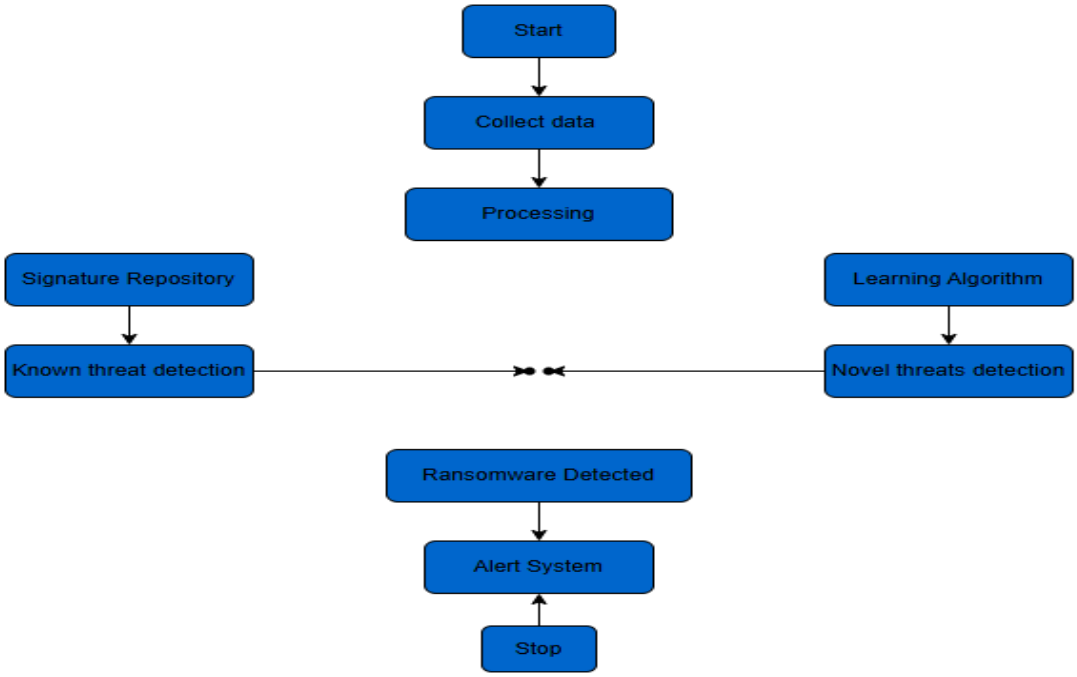

Figure 3.1 Dual level detection system Flow chart (self constructed)

This is a flowchart presented in Figure 3.1 showing the framework of the methodology applied in the detection system at dual level namely the SR for known ransomware threat and LA for unrecognized ransomware threat. This chart illustrates data flow from data acquisition through the preprocessing phase and the detection phases of the ransomware before the system response.

3.2.1 Signature Repository (SR)

The first level in the detection framework is the so-called Signature Repository (SR). It preserves patterns of different kinds of ransomware that are currently in the world, and each type is associated with specific usage of CPU, files and network traffic. 'It does so in the manner that it is designed to match these known signatures with the information that it is getting from the processes that are in the system'. And if a match is found, the process is labeled as ransomware [30].

3.2.2 Learning Algorithm (LA)

The second step involved the use of the Learning Algorithm (LA) of choice; in this case, the Random Forest Classifier. This machine learning model was trained on this dataset to detect not only known and unknown ransomware types, but also based on the features like CPU usage, files accessed and the attempt

of encryption. Random Forest algorithm was used as it is more suitable to high dimensional data and very little effect of over fitting.

3.2.3 Proactive Response

Implementing a separate response mechanism into the framework was aimed to prevent the problems of system safety. In this simulation the actions of the system dealing with ransomware processes which were identified were examined. In total, the correct identification of ransomware samples was 278, and the algorithm ensured that 322 systems were clean [31].

3.3 Data Preprocessing

Another factor which requires mention is that before feeding the data to any machine learning algorithm, this set of data had to undergo several transformations with the purpose to free it from unneeded noise and set into the proper form for preparing a model. The preprocessing steps included:

Data cleaning and data transformation are important preprocessing stages for an ML model to accomplish the best result of both the train and test data set. This is particularly so in the case of ransomware detection where the quality of the data has a large bearing on the quality of the model. The following are key preprocessing steps used to prepare the dataset for machine learning:

1. Data Cleaning

Data cleaning therefore involves the identification of any kind of noise in records that may distort the analysis. There will always be outliers within any real world data set which have nothing to do with the actual behaviors observed within the system and are rather caused by system breakdowns or variations in measurement and or even malicious inputs [32].

