# Analysis of Ransomware Impact on Android Systems using Machine Learning Techniques

Anfal Sayer M. Al-Ruwili[1], Ayman Mohamed Mostafa[2]

Department of Computer Science-College of Computer and Information Sciences, Jouf University, Saudi Arabia[1]
Department of Information Systems-College of Computer and Information Sciences, Jouf University, Saudi Arabia[2]
Department of Information Systems-College of Computers and Informatics, Zagazig University, Egypt[2]

*Abstract*—**Ransomware is a significant threat to Android systems. Traditional methods of detection and prediction have been used, but with the advancement of technology and artificial intelligence, new and innovative techniques have been developed. Machine learning (ML) algorithms are a branch of artificial intelligence that have several important advantages, including phishing detection, malware detection, and spam filtering. ML algorithms can also be used to detect ransomware by learning the patterns and behaviors associated with ransomware attacks. ML algorithms can be used to develop detection systems that are more effective than traditional signature-based methods. The selection of the dataset is a crucial step in developing an ML-based ransomware detection system. The dataset should be large, diverse, and representative of the real-world threats that the system will face. It should also include a variety of features that are informative for ransomware detection. This research presents a survey of ML algorithms for ransomware detection and prediction. The authors discuss the advantages of ML-based ransomware detection systems over traditional signature-based methods. They also discuss the importance of selecting a large, diverse, and representative dataset for training ML algorithms. Two datasets are applied during the conducted experiments, which are SEL and ransomware datasets. The experiments are repeated with different splitting ratios to identify the overall performance of each ML algorithm. The results of the paper are also compared to recent methods of ransomware detection and showed high performance of the proposed model.**

***Keywords**—Ransomware; machine learning; malware detection; phishing detection; spam filtering*

## I. Introduction

During today's rapidly evolving technological landscape, the menace of malware remains a formidable challenge, with ransomware at its forefront. Cybercriminals consistently innovate to breach computer systems, propelling the need for more advanced detection and prediction methods. Particularly, ransomware is a dire threat to Android systems, prompting the exploration of innovative strategies driven by the surge of artificial intelligence and machine learning (ML).

Ransomware is a pernicious form of malware that encrypts valuable data, demanding a ransom for decryption [1]. This cybersecurity threat spans servers, computers, and smartphones, jeopardizing critical personal data and daily operations [2]. Android systems, in particular, are a prime target due to their open-source nature, enabling attackers to encrypt and hold data hostage, thereby escalating the impact of ransomware attacks. The advent of cryptocurrencies like Bitcoin has further complicated tracking both the attackers and their extorted funds [3].

The integration of artificial intelligence particularly ML, has emerged as a potent tool in the fight against ransomware. ML algorithms, distinguished by their effectiveness in various domains, excel in detecting phishing attempts, identifying malware, and filtering spam [3]. These algorithms can be trained on extensive datasets containing benign and malicious software to discern the unique behavioral patterns that set ransomware apart. Once trained, they can recognize new ransomware variants, by analyzing these distinctive behavioral patterns. The advantages of using ML for ransomware detection over traditional methods are profound. ML algorithms can identify new, previously unknown ransomware variants, adapt to evolving threat patterns, and minimize false positives. This is achieved by focusing on behavior patterns rather than static signatures or predefined rules [4].

Ransomware behavior is evolving rapidly and it targets many important assets, including critical systems such as Android. Therefore, it poses a challenge to detect and prevent it, and automated learning methods are effective ways to detect malicious ransomware behavior.

This paper presents an analysis of the malicious behavior of ransomware targeting the Android system using several machine learning algorithms such as Naive bayes, Support vector machine, Decision tree, k-nearest neighbors, Random forest, and Logistic Regression. This was conducted on various data sets containing many of the most common types of ransomware, and the data was divided into different proportions to ensure the effectiveness of the ML models.

## II. Related Work

As presented in study [1], ransomware is malicious software that seizes a victim's data, encrypts data, and extorts money from the victim in exchange for their data. It evolves rapidly, making its detection challenging requiring continuously evolving detection tools. The authors of [2] categorized Ransomware into three main methods: Computer locker, I/O centric Locker, and Crypto miner. It follows a five-step process, including infection delivery, environment verification, hiding to avoid detection, target selection, and displaying a blackmail message to the victim.

As presented in study [4], the authors proposed an automated education-based approach for detecting ransomware through three key stages: data collection related to ransomware, extraction of shared ransomware behaviors, and precise identification of harmful ransomware behaviors. The authors also emphasized the effectiveness of automated education in evaluating and verifying information. Ransomware targets a wide range of categories, including individuals, due to the personal value of their data, business databases, and commercial companies. It also targets local servers to damage multiple systems potentially. Additionally, there are general guidelines for ransomware protection, such as encrypting backups, utilizing updated firewalls, using the latest antivirus software, and implementing a strategy of reduced user privileges [5].

As presented in study [6], ransomware has increased in recent years, primarily due to its profitability, and it has targeted various sectors, including healthcare, industry, and education. As explained in study [7], ransomware can be categorized into encryption-focused and screen-locking variants. The increase in ransomware attacks is attributed to its availability as a service, with some attackers offering ransomware creation tools and taking a 20% cut of the ransom. Victims facing such attacks have four options: pay the ransom, restore data from backups, and attempt to guess the decryption key through brute force, or lose the data.

As presented in [8], the methods for addressing ransomware are categorized into two key aspects: prevention, including measures like backups, and detection, further divided into four categories. These categories are behavior analysis of data to identify suspicious changes and trigger alarms and to compare current operations to past ransomware behaviors. Finally, event-based detection includes traffic and API monitoring and detection through automated learning algorithms. As shown in study [9], conventional intrusion detection systems fall short in countering advanced attacks, thus necessitating more sophisticated detection programs. It highlighted one advanced approach, the honey pot, which is a system purposefully crafted to emulate a genuine system luring in potential attackers. As presented in study [10], numerous mechanisms exist to safeguard against ransomware infections. These include maintaining up-to-date system updates, which address vulnerabilities with each release. Additionally, employing the latest ransomware detection tools is crucial.

As explained in study [11], several reasons contributed to the intensive increase of ransomware: Encryption algorithms are a double-edged sword used for privacy and attacks, and Electronic currencies that allow the attacker to be anonymous. With the rapid development of ransomware, it has become easy and available to obtain. As presented in study [12], a detection-assisting proposal called PEAD is based on the API of detecting the attack before encryption.

As presented in study [13], many victims find themselves compelled to pay the ransom. Email fraud is the primary method of victim targeting, accounting for 59%, followed by websites at 24%. Furthermore, it introduced a ransomware detection tool utilizing machine learning. This tool monitors CPU usage, detecting deviations from normal performance as indicators of potential ransomware activity. Additionally, it scrutinizes file extensions executed on the device, issuing warnings for suspicious programs. Notably, it successfully alerts users during an attack, displaying 0 for benign behavior and 1 for harmful behavior when detected. As presented in [14], ransomware poses a significant cybersecurity threat and is a prevalent form of malware in cybercrime. Numerous variants characterize ransomware and represent a criminal innovation primarily driven by monetary gains, often utilizing cryptocurrencies such as Bitcoin.

