International Journal of Basic and Applied Sciences, 14 (SI-1) (2025) 594-605

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/zk0xqn54
Research paper



Ransomware Detection from Abnormal Behavior Traffic Using The Ensemble Model

S. Gomathi [1]*, K. Anithakumari [2]

[1] Assistant Professor/CSE, Dr. N.G.P. Institute of Technology, Coimbatore, India
[2] Associate Professor/IT, PSG College of Technology, Coimbatore, Tamil Nadu, India
**Corresponding author E-mail: mail2mathi86@gmail.com*

Received: July 15, 2025, Accepted: July 24, 2025, Published: November 1, 2025

Abstract

Ransomware is a grave online security menace to both personal and business data and information. Computer resource owners can be affected by authentication and privacy breaches, as well as financial damage and reputational damage, in the event of a Ransomware attack. However, the majority of machine learning-based ransomware detector studies are limited by malware obscurity, a lack of a proper analysis ecosystem, incorrect models, and low false-positive rates. To address these issues, this paper proposes Ransomware Detection through Voting and Learning (RDVL) as an ensemble-based approach for ransomware detection. We retrieved the ransomware data from the Kaggle repository in the first place. Then, the collected dataset is normalized with the assistance of L1-Norm Maximization and a Principal Component Analysis (PCA), and the most appropriate ransomware attributes are chosen. Finally, to categorize Ransomware as a subset of the abnormal traffic, RDVL-based ensemble methods are used, including bagging, boosting, and voting. The research aim will be to identify the first instance of a malicious notification of network traffic and its position within a ransomware procedure.
It also enables the identification of the activity of Ransomware at an early stage before it is influential. This research uses ransomware abnormal traffic data to complete all the experiments on our proposed framework. As per the outcome of the experiment, the proposed classifier proves to be stronger than other techniques in terms of accuracy scores and sensitivity scores.

Keywords: *Ransomware; Voting Ensemble Learning; Machine Learning; Network Traffic Analysis; Preprocessing; Feature Selection.*

1. Introduction

Day by day, internet usage is increasing due to the growth of new inventions. Inventions made possible by the Internet enable people to access more information and acquire knowledge and skills. In this modern world, the development of the Internet has become vast, and at the same time, people often lose valuable information and money. A cyberattack is one of the techniques used in the online world to steal our data. Cybercriminals design malicious software to attack systems and steal valuable information. Systems can be infected or hacked through various methods. Some of the most typical online hazards include the categories listed below. The term "hacking" refers to attempting to break into computer systems or private networks to access their data without proper authentication [1]. When a malicious SMS message is used to get logins or other confidential information, this is known as "smishing." Phishing is an email scam that targets people's personal information. Malicious software that enables attackers to take over control of files and hold them hostage for a price, like Ransomware. Websites are frequently the targets of Distributed Denial of Service (DDoS) attacks, which are often followed by extortion attempts.
Nowadays, cybercrime is becoming increasingly popular, because of which people are aware of various threats and attacks that have been prevalent in recent days [2]. Malicious hackers know how to exploit human or security vulnerabilities to steal identities, information, or revenue. Though the users are cautious, in our normal web activities, such as when an individual interacts with suspicious URLs and accesses a website that is infected, their information is exposed to threats. Installing software from a commercial source without reading the license agreements. Downloadable software includes music players, toolbars, games, and other system utilities. Music, files, and pictures can be shared with other users on a network when a spam email, text message, or email attachment is opened [3]. Infected computers utilize asymmetric encryption. A collection of keys is used in this type of encryption to both encrypt and decrypt a file. A random public-private key pair is produced by the attackers; these key pairs are used for the decryption of data on the target's server using the private key. Without the private key, it is not possible to obtain access to the information that is being locked in captivity. There are many different formats for Ransomware. Ransomware (and other malware) often spreads through targeted attacks and email spam [4].
Three categories—signature-based, behavioral-based, and anomaly-based—are used to categorize malware detection. Generally, manually performed anomaly detection was widely used. Machine learning approaches, on the other hand, are improving the reliability of anomaly detectors. Of course, there are start-up costs associated with machine learning, such as data requirements and engineering talent. Machine learning antimalware software cannot be client-driven since a desktop PC or mobile device is exposed to far fewer, more restricted samples