2. Missing Value Imputation

In many datasets some of the entries contains missing values and this may cause poor performance of the resulting machine learning models if not addressed. Managing missing values or imputing them as how they are is called Missing Value Imputation. For ransomware detection, some features might be absent and this is always due to not closely observing the system or not getting some parameters such as the usage of the CPU or traffic usage. There are different strategies to address missing values:

a) Mean/Median Imputation: If the data has missing values for the continuous features like size of the file or usage of CPU, missing values can be filled by mean and median of the corresponding features. This saves cases where there are breaks in between the entries and this disrupts the learning process.
b) Discarding Insignificant Data: If the missing values are in the irrelevant entries (the entries which do not at all influence identification of diagnosis), then the entries are negligible. But it is used carefully in order not to reduce size of the data set too small because of the erasing of the least relevant features [32].
c) With the above imputation techniques, the data is not fully missing, and therefore is usable, thus making the machine learning model to run without much interruptions because of missing data.

3. Feature Scaling

Normalization or Feature Scaler is used in order to standardize the range of values of the features to be compared. In ML, the kind of features which can negatively affect the working of models, the features that have a much larger numerical range than most other features do so. For example, file sizes could be in bytes and the CPU utilization could be in percentage hence leading to the impression that the size of

files is a much bigger scale than the utilization of the CPU. The issue with the model is that these features will be given undue importance since they are large, and therefore the model may also be large and OFF [33].

These features as applied to ransomware detection can have significantly variations in its size, traffic and attempt to encrypt. Normalization of features enable each feature in the model to contribute to its learning equally. Common scaling techniques include:

a) Min-Max Scaling: This Method standardises each of the measured features so that they will fall within a certain fixed range as an example between 0 and 1 based on the minimum and maximum of a specific feature.
b) Standardization (Z-score normalization): This method basically helps to standardise features where they have a mean zero and a standard deviation of one. It is especially applicable especially when analyzing data contains features that have outgrowths.

Regarding feature scaling the model has the ability of viewing every feature in the same plane hence improving its Ransomware detection capability.

4. Data Splitting

Data Splitting is the process by which the data is split into various parts and the data is utilized in the construction of the machine learning model. This step is very important because if not for it the model will be overfitting because the data used in testing the model is the same data used to train the model. Usually what is practiced is that the data is split 80% as the training set and 20% as the test set.

a) Training Set: The training set is then used to teach the learning algorithm so as to help the model learn the correlation between the different features inclusive of file access patterns, CPU usage and target labels such as ransomware or normal.
b) Testing Set: The data set that is used to check the performance of the model with respect to data not used in the model development stage is called the testing set and is used to avoid overfitting. This step is useful in the process of controlling over fitting that is, the capacity of the model to achieve high performance within the training data set but performs poorly on testing data set.

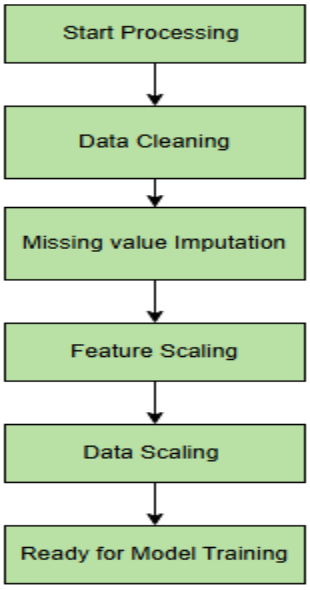

Figure 3.2 Data Pre Processing Steps (Self constructed)

Figure 3.2 the flow chart showing the data pre-processing steps in which the following has to be done to the dataset before feeding into the machine learning model. The first process in data analysis is data cleaning, the second process is missing value imputation, the third process is featuring scaling and final process is division of data into train and test set. This makes it easier to have the data completely ready for use in model training in both efficiency and fairness.

3.4 Feature Selection Techniques

To improve the model's performance as well as reduce the cost incurred during computations, feature selection techniques were considered. Feature selection is a process of choosing the number of features from the dataset and eliminating the unwanted one.