As presented in study [15], a strategy that relies on dynamic analysis of prevalent ransomware families, such as WannaCry, was devised. This strategy involves monitoring the real-time impact on a system, including adding or deleting files. Furthermore, it involves observing packet behavior in Wireshark; a change in the Multiplex ID indicates a potential infection.

As proposed in study [16], the researchers introduced a static analysis technique for identifying ransomware. It gathered executable code samples from various ransomware families, categorized them based on their characteristics and employed automated machine learning for classification. Ransomware detection can be approached through various primary methods. The first method is signature-based detection, which involves comparing malicious signatures with known ones. The second method is inferential disclosure, which relies on comparing malicious code [17]. The conducted experiments in virtual environments are used to assess ransomware detection techniques. It observed that these techniques exhibited improved performance after a 24-hour period. Dynamic analysis emerged as a more accurate method, while signature detection and inferential detection were found to be less effective in identifying new and mysterious malicious families.

As presented in study [18], ransomware typically leaves victims with limited recourse for addressing the attack, often necessitating a ransom payment. The study detailed an examination of various ransomware variants and established a virtual environment for scrutinizing DNS activity during such attacks. The researchers employed a trace capture tool both before and following the execution of the attack. As authors of [19, 20], the ransomware follows a specific lifecycle. It initiates by constructing a malicious program, followed by propagation, reaching the target device, identifying the data to encrypt, performing encryption or locking, and ultimately resorting to blackmail.

Ransomware employs various methods to infiltrate victims and compromise their critical data. One of the primary tactics involves encrypting the victim’s data to seize control and another method employs a lock screen approach. It’s essential to recognize that cybercriminals pursue their malicious objectives, such as extortion, sabotage, and financial gain. Understanding the consequences of ransomware and being knowledgeable about defenses against this threat can enhance user and asset security. Therefore, in this section we will conduct a comparative analysis of the studies, focusing on four main aspects: the ransomware methods employed in each paper, the objectives behind ransomware infection and its impact, and the countermeasures utilized.

TABLE I. COMPARATIVE ANALYSIS OF RANSOMWARE AND THEIR COUNTERMEASURES

| Ref | Ransomware Method | Objective | Ransomware Effect | Security Countermeasure |
|---|---|---|---|---|
| [1] | Ransomware in general | Profitability –Disruptive – blackmail. | Encryption of victims data | A model that analyzes the level of risk using inferential detection. |
| [2] | Crypto-Ransomware infection | Profitability –Disruptive – blackmail. | Encryption of victims data | Inferential Behavior & API Linking. |
| [4] | Crypto Ransomware | Profitability –Disruptive – blackmail. | Encryption of victims data | Automated learning algorithm for ransomware prevention. |
| [5] | Lock screen or encryption | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | A proposal that prevents the attack, slows the encryption and reduces the impact. |
| [6] | Cryptographic Ransomware | Profitability –Disruptive – blackmail. | Encryption of victims data | Survey of ransomware detection techniques. |
| [7] | Ransomware in general | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | Diagram with instructions for dealing with ransomware. |
| [8] | Crypto Ransomware Attack | .Profitability –Disruptive – blackmail | Encryption of victims data | Proposal for early detection. |
| [9] | Ransomware in general | .Profitability –Disruptive – blackmail | Data damage either by encryption or lock screen. | A three-layer proposal based on a honey pot. |
| [10] | Lock screen or encryption | .Profitability –Disruptive – blackmail | Data damage either by encryption or lock screen. | A proposal based on three-tiered security. |
| [11] | Ransomware in general | .Profitability –Disruptive – blackmail | Data damage either by encryption or lock screen | A proposal that uses machine learning algorithms. |
| [12] | Crypto-Ransomware | Profitability –Disruptive – blackmail. | Encryption of victims data | PEDA pre-encryption algorithm. |
| [13] | Ransomware in general | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | A proposal to identify benign or harmful behavior with an API-based proposal. |
| [14] | Bitcoin and the Financial Impact of Ransomware | Profitability –Disruptive – blackmail. | Financial impact awareness and vision that contributes to solutions against attack | NA |
| [15] | WannaCry Ransomware | .Profitability –Disruptive – blackmail | Encryption of victims data | Dynamic analysis to collect malware indicators |
| [16] | Lock screen or encryption | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | Static analysis based proposal. |
| [17] | Lock screen or encryption | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | Dynamic analysis and code comparison. |
| [18] | WannaCry Ransomware | Profitability –Disruptive – blackmail. | Encryption of victims data | Conducting an analysis only contributes to the manufacture of mechanism. |
| [19] | Lock screen or encryption | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | Analysis of detection mechanisms such as honey pot and dynamic analysis based on API programming. |
| [20] | Ransomware in general | Profitability –Disruptive – blackmail. | Data damage either by encryption or lock screen | A proposal based on machine learning that compares the neural network. |

In Table I, it proposes a comparative analysis of different ransomware methods by explaining the main objective of each method, its main effect and the proposed security countermeasure for preventing the threat.

## III. MACHINE LEARNING METHODS FOR RANSOMWARE APPLICATIONS

Machine learning (ML) is a powerful tool that can be used to detect ransomware applications. ML algorithms can be trained on large datasets of both benign and malicious software to learn the behavioral characteristics that distinguish ransomware from legitimate software. Once trained, these algorithms can be used to identify new and previously unseen variants of ransomware, including zero-day attacks, based on their behavioral patterns. ML-based ransomware detection has several advantages over traditional signature-based and heuristic-based detection methods. First, ML algorithms can detect new or unknown ransomware variants that do not match existing signatures or patterns. Second, ML algorithms can adapt to changing ransomware behavior patterns over time. Third, ML algorithms are less prone to false positives than signature-based and heuristic-based detection, as they rely on detecting actual behavior patterns rather than static code signatures or predefined rules [21].

Ransomware threatens the Android system significantly, as we have seen its impact in the previous part of this research. Many traditional methods have been applied to detect and predict this malicious attack. However, with the recent development of technology and artificial intelligence, modern and innovative methods have been developed to detect ransomware attacks. The application of machine learning algorithms is one of the branches of artificial intelligence that has several important advantages, including phishing detection, malware detection, and spams filtering and contributes to commercial tasks.

It can be classified into different categories. One such method is supervised machine learning, which uses algorithms that require outside supervision in order to provide a data set for testing and training. As a result, the model uses decision trees and support vector machines to enable categorization and prediction. The other form of algorithm is unsupervised education, where the algorithms produce data based on their models K-Means. Combining the first two forms, semi-supervised machine learning is the third type. The final type is reinforcement machine learning, in which the algorithm responds to good or bad signals by repeating the task performance [22].

There are numerous techniques to implement ML-based ransomware detection. Utilizing ML algorithms to search for suspicious activities in network traffic is a typical strategy. The usage of ML algorithms, for instance, can be utilized to spot ransomware attacks by spotting surges in encryption activity. Utilizing ML algorithms to examine the behavior of active processes is an alternative strategy. For instance, ML algorithms can be used to spot processes that attempt to connect to known ransomware servers or that encrypt a huge number of files. ML-based ransomware detection is a rapidly evolving field, and new techniques are being developed all the time. As ransomware attacks become more sophisticated, ML will continue to play an increasingly important role in ransomware detection and prevention [23].