Copyright © S. Gomathi, K. Anithakumari. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

of malware [5-6]. Because of the lower costs and increased availability of cloud servers, machine learning malware analysis is within reach like never before, be it in business or at home. Big data processing and cloud-based services are necessary to implement machine learning. To overcome these shortcomings, scholars have resorted to machine learning (ML) methods that can detect complex trends in large data sets and categorize ransomware attacks in accordance with their actions [7-8]. Deep learning models are among the most promising methods that have already proven to analyze abnormal traffic patterns, detect encrypted data, and other signs that ransomware activity occurred. The Ensemble learning techniques, which are approaches that merge the errors of several machine learning models to produce more resilient and accurate forecasts, have likewise been found to be very promising in the increment of the detection performance [9-10]. Ensemble approaches can be used to alleviate the shortcomings of single models, especially in the case of complex and dynamic attacks. The Ransomware Detection via Voting and Learning (RDVL) method is the only novel technique examined in the paper, which suggests identifying Ransomware through the patterns of anomalies in traffic with the help of ensemble learning algorithms. Ransomware is classified and detected by a deep learning system known as the RDVL, which concurrently integrates two or more models to do so in an online manner. The present ensemble design is to incorporate such types of algorithms to provide superior reading and additional flexibility in being able to predict. The RDVL can identify anomalous activity patterns in connection with ransomware attacks because of the attention it pays to network traffic-related features of Ransomware, such as encrypted file transfer, abnormally high data access frequency, and network out-of-band attempts.

The RDVL is necessitated by the need to enter real-time ransomware detection, in which it relies on votes made by various machine learning classifiers to form its decision. It is these ransomware-like network traffic volume patterns that the RDVL method can exploit to recognize new forms and variants of Ransomware. The voting mechanism ensures that the final decision will share the merits of many classifiers, and will rule out the chances of a significant false positive rate to maximize the detection reliability. One of the methods is the RDVL method, and it is developed in this field as an addition to other techniques and procedures, such as ensemble learning and deep learning techniques, to recognize the pathognomonic traffic pattern of ransomware attacks.

2. Literature Survey

A few ML-based approaches to malware detection, which apply to computer systems, were evaluated using a simple yet reliable machine learning based malware detection system [6]. Smartphones are becoming a common target of ransomware attacks on top of desktop computers. Hackers are interested in smartphones to steal sensitive information of users and gain financial gain. New variations are becoming more challenging to do so with the anti-ransomware programs [7]. Among the threats that have been favorable to people, depending on new applications and technologies, is Ransomware. The above caution is not part of the traditional notions of threat and intimidation and is a branding, an outlawed marketing technique. Such a plan, as limiting the entry of people and requiring them to buy hush money to recover information, must be justified by the fact that people and enterprises become more and more dependent on and even demand sensitive information. Sadly, this business approach has been highly lucrative since 2013, and it has resulted in colossal financial losses. As cybercrime emerged from being more than a malware that exploits the vulnerabilities of a system to one that can propagate beyond a network, it underwent numerous processes. Ransomware-as-a-service (RaaS), providing access to this tool to non-experts and increasing the number of potential sources of delivery, adds even more threat to this attack [8].

Crypt virology proves how cryptography can be used in an ill-intentioned manner by introducing a creative twist to it. It is obnoxious, which means it can be used to perform malware attacks that lead to data leakage, loss of secrecy, and loss of access, which cryptography usually prevents [9]. A novel malware detection method that does not need a PE executable file. An image of the picture PE header is produced to take advantage of the handy characteristics of PE header files. After that, CNN is used in accordance with its strengths to classify and extract features of images [10]. Ransomware attacks have three stages: the analysis of the victim's network traffic, the discovery of a vulnerability, and, finally, the attack. This has led to ransomware detection emerging as an essential activity [11] that has a variety of solutions based on security enhancement. The data is encrypted when it is prepared to be analyzed, and then the application programming interface (API) is invoked. PEDA is a two-technique detection algorithm that was created to increase the overall performance and accuracy of detection. Signature Repository (SR) identifies threats using a signature matching process. The predictive model undertakes the second tier of detecting a location, which is known as the Learning Algorithm [12]. Using MOGWO and BCS algorithms to decide whether an application is an instance of malware or Ransomware, DNA act-Ran and BCS algorithms first select the significant features via the concept of effective learning, so that after a digital Target DNA of a small set of features has been generated, the instances are then classified as either goodware or Ransomware. RanDNAact-Ran method identifies malware at three decisive steps of the process. Generation of DNA sequences, feature selection, and ransomware detection [13]. An architecture that leverages the specifics of Android ransomware, uses machine learning models to detect malicious and safe applications, and carries out a comparative analysis to determine how long the machine learning model detection of Android ransomware will take [14].