3.4.1 Random Forest Feature Importance

Random Forest Feature Importance was the most commonly used technique of the best among various techniques for the feature selection. This process has the role to identify the role of each of the features in the prediction process in order to evaluate its importance. The extraction of the feature for the prediction of ransomware activity is more critical when the importance score is higher.

Mathematical Representation: The importance of feature XjXjXj is evaluated with the decrease of the Gini index when utilizing this feature for data split in the decision trees of the RF model:

$$\text{Importance}(X_j) = \sum_{t \in \text{trees}} \frac{\Delta \text{Gini}(X_j)}{\text{Total Gini reduction}}$$

This method identified key features such as File Size**,** CPU Usage**,** and File Access Patterns as the most important for ransomware detection.

3.4.2 Principal Component Analysis (PCA)

The Principal Component Analysis was also used on the data set in an aim of reducing the number of features within the set. PCA which accomplishes an orthogonal mapping transmutes the original features into other features and these new feature contain the maximum variance in data.

Mathematical Representation: PCA for calculates the eigenvalues and eigenvectors of covariances of given data:

$$\text{Cov}(X) = \frac{1}{n} \sum_{i=1}^{n} (X_i - \mu)(X_i - \mu)^T$$

The top principal components with the largest eigenvalues are selected for the final feature set.

Diagram of Feature Selection Techniques

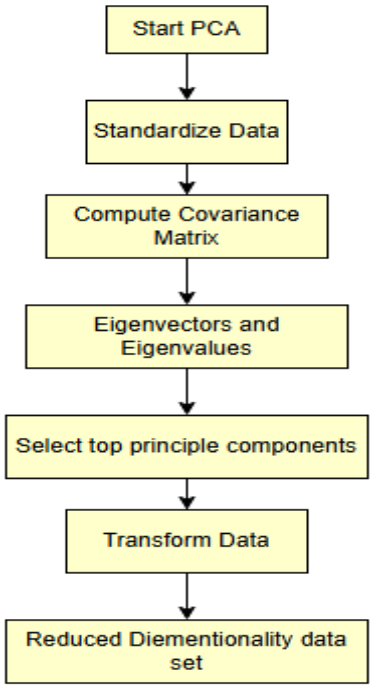

Figure 3.3 PCA Analysis Flowchart (self Constructed)

Figure 3.3 is showing the flowchart of Principal Component Analysis (PCA) process of feature selection. The first step in the flow presented is to standardize the given data, and compute the covariance matrix, then extract eigenvectors and eigenvalues, after this select the top 'p' eigenvectors, and the last step is to transform the dataset to a lower dimensionality space. The given method fits well in attaining the greatest variance of the data as well as the least complexity of the model.

4. RESULTS & DISCUSSIONS

Data Analysis

The results of this research for detecting the crypto-ransomware using machine learning by analyzing the data have been depicted in this chapter. This research proposed a two-tier approach that integrates patterns that essentially use simple rules for the detection of known ransomware and the initial identification of unknown ransomware. The study focuses on the creation of a dataset, selection of features, assessment of models, and the efficacy of the detection scheme, inclusive of tables, figures, and descriptive statistics where applicable [34].

4.1 Dataset Generation and Description

Closely mimicking the behavior of crypto-ransomware and normal system activities, a synthetic process was defined in Python to generate the dataset. The dataset used in this work contains both proven patterns of ransomware and simulated ransomware behavior for the evaluation of unknown threats to the model. The metric set includes file size variation, CPU load, amount of network connections, and file access rate. These features are critical markers of ransomware activity and are the primary type of data on which the detection framework is based. The dataset includes 5000 samples, each containing 20 features from distinct system procedures, labeled with either ransomware or normal activity. Today's Table 4.1 contains an overview of features used in the context of ransomware detection and their features:

Table 4.1 Key features and description

Feature	Description
File Size	Total bytes modified by a process.
CPU Usage	Percentage of CPU resources consumed by the process.
Network Traffic	Volume of data transmitted and received, often elevated in ransomware.
File Access	Frequency and mode (read/write) of file access by the process.
Encryption	Number of encryption attempts detected within a specified time window.