## IV. Proposed Methodology

In this paper, ML algorithms can be used to detect ransomware by learning the patterns and behaviors associated with ransomware attacks. ML algorithms can be used to develop detection systems that are more effective than traditional signature-based methods. As illustrated in Fig. 1, the ransomware dataset is split into training and testing where the training dataset is preprocessed and then different ML algorithms are applied to measure accuracy. The testing dataset is then applied to identify the best accuracy for each splitting ratio.

The objective of using machine learning (ML) in ransomware detection is to develop systems that are more effective and efficient at detecting ransomware attacks than traditional signature-based methods. Ransomware attacks are becoming increasingly sophisticated and difficult to detect using traditional signature-based methods. ML algorithms can learn to identify the patterns and behaviors associated with ransomware attacks, and they can be used to develop detection systems that are able to detect new and emerging ransomware variants. ML-based ransomware detection systems can also be more efficient than traditional signature-based methods. Traditional signature-based methods require security vendors to maintain and update databases of signatures for known ransomware variants [24]. The overall methodology for managing ransomware attacks is presented in the following steps:

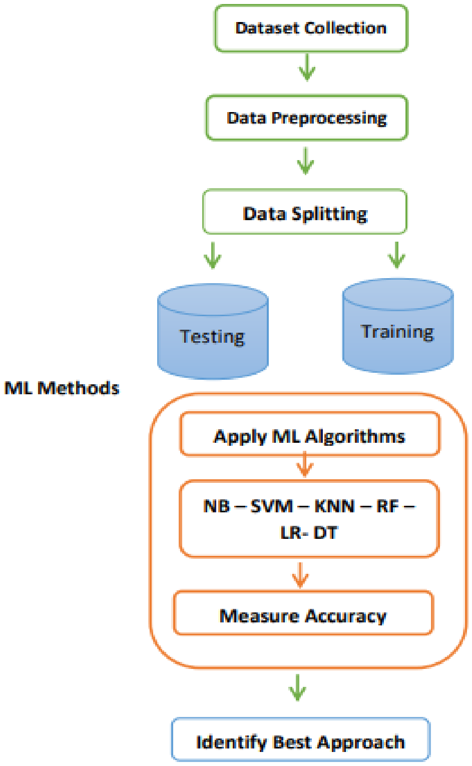

Fig. 1. Proposed ransomware methodology.

### A. Dataset Collection

The selection of the dataset is a crucial step in developing a machine learning (ML)-based ransomware detection system. The dataset should be large, diverse, and representative of the real-world threats that the system will face. It should also include a variety of features that are informative for ransomware detection.

*1) Security Engineering Lab (SEL) dataset:* Ransomware threatens the Android system significantly, Many traditional methods have been applied to detect and predict this malicious attack. However, with the recent development of technology and artificial intelligence, modern and innovative methods have been developed to detect ransomware attacks. Applying machine-learning algorithms is a major advantage for phishing, malware, and spam detection. The Security Engineering Lab (SEL) dataset built in [25] based on 10153 samples of Android apps was used to be one of the latest data sets related to Android ransomware and benign software. As presented in Table I, the dataset contains 500 ransomware applications from several sources, such as the Ransom Proper Project, a timeline that classifies ransomware based on 15 families. It also contains 9653 benign applications collected from several reliable sources, such as the official Android store Google Play , that are distributed as presented in Table II.

TABLE II. DISTRIBUTION OF SEL DATASET

| Type | Benign Programs | Ransomware Programs | Total |
|---|---|---|---|
| Number of Programs | 9653 | 500 | 10153 |

*2) Android ransomware dataset:* In this dataset, 10 types of the latest ransomware for Android Taken from Kaggle [26], such as Pletor, Sim blocker, Wanna Locker, Jisut, SV peng, Porn Droid, Koler, Ransom BO, Charger, and Locker pin. The dataset contains 392034 records of benign data for Android programs , that are distributed as presented in Table III.

TABLE III. DISTRIBUTION OF RANSOMWARE DATASET

| Attack Name | Number of Records |
|---|---|
| SVpeng | 54161 |
| PornDroid | 46082 |
| Koler | 44555 |
| Benign | 43091 |
| RansomBO | 39859 |
| Charger | 39551 |
| Simplocker | 36340 |
| WannaLocker | 32701 |
| Jisut | 25672 |
| Lockerpin | 25307 |
| Pletor | 4715 |

## B. Data Preprocessing

To analyze machine learning algorithms more accurately and reliably and to make sure there are no duplicate, wrong, or corrupted data in the dataset, it must be cleaned. This could lower the level of analysis and produce results that are inaccurate. Therefore, the cleaning procedure raises the data's quality. As can be seen in Table IV, methods have thus been used to enhance the data-cleansing process.

TABLE IV. DATA CLEANING OF SEL AND RANSOMWARE DATASETS

| Process | SEL Dataset | Ransomware Dataset |
|---|---|---|
| Data Duplication | The dataset is of high quality and does not contain duplicate data | The dataset is of high quality and does not contain duplicate data |
| Unnecessary Data | - | Unnecessary data are eliminated |
| Missing Values | Removing missing values from the columns | - |
| Validation | The data has been validated | The data has been validated |
| Data Conversion | - | The dataset is converted into binary classification |

## C. Selection of ML Algorithms

This paper explains varieties of ML algorithms that can be used for ransomware detection as follows:

*1) Naïve Bayes (NB):* Ransomware detection is one of the many classification tasks that can be performed using the straightforward yet effective machine learning method known as Naive Bayes (NB). Based on the likelihood that each feature in a given data point belongs to a particular class, NB determines the likelihood that a given data point belongs to that class [24].

*2) Support Vector Machine (SVM):* Another effective machine learning approach for ransomware detection is SVM. Finding a hyperplane in the feature space that divides the data points into two classes (harmless and malevolent) is how SVMs operate. You would first need to compile a dataset containing both benign and dangerous software in order to employ SVMs for ransomware detection [24]. The file type, file size, file permissions, and file contents should all be included in this dataset. The model can be used to categorize fresh data points as benign or malicious after it has been trained. The model would determine the distance between the new data point and the hyperplane in order to accomplish this. The data are only used if the distance exceeds a predetermined threshold.

*3) Decision Tree (DT):* A supervised machine learning approach called decision trees (DT) is useful for both classification and regression tasks. DTs divide the feature space recursively into smaller and smaller sections until each zone only contains data points from one class [22]. After the dataset has been gathered, a DT model would need to be trained using the information. In order to achieve this, the feature space is recursively divided into smaller and smaller sections, until each zone only contains data points from a single class.

*4) K-nearest neighbors (KNN):* It is a simple and effective machine-learning algorithm that can be used for classification and regression tasks. KNN works by finding the K most similar data points to the new data point and assigning the new data point to the class of the majority of the K most similar data points [21].