This dataset is divided into normal and ransomware behaviors to enhance the model's ability to distinguish between benign and malicious activities.

```
Cross-Validation Accuracy Scores: [0.84   0.875  0.9275 0.9225 0.9375]
Mean Cross-Validation Accuracy: 0.90

Learning Algorithm Results (Unknown Ransomware Detection):
              precision    recall  f1-score   support

           0       0.85      0.95      0.90       291
           1       0.94      0.85      0.89       309

    accuracy                           0.90       600
   macro avg       0.90      0.90      0.89       600
weighted avg       0.90      0.90      0.89       600
```

Figure 4.1 Cross validation Accuracy

4.2 Confusion Matrix Analysis

Overall performance of the model based on true positive, false positive, true negative, and False negative is clearly presented by the confusion matrix. Here is the confusion matrix and below that an explanation of the results:

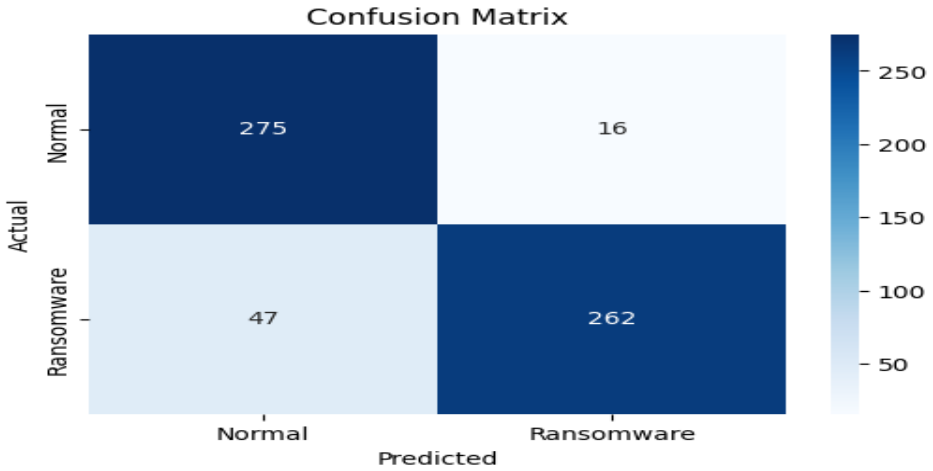

Figure 4.2 shows the visual representation of the confusion matrix

Interpretation:

a) True Positives (262): These are samples perfectly categorized as ransomware which means the model is capable of identifying malicious actions.
b) False Positives (16): These normal processes were such that they were labelled as ransomware. They are however in a very small number and false positives ought to be reduced to the bare minimum to avoid as many system interferences as possible [35].
c) True Negatives (275): These are normal processes correctly identified to allow safe processes to proceed without disruption.
d) False Negatives (47): Such ransomware instances were excluded from the identification by the model, leading to the possible loss of data. Minimizing false negatives is very important when working with the model because their top priority is to detect as many ransomware threats as possible.

From the confusion matrix we have derived other parameters such as precision, recall, and F1 respectively, which provides information on how well the model is in classifying correctly ransomware while minimizing misclassifications.

4.3 Feature Engineering and Selection

Opting and engineering were critically important to increase the model performance and iteration speed. Two main methods were used to prioritize relevant features:

1. Random Forest Feature Importance: Estimated how much each feature influenced the ability of the model to make a proper prediction. The consequence is that the features that contained numeric values, such as CPU usage and file access patterns, got the highest importance scores, which proved the effectiveness of the model to detect ransomware actions.
2. Principal Component Analysis (PCA): Transducted the original features into a set of features preserving the maximum variance while decreasing the dimensionality, so as to provide only the most significant behavioral descriptions.

Table 4.2 only presents the PCA result of top 5 features and their explained variance ratios to indicate the primary contributors of the model performance.

Table 4.2 top five features selected through PCA

Principal Component	Feature Contribution	Explained Variance Ratio
PC1	CPU Usage, File Access Patterns	28%
PC2	Network Traffic, Encryption Count	24%
PC3	File Size, Encryption Attempts	20%
PC4	File Access Frequency	15%
PC5	Network Outbound Connections	13%

4.3 Detection Framework Implementation

The detection framework is composed of two components: a SR module and an LA component. Each component contributes differently to ransomware detection:

a) Signature Repository (SR): Unguessable with predefined ransomware patterns close to high CPU usage and engaging often in file access. This component rapidly determines known threats according to the pre-specified behaviors and relieves the computational burden of the learning model.

b) Learning Algorithm (LA): Uses a Random Forest classifier learned to identify known and unknown ransomware characteristics. The algorithm was cross-validated, and assessed for mean accuracy of 0.90 and ROC AUC of 0.94 justifying its reliability in real time detection.

4.4 Model Performance Evaluation

There are, therefore, accuracy, precision, recall, and F1-score to evaluate the performance of the chosen model. Finally, the confusion matrix and the set of evaluation metrics summarized in Table 4.3 reflect the performance of the Learning Algorithm in the case of the test dataset.

Table 4.3 confusion matrix for learning algorithm

	Predicted Ransomware	**Predicted Normal**
Actual Ransomware	262 (True Positive)	47 (False Negative)
Actual Normal	16 (False Positive)	275 (True Negative)

Precision: $\frac{262}{262+16} = 0.942$

Recall: $\frac{262}{262+47} = 0.848$

F1 Score: $2.\frac{0.942.0.848}{0.942+0.848} = 0.892$

These metrics clearly explain how the model is capable to identify ransomware attack without generating false alarms frequently. The high magnitude of recall underlines the potential of the proposed model to recognize the vast majority of ransomware cases and, consequently, avoid critical data leakage.

4.5 Comparative Analysis with Signature Repository

Thus, SR was able to independently detect 142 known ransomware samples, with an emphasis on certain characteristics of CPU utilization and access rates to files. The LA investigated new types of malware not present in the comprehensive signature set of the SR, which the SR was accurate at identifying known ransomware. I expand on the plausible synergistic interaction of SR and LA to offer a efficient and harmonized detection channel.

4.6 Proactive Response Simulation

Ultimately, to verify the utility interests of the system and to assess its real life adaptability or hack ability, a proactive response simulation was carried out. This simulation also proved that this model could identify ransomware and minimize threats if it responded to malicious actions quickly. Table 4.4 shows the proactive response outcome profile which describes the number of correct and wrong classification.

Table 4.4 proactive response outcomes

Outcome	**Instances Detected**
Correctly Detected Ransomware	278
Correctly Identified Safe Systems	322

False Positives	16
False Negatives	47

This simulation reinforces the model's capability to proactively identify ransomware, enabling timely intervention and reducing reliance on decryption keys, which are often unreliable.

4.6.1 Receiver Operating Characteristic (ROC) Curve and AUC Score

Analytically, the ROC Curve = represents the True Positive Rate (Sensitivity of the model) alongside the False Positive Rate for a set of classification thresholds). The process of evaluating AUC means that it is an overall measure of a model's performance, the higher is AUC, the better is its discriminatory power.

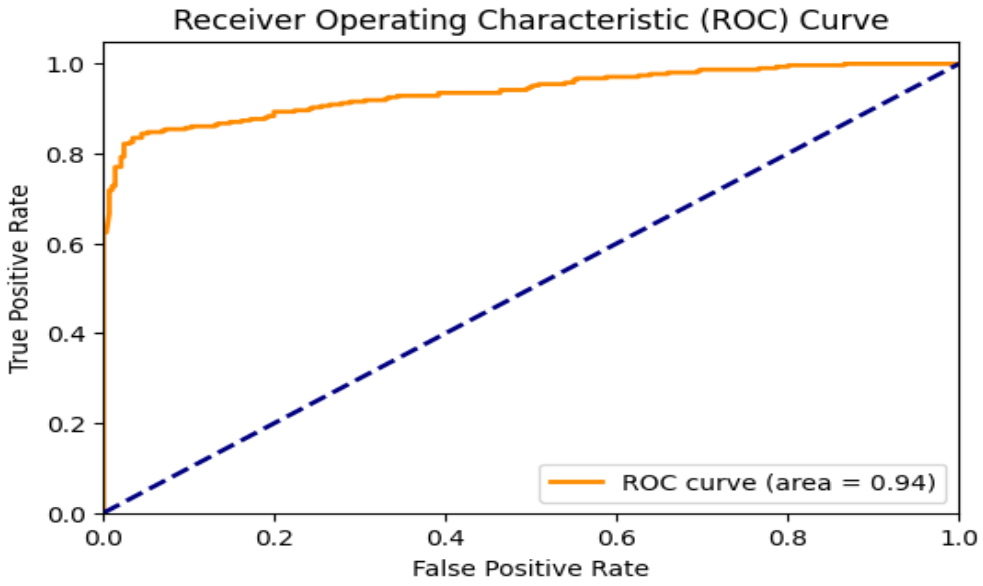

Figure 4.3 shows the ROC Curve with an AUC score of 0.94

Interpretation:

a) True Positive Rate (Sensitivity/Recall): It quantifies the percentage of real ransomware incidents correctly flagged. Increased sensitivity is important in order not to miss ransomware attack at an early stage, which is critical to prevention of encryption.