*5) Random Forest (RF):* An ensemble learning approach called random forests (RFs) combines the predictions of various decision trees to create a forecast that is more accurate [22]. A huge number of decision trees are built using random selections of the data using RFs, and their predictions are then averaged. After gathering your dataset, you would need to use the information to train an RF model. To do this, many decision trees are built using random subsets of the data, and their predictions are then averaged.

*6) Logistic Regression (LR):* A machine learning approach for classification tasks is logistic regression (LR). By applying a logistic function to the data, LR operates [22]. A real integer is entered into the logistic function, a sigmoid function that returns a probability between 0 and 1. After gathering your dataset, you would need to use the information to train an LR model. In order for the model to forecast the likelihood that a data point would be malicious given its characteristics, the data must be fitted using a logistic function.

## V. Experimental Results

The experimental results on two ransomware datasets are promising and suggest that ML algorithms can be used to develop effective ransomware detection systems. The accuracy, precision, recall, and F1-score are measured in both SEL and Ransomware datasets to explore the main performance of each dataset with different ML algorithms.

The accuracy is used to measure the performance of a ML model in predicting the correct class $TP_c$ of a new data point. It is defined as the percentage of correct predictions made by the model as given in Formula (1).

$$Accuracy = \frac{TP_c+TN_c}{TP_c+FP_c+TN_c+FN_c} \quad (1)$$

A measure of precision in ML is the percentage of positive predictions that are in fact accurate. It is calculated by dividing the total number of accurate positive forecasts by the number of true positives as given in Formula (2).

$$Precision = \frac{TP_c}{TP_c+FP_c} \quad (2)$$

Recall is a metric used in ML that evaluates the percentage of real positives that are properly expected. It is calculated by dividing the total number of real positives by the number of true positives as given in Formula (3).

$$Recall = \frac{TP_c}{TP_c+FN_c} \quad (3)$$

The precision and recall measures for ML are combined into a single statistic called the F1-score. Given that it is defined as the harmonic mean of the precision and recall scores, both precision and recall are given equal weight as given in Formula (4).

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

### A. Results of SEL Dataset

The experimental results are executed for training and testing the dataset using the predefined ML algorithms. The dataset is split based on two ratios 80:20 and 70:30. For the dataset split ratio of 80:20, the 80% is applied for training and the 20% is applied for testing. For the dataset split ratio of 70:30, the 70% is applied for training and the 30% is applied for testing. As presented in Table V the data is split to 20% for testing 80% for training and the performance of different machine learning is measured based on the accuracy, recall, and F1-score.

TABLE V. Performance Analysis of ML Algorithms with 80:20 Splitting for SEL Dataset

| ML Alg. | Accuracy | Recall | F1-score |
|---|---|---|---|
| SVM | 99.26% | 99.43% | 99.61% |
| RF | 97.68% | 99.89% | 98.79% |
| NB | 95.02% | **100%** | 97.44% |
| KNN | 99.06% | 99.48% | 99.50% |
| DT | 97.48% | 97.97% | 98.66% |
| LR | **99.31%** | 99.63% | **99.63%** |

As presented in Fig. 2, the LR algorithm achieved the highest accuracy of 99.31% on the SEL dataset, while the SVM algorithm recorded 99.26%. The KNN algorithm recorded the third-highest accuracy, with 99.06%. It is followed by the RF algorithm with an accuracy of 97.68%, while the DT algorithm recorded an accuracy of 97.48%. The NB algorithm achieved 95.02%, which is considered the lowest accuracy on the SEL dataset.

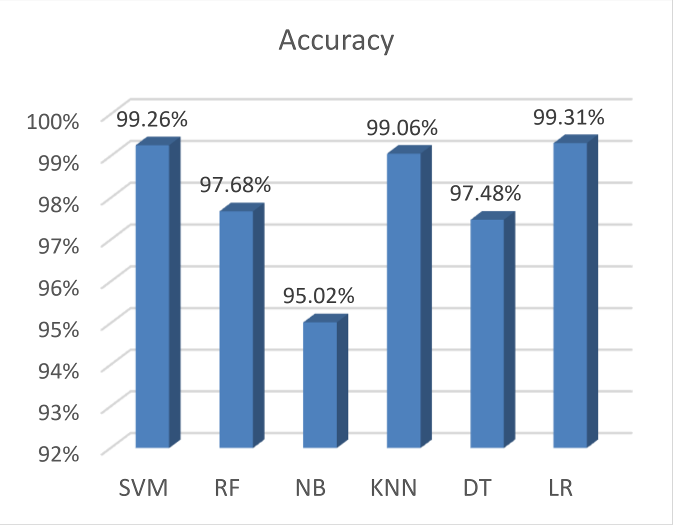

Fig. 2. Accuracy of different ML algorithms on SEL dataset with splitting ratio of 80:20.

As presented in Fig. 3, the NB algorithm achieved the highest recall of 100% on the SEL dataset, while the RF algorithm recorded 99.89%. The LR algorithm recorded 99.63%. It is followed by the KNN algorithm with a recall of 99.48%. The SVM algorithm recorded a recall of 99.43%, while the DT algorithm achieved 97.97%, which is considered the lowest recall on the SEL dataset.

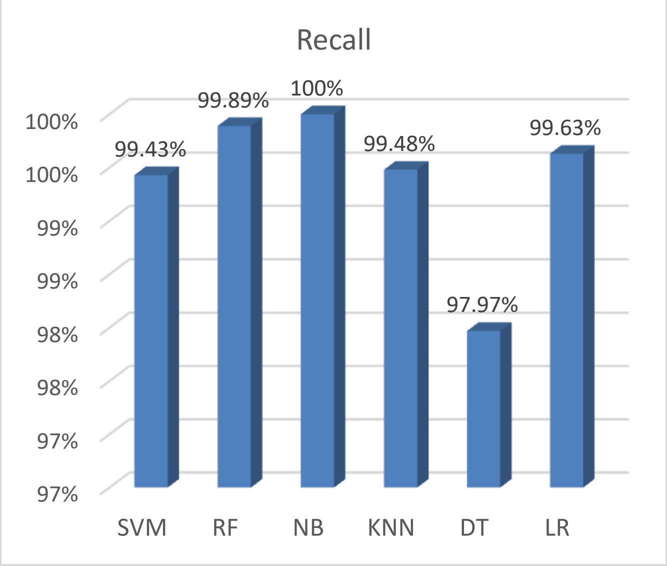

Fig. 3. Recall of different ML algorithms on SEL dataset with splitting ratio of 80:20.

As presented in Fig. 4, the LR algorithm achieved the highest F1-score of 99.63% on the SEL dataset, while SVM recorded 99.61%. The KNN algorithm recorded 99.50%, while the RF algorithm achieved an F1-score of 98.79%. The DT algorithm recorded an F1-score of 98.66%, while the NB algorithm achieved 97.44%, which is considered the lowest F1-score on the SEL dataset.