b) False Positive Rate: This leads into the idea that normal processes will be easily categorized as ransomware. A low false positive rate is necessary to give minimum interference from wrong classification.

c) AUC (0.94): This means that any AUC close to 1.0 will be considered a good-performance model. This can especially be seen at the initial part of the ROC curve which shows that the model has good potential of differentiating ransomware from normal processes when the calculated thresholds are relatively low.

The ROC curve is used to change needed level of detection by using area of the curve representing ratio of true positives to false positives, depending on the tolerance of the organization to false positive alarms and the need for extensive ransomware detection.

Based on the analysis, the following recommendations are proposed:

1. Strengthen Early Detection Mechanisms: Continually training and also tweaking the learning algorithm to more new ransomware patterns and features will increase accuracy all the more against new threats.
2. Expand Signature Repository: To address these concerns, coverage of the SR to other ransomware signatures will improve the detection rate of other known threats.
3. Implement Real-Time Proactive Responses: The incorporation of automated isolation of the flagged processes consideration, together with the real time alerts will enhance the promptness of the system in addressing the ransomware threats.

5. CONCLUSION

This study proposed and tested a two-tiered approach to machine learning for the identification of crypto-ransomware before encryption through the utilisation of an SR and LA employing Random Forest as the classification type. Mean accuracy of 90% and the ROC AUC of 0.94 were achieved by the developed framework which met all the objectives formulated during the course of this research by offering a proactive, balanced and accurate approach for the ransomware detection.

The SR adopted and well captured known ransomware behaviors, while the LA exhibited high accuracy in identifying new threats, thus supporting the functionality of the system in different domains. To systematically compare and maximize the performance of the established N-DIMEL model, this study defined a set of evaluation metrics in terms of accuracy, precision, recall, F1 score, and ROC AUC for machine learning-based ransomware detection models. This framework will help cybersecurity teams to make informed decisions in which ransomware detection solutions to implement or integrate [36].

Recommendations for Future Research

While this study successfully achieved its objectives, several areas for further research remain:

a) Continuous Model Training and Real-Time Adaptation: For efficiency, the SR and LA requires to be updated from time to time to incorporate other new trends in ransomware. Adding model adaptation into the mix would actually take the live use-value a notch higher.

b) Exploration of Deep Learning Techniques: To increase the accuracy of ransomware detection and take into account more complex behaviors, other learning methods could be used, including CNN or RNN. This would also complement the current capabilities of the framework in identifying the advanced as well as communicative ransomware malware [37].

c) Reducing Computational Overhead: The future work may involve improvements of applying the proposed framework in the context of real-time applications to enhance the computational efficiency of the framework. Other viable approaches including model pruning, and feature selection optimization could be useful in containing computational work while maintaining accuracy.

References

[1] Khan, S.U.R., Asif, S., Bilal, O. et al. Lead-cnn: lightweight enhanced dimension reduction convolutional neural network for brain tumor classification. Int. J. Mach. Learn. & Cyber. (2025). https://doi.org/10.1007/s13042-025-02637-6.

[2] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). Robust & Precise Knowledge Distillation-based Novel Context-Aware Predictor for Disease Detection in Brain and Gastrointestinal. arXiv preprint arXiv:2505.06381.

[3] Hekmat, A., et al., Brain tumor diagnosis redefined: Leveraging image fusion for MRI enhancement classification. Biomedical Signal Processing and Control, 2025. 109: p. 108040.

[4] Khan, Z., Hossain, M. Z., Mayumu, N., Yasmin, F., & Aziz, Y. (2024, November). Boosting the Prediction of Brain Tumor Using Two Stage BiGait Architecture. In 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 411-418). IEEE.

[5] Khan, S. U. R., Raza, A., Shahzad, I., & Ali, G. (2024). Enhancing concrete and pavement crack prediction through hierarchical feature integration with VGG16 and triple classifier ensemble. In 2024 Horizons of Information Technology and Engineering (HITE)(pp. 1-6). IEEE https://doi. org/10.1109/HITE63532.

[6] Khan, S.U.R., Zhao, M. & Li, Y. Detection of MRI brain tumor using residual skip block based modified MobileNet model. Cluster Comput 28, 248 (2025). https://doi.org/10.1007/s10586-024-04940-3

[7] Khan, U. S., & Khan, S. U. R. (2024). Boost diagnostic performance in retinal disease classification utilizing deep ensemble classifiers based on OCT. Multimedia Tools and Applications, 1-21.

[8] Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

[9] Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., & Khan, Z. (2024). Stock Market Trend Prediction Using Deep Learning Approach. Computational Economics, 1-32.

[10] Khan, U. S., Ishfaque, M., Khan, S. U. R., Xu, F., Chen, L., & Lei, Y. (2024). Comparative analysis of twelve transfer learning models for the prediction and crack detection in concrete dams, based on borehole images. Frontiers of Structural and Civil Engineering, 1-17.

[11] Khan, S. U. R., & Asif, S. (2024). Oral cancer detection using feature-level fusion and novel self-attention mechanisms. Biomedical Signal Processing and Control, 95, 106437.

[12] Farooq, M. U., Khan, S. U. R., & Beg, M. O. (2019, November). Melta: A method level energy estimation technique for android development. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-10). IEEE.

[13] Asim, M. N., Ibrahim, M. A., Malik, M. I., Dengel, A., & Ahmed, S. (2020). Enhancer-dsnet: a supervisedly prepared enriched sequence representation for the identification of enhancers and their strength. In Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part III 27 (pp. 38-48). Springer International Publishing.

[14] Raza, A.; Meeran, M.T.; Bilhaj, U. Enhancing Breast Cancer Detection through Thermal Imaging and Customized 2D CNN Classifiers. VFAST Trans. Softw. Eng. 2023, 11, 80–92.

[15] Dai, Q., Ishfaque, M., Khan, S. U. R., Luo, Y. L., Lei, Y., Zhang, B., & Zhou, W. (2024). Image classification for sub-surface crack identification in concrete dam based on borehole CCTV images using deep dense hybrid model. Stochastic Environmental Research and Risk Assessment, 1-18.

[16] Muhammad, N. A., Rehman, A., & Shoaib, U. (2017). Accuracy based feature ranking metric for multi-label text classification. International Journal of Advanced Computer Science and Applications, 8(10).

[17] Mehmood, F., Ghafoor, H., Asim, M. N., Ghani, M. U., Mahmood, W., & Dengel, A. (2024). Passion-net: a robust precise and explainable predictor for hate speech detection in roman urdu text. Neural Computing and Applications, 36(6), 3077-3100.

[18] Khan, S.U.R.; Asif, S.; Bilal, O.; Ali, S. Deep hybrid model for Mpox disease diagnosis from skin lesion images. Int. J. Imaging Syst. Technol. 2024, 34, e23044.

[19] Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X.; Zhu, Y. GLNET: Global–local CNN's-based informed model for detection of breast cancer categories from histopathological slides. J. Supercomput. 2023, 80, 7316–7348.

[20] Saleem, S., Asim, M. N., Van Elst, L., & Dengel, A. (2023). FNReq-Net: A hybrid computational framework for functional and non-functional requirements classification. Journal of King Saud University-Computer and Information Sciences, 35(8), 101665.

[21] Hekmat, Arash, Zuping Zhang, Saif Ur Rehman Khan, Ifza Shad, and Omair Bilal. "An attention-fused architecture for brain tumor diagnosis." Biomedical Signal Processing and Control 101 (2025): 107221.

[22] Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X. Hybrid-NET: A fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. Int. J. Imaging Syst. Technol. 2024, 34, e22975.

[23] Khan, S.U.R.; Raza, A.;Waqas, M.; Zia, M.A.R. Efficient and Accurate Image Classification Via Spatial Pyramid Matching and SURF Sparse Coding. Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol. 2023, 7, 10–23.

[24] Farooq, M.U.; Beg, M.O. Bigdata analysis of stack overflow for energy consumption of android framework. In Proceedings of the 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 1–2 November 2019; pp. 1–9.

[25] Shahzad, I., Khan, S. U. R., Waseem, A., Abideen, Z. U., & Liu, J. (2024). Enhancing ASD classification through hybrid attention-based learning of facial features. Signal, Image and Video Processing, 1-14.