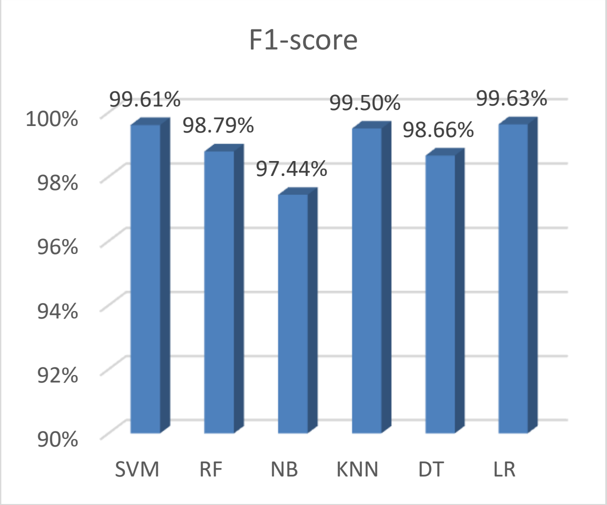

Fig. 4. F1-score of different ML algorithms on SEL dataset with splitting ratio of 80:20.

As presented in Table VI, the data is split to 30% for testing 70% for training and the performance of different machine learning is measured based on the accuracy, recall, and F1-score.

TABLE VI. PERFORMANCE ANALYSIS OF ML ALGORITHMS WITH 70:30 SPLITTING FOR SEL DATASET

| ML Alg. | Accuracy | Recall | F1-score |
|---|---|---|---|
| SVM | 99.24% | 99.41% | 99.60% |
| RF | 98.32% | 99.89% | 99.12% |
| NB | 95.27% | **99.96%** | 97.5% |
| KNN | 99.31% | 99.65% | 99.63% |
| DT | 93.30% | 93.17% | 96.36% |
| LR | **99.47%** | 99.72% | **99.72%** |

As presented in Fig. 5, the LR algorithm achieved the highest accuracy of 99.47% on the SEL dataset, while the KNN algorithm recorded 99.31%. The SVM algorithm recorded the third-highest accuracy, with 99.24%. It is followed by the RF algorithm with an accuracy of 98.32%, while the NB algorithm recorded an accuracy of 95.27%. The DT algorithm achieved 93.30%, which is considered the lowest accuracy on the SEL dataset.

As presented in Fig. 6, the NB algorithm achieved the highest recall of 99.96% on the SEL dataset, while the RF algorithm recorded 99.89%. The LR algorithm recorded 99.72%. It is followed by the KNN algorithm with a recall of 99.65%. The SVM algorithm recorded a recall of 99.41%, while the DT algorithm achieved 93.17%, which is considered the lowest recall on the SEL dataset.

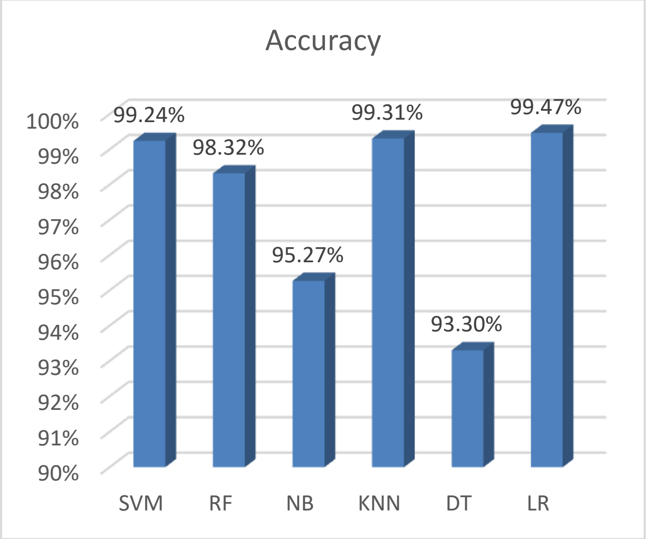

Fig. 5. Accuracy of different ML algorithms on SEL dataset with splitting ratio of 70:30.

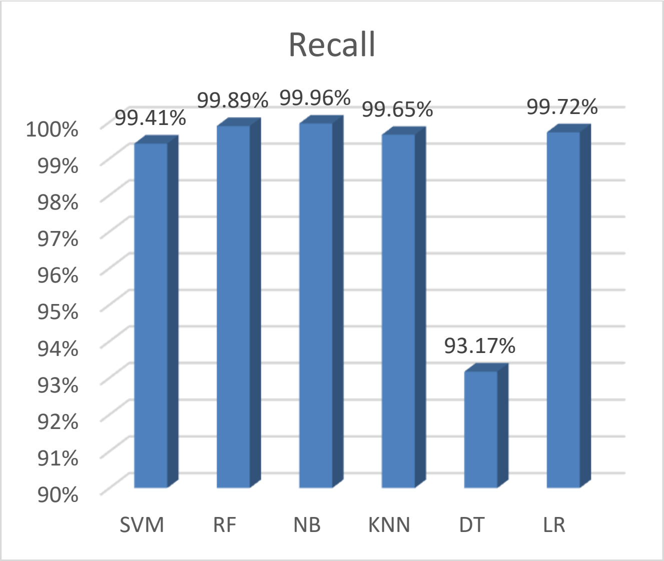

Fig. 6. Recall of different ML algorithms on SEL dataset with splitting ratio of 70:30.

As presented in Fig. 7, the LR algorithm achieved the highest F1-score of 99.72% on the SEL dataset, while KNN recorded 99.63%. The SVM algorithm recorded 99.60%, while the RF algorithm achieved an F1-score of 99.12%. The NB algorithm recorded an F1-score of 97.50%, while the DT algorithm achieved 96.36%, which is considered the lowest F1-score on the SEL dataset.

### *B. Results of Ransomware Dataset*

The experimental results are conducted again on the Ransomware dataset to reevaluate the overall performance of the detection methodology. The Ransomware dataset is also split to 80:20 and 70:30 ratios to check whether the performance will be enhanced or not. As presented in Table VII, the data is split to 20% for testing 80% for training and the performance of different machine learning is measured based on the accuracy, recall, and F1-score.

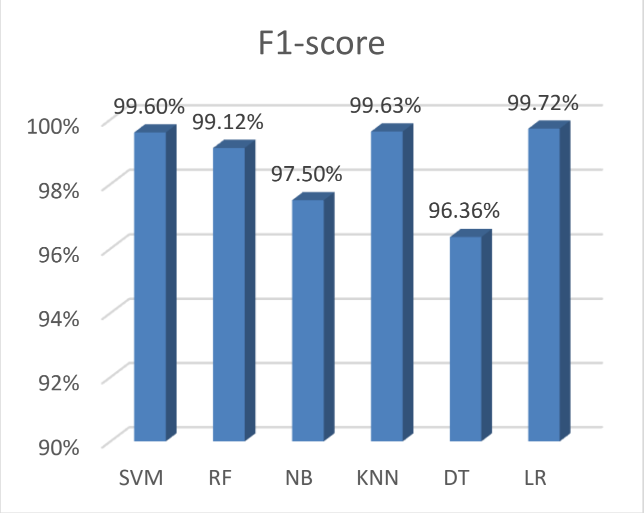

Fig. 7. F1-score of different ML algorithms on SEL dataset with splitting ratio of 70:30.

TABLE VII. PERFORMANCE ANALYSIS OF ML ALGORITHMS WITH 80:20 SPLITTING FOR RANSOMWARE DATASET

| ML Alg. | Accuracy | Recall | F1-score |
|---|---|---|---|
| RF | 90% | 96.4% | 94.7% |
| NB | 89.01% | **100%** | 94.18% |
| KNN | **93.25%** | 97.12% | **96.24%** |
| DT | 79.31% | 84% | 88% |
| LR | 89.49% | 96.9% | 94.2% |

As presented in Fig. 8, the KNN algorithm achieved the highest accuracy of 93.25% on the Ransomware dataset, while the RF algorithm recorded 90.00%. The LR algorithm recorded the third-highest accuracy, with 89.49%. It is followed by the NB algorithm with an accuracy of 89.01%. The DT algorithm achieved 79.31%, which is considered the lowest accuracy on the Ransomware dataset.

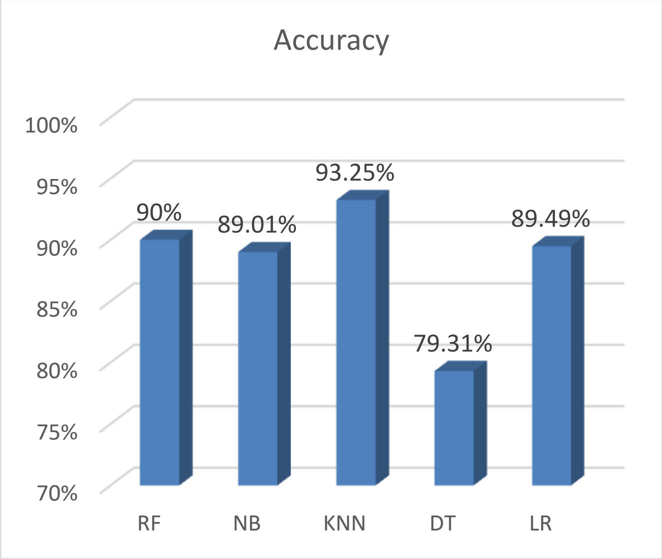

Fig. 8. Accuracy of ML algorithms on Ransomware dataset with splitting ratio of 80:20.

As presented in Fig. 9, the NB algorithm achieved the highest recall of 100% on the Ransomware dataset, while the KNN algorithm recorded 97.12%. The LR algorithm recorded the third-highest recall, with 96.90%. It is followed by the RF algorithm with a recall of 96.40%. The DT algorithm achieved 84%, which is considered the lowest recall on the Ransomware dataset.

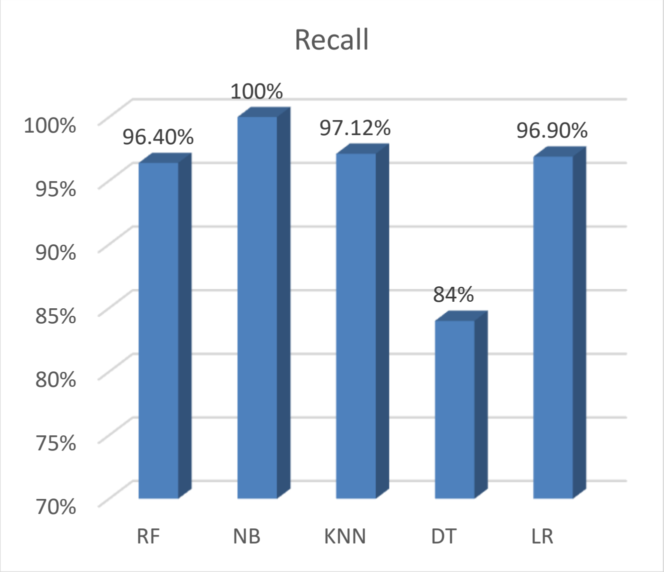

Fig. 9. Recall of ML algorithms on Ransomware dataset with splitting ratio of 80:20.

As presented in Fig. 10, the KNN algorithm achieved the highest F1-score of 96.24% on the Ransomware dataset, while the RF algorithm recorded 94.70%. The LR algorithm recorded the third-highest F1-score, with 94.20%. It is followed by the NB algorithm with an F1-score of 94.18%. The DT algorithm achieved 88%, which is considered the lowest F1-score on the Ransomware dataset.

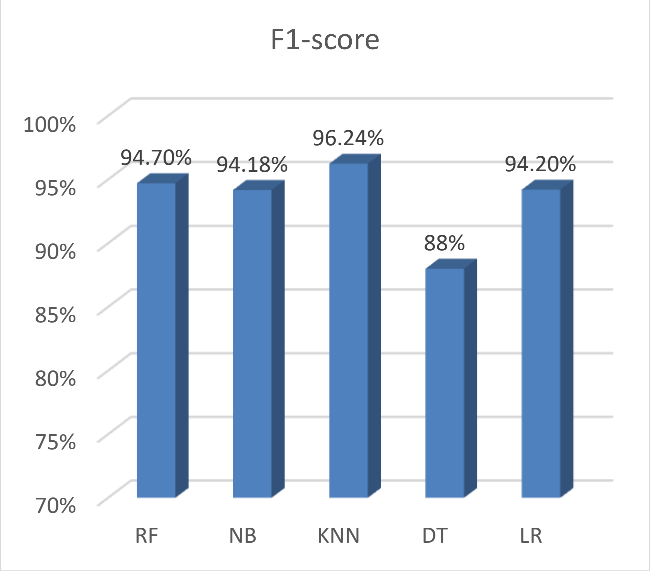

Fig. 10. F1-score of ML algorithms on Ransomware dataset with splitting ratio of 80:20.

TABLE VIII. PERFORMANCE ANALYSIS OF ML ALGORITHMS WITH 70:30 SPLITTING FOR RANSOMWARE DATASET

| ML Alg. | Accuracy | Recall | F1-score |
|---|---|---|---|
| RF | 89.32% | 97.77 % | 94.21% |
| NB | 88.96% | **99.97%** | 94.15% |
| KNN | **93.12%** | 97.11% | **96.16%** |
| DT | 82.88% | 87.79% | 90.12% |
| LR | 89.45% | 96.95% | 94.24% |

As presented in Table VIII, the data is split to 30% for testing 70% for training and the performance of different machine learning is measured based on the accuracy, recall, and F1-score.

As presented in Fig. 11, the KNN algorithm achieved the highest accuracy of 93.12% on the Ransomware dataset, while the LR algorithm recorded 89.45%. The RF algorithm recorded the third-highest accuracy, with 89.32%. It is followed by the NB algorithm with an accuracy of 88.96%. The DT algorithm achieved 82.88%, which is considered the lowest accuracy on the Ransomware dataset.

As presented in Fig. 12, the NB algorithm achieved the highest recall of 99.97% on the Ransomware dataset, while the RF algorithm recorded 97.77%. The KNN algorithm recorded the third-highest recall, with 97.11%. It is followed by the LR algorithm with a recall of 96.95%. The DT algorithm achieved 87.79%, which is considered the lowest recall on the Ransomware dataset.

As presented in Fig. 13, the KNN algorithm achieved the highest F1-score of 96.16% on the Ransomware dataset, while the LR algorithm recorded 94.24%. The RF algorithm recorded the third-highest F1-score, with 94.21%. It is followed by the NB algorithm with an F1-score of 94.15%. The DT algorithm achieved 90.12%, which is considered the lowest F1-score on the Ransomware dataset.