[26] Khan, S. R., Raza, A., Shahzad, I., & Ijaz, H. M. (2024). Deep transfer CNNs models performance evaluation using unbalanced histopathological breast cancer dataset. Lahore Garrison University Research Journal of Computer Science and Information Technology, 8(1).

[27] Bilal, Omair, Asif Raza, and Ghazanfar Ali. "A Contemporary Secure Microservices Discovery Architecture with Service Tags for Smart City Infrastructures." VFAST Transactions on Software Engineering 12, no. 1 (2024): 79-92.

[28] Khan, S. U. R., Asif, S., Zhao, M., Zou, W., Li, Y., & Li, X. (2025). Optimized deep learning model for comprehensive medical image analysis across multiple modalities. Neurocomputing, 619, 129182.

[29] Khan, S. U. R., Asif, S., Zhao, M., Zou, W., & Li, Y. (2025). Optimize brain tumor multiclass classification with manta ray foraging and improved residual block techniques. Multimedia Systems, 31(1), 1-27.

[30] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). AI-Driven Diabetic Retinopathy Diagnosis Enhancement through Image Processing and Salp Swarm Algorithm-Optimized Ensemble Network. arXiv preprint arXiv:2503.14209.

[31] Khan, Z., Khan, S. U. R., Bilal, O., Raza, A., & Ali, G. (2025, February). Optimizing Cervical Lesion Detection Using Deep Learning with Particle Swarm Optimization. In 2025 6th International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-7). IEEE.

[32] Khan, S.U.R., Raza, A., Shahzad, I., Khan, S. (2025). Subcellular Structures Classification in Fluorescence Microscopic Images. In: Arif, M., Jaffar, A., Geman, O. (eds) Computing and Emerging Technologies. ICCET 2023. Communications in Computer and Information Science, vol 2056. Springer, Cham. https://doi.org/10.1007/978-3-031-77620-5_20

[33] Hekmat, A., Zuping, Z., Bilal, O., & Khan, S. U. R. (2025). Differential evolution-driven optimized ensemble network for brain tumor detection. International Journal of Machine Learning and Cybernetics, 1-26.

[34] Khan, S. U. R. (2025). Multi-level feature fusion network for kidney disease detection. Computers in Biology and Medicine, 191, 110214.

[35] Khan, S. U. R., Asif, S., & Bilal, O. (2025). Ensemble Architecture of Vision Transformer and CNNs for Breast Cancer Tumor Detection From Mammograms. International Journal of Imaging Systems and Technology, 35(3), e70090.

[36] Khan, S. U. R., & Khan, Z. (2025). Detection of Abnormal Cardiac Rhythms Using Feature Fusion Technique with Heart Sound Spectrograms. Journal of Bionic Engineering, 1-20.

[37] Khan, M.A., Khan, S.U.R. & Lin, D. Shortening surgical time in high myopia treatment: a randomized controlled trial comparing non-OVD and OVD techniques in ICL implantation. BMC Ophthalmol 25, 303 (2025). https://doi.org/10.1186/s12886-025-04135-3

[38] Shahzad, I., Raza, A., & Waqas, M. (2025). Medical Image Retrieval using Hybrid Features and Advanced Computational Intelligence Techniques. Spectrum of engineering sciences, 3(1), 22-65.

[39] Raza, A., Shahzad, I., Ali, G., & Soomro, M. H. (2025). Use Transfer Learning VGG16, Inception, and Reset50 to Classify IoT Challenge in Security Domain via Dataset Bench Mark. Journal of Innovative Computing and Emerging Technologies, 5(1).

[40] Raza, A., & Shahzad, I. (2024). Residual Learning Model-Based Classification of COVID-19 Using Chest Radiographs. Spectrum of engineering sciences, 2(3), 367-396.

[41] Raza, A., Soomro, M. H., Shahzad, I., & Batool, S. (2024). Abstractive Text Summarization for Urdu Language. Journal of Computing & Biomedical Informatics, 7(02).

[42] HUSSAIN, S., RAZA, A., MEERAN, M. T., IJAZ, H. M., & JAMALI, S. (2020). Domain Ontology Based Similarity and Analysis in Higher Education. IEEEP New Horizons Journal, 102(1), 11-16.

[43] Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.