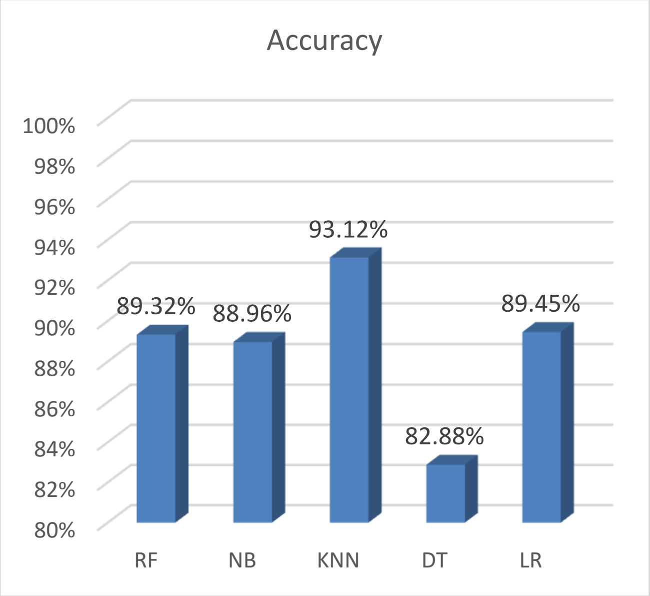

Fig. 11. Accuracy of ML algorithms on Ransomware dataset with splitting ratio of 70:30.

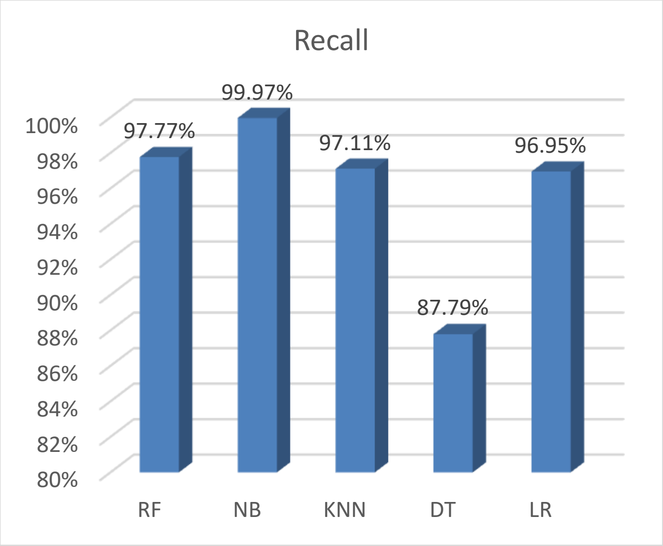

Fig. 12. Recall of ML algorithms on Ransomware dataset with splitting ratio of 70:30.

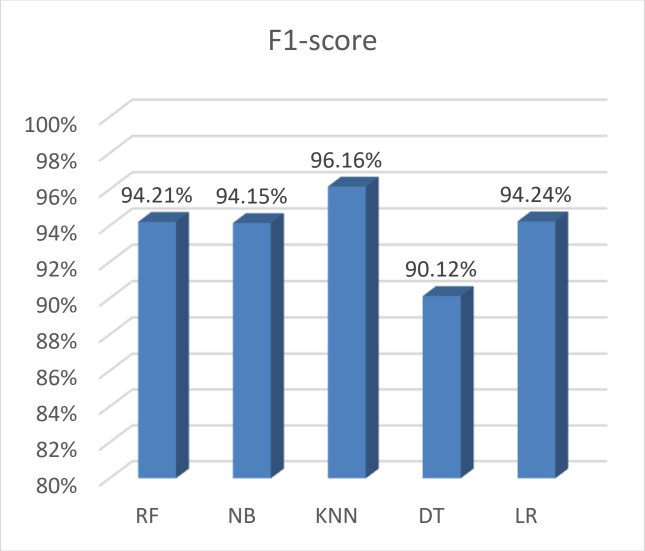

Fig. 13. F1-score of ML algorithms on Ransomware dataset with splitting ratio of 70:30.

## VI. DISCUSSION

LR achieved the highest accuracy rate of 99.47%, and accuracy is the classifier's overall accuracy in correctly classifying samples as either ransomware or benign software. But if we were to verify the accuracy of the classifier in classifying the positive samples that are actually ransomware, we can see that Recall achieved the highest percentage of NB in the SEL dataset, at 100% in dividing the data 80:20 and 99.96% in dividing the data 70:30. It also achieved the highest accuracy rate in the KNN ransomware data set, at 93.25%, which is the overall accuracy of the classifier for classifying ransomware and benign samples. It is also worth mentioning that NB also achieved the highest Recall rate in the second data set at 100%. Therefore, NB is considered a good model to verify the accuracy of the classifier in detecting samples that are actually ransomware.

TABLE IX. COMPARISON OF RECENT ML ALGORITHMS ACCURACY WITH THE PROPOSED MODEL

| Ref | ML Algorithm | Dataset | Accuracy |
|---|---|---|---|
| **[4]** | accuracy of classification algorithms | Dataset of 10 ransomware types | Not stated |
| **[11]** | regression and rule- based algorithms | Not stated | Not stated |
| **[12]** | Pre-Encryption Detection(PED) ,Random Forest (RF),Naïve Bayes (NB) and Ensemble Algorithms | dataset RISS containing 10 Ransomware types , 942 benign. | 98.44% |
| **[20]** | decision tree classifier, random forest classifier, naïve bayes classifier, logistic regression classifier | Dataset containing 70% ransomware | 99% |
| **[27]** | 1-NN ,3-NN, 5-NN,MLP , synthetic minority oversampling technique (SMOTE),NB,RF, | Dataset of 500 ransomware , 9653 benign | 97% |
| **[28]** | Logistic Regression(LR), Support Vector Machine (SVM), Neural Network | Dataset containing 2721 Ransomware , 2000 benign | 99.59% |
| **[29]** | Random Forest (RF) ,support vector machine (SVM), Naïve Bayes (NB) , Decision trees (J48) | Android benign dataset, Android ransomware datasets 2959,500 Simples | 97% |
| **[30]** | Support Vector Machine (SVM) , Decision Tree (DT) , Random Forest (RF) , Logistic Regression (LR) | dataset locker-ransomware simples containing 664 15751 benign | 99.98% |
| **Proposed Model** | NB , RF, LR , KNN , SVM , DT | RDA1(500Ransomware)<br>(Contains 9653 benign). RDA1(10 types ransomware ) (43091 Records for Benign ) | **99.47%** |
| | | Ransomware Dataset with 392034 records | **99.47%** |

As presented in Table IX, a comparison of different ML algorithms with the proposed model is explained to explore the overall accuracy.

## VII. CONCLUSION

Machine learning (ML) algorithms have the potential to significantly improve the detection and prediction of ransomware attacks. ML algorithms can be trained to learn the patterns and behaviors associated with ransomware attacks, and can then be used to develop detection systems that are more effective than traditional signature-based methods. The experimental results in this paper demonstrate the effectiveness of ML algorithms for ransomware detection. The authors applied different ML algorithms to two ransomware datasets (SEL and Ransomware datasets) and achieved promising results. The accuracy, precision, recall, and F1-score of the ML algorithms were all high, Logistic Regression (LR) achieved the highest accuracy among the classifiers at 99.47%, and (LR) also achieved the highest F1-score at 99.72%. Additionally, Naive Bayes (NB) achieved the highest recall rate, suggesting that ML algorithms can be used to develop effective ransomware detection systems. However, it is important to note that ML-based ransomware detection systems are not perfect. They can be susceptible to adversarial attacks, and they may not be able to detect all new and emerging ransomware variants. Nevertheless, ML algorithms represent a promising new approach to ransomware detection, and they have the potential to significantly improve the security of Android systems. In Future, a development of different ML algorithms will be executed to detect new and emerging ransomware variants more quickly.

## REFERENCES

[1] Aldaraani, N., & Begum, Z. (2018, April). Understanding the impact of ransomware: a survey on its evolution, mitigation and prevention techniques. In *2018 21st Saudi Computer Society National Computer Conference* (*NCC*) (pp. 1-5). IEEE.

[2] Olaimat, M. N., Maarof, M. A., & Al-rimy, B. A. S. (2021, January). Ransomware anti-analysis and evasion techniques: A survey and research directions. In *2021 3rd international cyber resilience conference* (*CRC*) (pp. 1-6). IEEE.

[3] Almohaini , R., Almomani, I., & AlKhayer , A. (2021). Hybrid-based analysis impact on ransomware detection for Android systems. *Applied Sciences*, *11*(22), 10976.

[4] Abraham, J. A., & George, S. M. (2019, July). A survey on preventing crypto ransomware using machine learning. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies* (*ICICICT*) (Vol. 1, pp. 259-263). IEEE.

[5] Patel, A., & Tailor, J. (2020). A malicious activity monitoring mechanism to detect and prevent ransomware. *Computer Fraud & Security, 2020*(1), 14-19.

[6] Berrueta, E., Morato, D., Magaña, E., & Izal, M. (2019). A survey on detection techniques for cryptographic ransomware. *IEEE Access*, *7*, 144925-144944.

[7] Maurya, A. K., Kumar, N., Agrawal, A., & Khan, R. A. (2018). Ransomware: evolution, target and safety measures. *International Journal of Computer Sciences and Engineering*, *6*(1), 80-85.

[8] Alqahtani, A., & Sheldon, F. T. (2022). A survey of crypto ransomware attack detection methodologies: an evolving outlook. *Sensors*, *22*(5), 1837.

[9] El-Kosairy, A., & Azer, M. A. (2018, April). Intrusion and ransomware detection system. In *2018 1st International Conference on Computer Applications & Information Security* (*ICCAIS*) (pp. 1-7). IEEE.

[10] Ren, A., Liang, C., Hyug, I., Broh, S., & Jhanjhi, N. Z. (2020). A three-level ransomware detection and prevention mechanism.*EAI Endorsed Transactions on Energy Web*, *7*(26).

[11] Kok, S., Abdullah, A., Jhanjhi, N., & Supramaniam, M. (2019). Ransomware, threat and detection techniques: A review. *Int. J. Comput. Sci. Netw. Secur*,*19*(2), 136.

[12] Kok, S. H., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Prevention of crypto-ransomware using a pre-encryption detection algorithm.*Computers 8*(4), 79.

[13] Arabo, A., Dijoux, R., Poulain, T., & Chevalier, G. (2020). Detecting ransomware using process behavior analysis.*Procedia Computer Science*,*168*, 289-296.

[14] Paquet-Clouston, M., Haslhofer, B., & Dupont, B. (2019). Ransomware payments in the bitcoin ecosystem.*Journal of Cybersecurity*,*5*(1), tyz003

[15] Kao, D. Y., & Hsiao, S. C.(2018, February). The dynamic analysis of WannaCry ransomware. In *2018 20th International conference on advanced communication technology* (*ICACT*)(pp. 159-166). IEEE.

[16] Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., & Sangaiah, A. K.(2019). Classification of ransomware families with machine learning based onN-gram of opcodes.*Future Generation Computer Systems*,*90*, 211-221.

[17] Sechel, S.(2019). A comparative assessment of obfuscated ransomware detection methods.*Informatica Economica*,*23*(2), 45-62.

[18] Akbanov, M., Vassilakis, V. G., & Logothetis, M. D. (2019). WannaCry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms. *Journal of Telecommunications and Information Technology*,(1), 113-124.

[19] Silva, J. A. H., López, L. I. B., Caraguay, Á. L. V., & Hernández-Álvarez, M.(2019). A survey on situational awareness of ransomware attacks—detection and prevention parameters.*Remote Sensing*,*11*(10).

[20] Masum, M., Faruk, M. J. H., Shahriar, H., Qian, K., Lo, D., & Adnan, M. I.(2022, January). Ransomware classification and detection with machine learning algorithms. In*2022 IEEE 12th Annual Computing and Communication Workshop and Conference* (*CCWC*)(pp. 0316-0322). IEEE.

[21] M. Masum, M. J. Hossain Faruk, H. Shahriar, K. Qian, D. Lo and M. I. Adnan, "Ransomware Classification and Detection With Machine Learning Algorithms," *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2022, pp. 0316-0322, doi: 10.1109/CCWC54503.2022.9720869.

[22] Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research* (*IJSR*).*[Internet]*, *9*(1), 381-386.

[23] A. Alraizza, A. Algarni, “Ransomware Detection Using Machine Learning: A Survey,” *Big Data and Cognitive Computing*, vol. 7, no. 3, pp. 143, 2023 https://doi.org/10.3390/bdcc7030143.

[24] G. Usha, P. Madhavan, M. Vimal Cruz, N. A. S. Vinoth, Veena and M. Nancy, "Enhanced Ransomware Detection Techniques using Machine Learning Algorithms," *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, Chennai, India, 2021, pp. 52-58, doi: 10.1109/ICCCT53315.2021.9711906.

[25] Dataset1 :https://sel.psu.edu.sa/Research/datasets/2020_RansIm-DS.php

[26] Dataset2 :https://www.kaggle.com/datasets/subhajournal/android-ransomware-detection

[27] Almomani, I., Qaddoura, R., Habib, M., Alsoghyer, S., Al Khayer, A., Aljarah, I., & Faris, H. (2021). Android ransomware detection based on a hybrid evolutionary approach in the context of highly imbalanced data. *IEEE Access*, *9*, 57674-57691.

[28] Sharma, S., Krishna, C. R., & Kumar, R. (2020, November). Android ransomware detection using machine learning techniques: A comparative analysis on GPU and CPU. In *2020 21st International Arab Conference on Information Technology (ACIT)* (pp. 1-6). IEEE.

[29] Alsoghyer, S., & Almomani, I. (2019). Ransomware detection system for Android applications. *Electronics*, *8*(8), 868.

[30] Su, D., Liu, J., Wang, X., & Wang, W. (2018). Detecting Android locker-ransomware on chinese social networks. *IEEE Access*, *7*, 20381-20393.