In the case of unsupervised ransomware detection, previous work has had trouble with high false positives. RDVL, with its ensemble voting process, minimizes such false alarms by making use of several models voting for the final prediction to increase the sensitivity and specificity of the ransomware detection. Also, RDVL solves scalability problems by combining lightweight models with ensemble learning, enabling it to operate effectively even in massive real-time network conditions.

Ransomware criminals are faced with new copies of the malware every day and are knowledgeable about new ransomware families that are announced in the wild. To counter this growing threat, scholars and managers have come up with new measures of preventing such cyberattacks. This follows a newly identified research paradigm of ransomware security, a low-level file system I/O log, that is grounded in IRP. This therefore means that the logs of minimal ransomware samples are evaluated in this study among the various ransomware methodologies retrieved each time to get viable and detailed information of the behavior of a ransomware. Besides learning the fundamental trends of the IRP logs, a research undertaking on the creation of an appropriate ANN architecture to detect the viruses. To demonstrate the efficacy of our strategy, an experiment with an ANN model is conducted, and three experimental settings exist [15]. Primary results of our extensive study on the BadRabbit malware are presented. The hardware specifications used to analyze the main hardware are the Intel Core i7 8550U, having a 1.80 GHz speed and 16GB RAM. VirtualBox is the one that controls the machines of an analysis. Two virtual machines, again, the REMnux and Windows 10, were used to perform formal verification. Android ransomware detection is not equipped with essential elements and uses monitored machine learning processes. However, there are several weaknesses with supervised machine learning approaches.

The suggested innovative IDH suggests accepting CEP to collect data in accordance with the sources, such as HoneyFolder, software-defined network, hosts, Audit Watch, and firewall. Such data is then converted to an instance of an event, and these events are recognized in the CEP engine based on knowledge of the rules at hand (their availability) and acting on them through aggregation to acknowledge the

activity of the malevolent and profile outbreaks and take immediate action. Distributed systems of security at the enterprise level are likely to produce huge volumes of data. It is very challenging to determine the abnormalities in data streams, and it goes a long way in consuming a lot of computing power. CEP has been applied to gauge, predict, and identify the occurrence of aberrant activities in complex data streams. Additional previous studies on CEP-based hybrid IDS [16] explain how CEP-based IDS may be used in real-time intrusion detection. It fails to acknowledge the privilege escalation, the actual attack that follows the release of the Ransomware and malware. In this article, they will be used as an example, and the answers will be drawn on various streams of events in multiple situations, such as malware and user activities analysis, to prove the validity of real-time answers. This research contributes to the sound IDH to isolate Ransomware and prevent it as follows [17].

The value will be placed on all the pages, and the value of RARI will be placed on the value of backing up or not backing up the page before it. That is the way AMOEBA provides a more detailed method of management of backup copies. AMOEBA also does not save more than one page of backup on each page to save storage space. With AMOEBA, the overhead of running time and backup space will be reduced since only the required copies of backup are being kept. The ransomware post-attack period is also reduced. ACO is also helpful to optimize the GC operation by transferring backup pages and pages edited by Ransomware to an alternative NAND block. This separation is beneficial to the SSD in reducing the GC overhead that is caused by legal page copy operations during the GC process [18].

The change of the computer system will serve as the basis of the proposed mechanism to detect the ransomware attacks. These alterations are rather opposite to the regular pattern of system utilization and are aligned with the behavioral pattern of the Ransomware. To make this identification easier, researchers came up with an FSM model. The states in this system are the current state of the computer system. These changes of state are a result of the events in the system that should satisfy some requirement. The proposed plan is segmented into two modules, which are the behavior analytics and decision-making modules [19]. A behavior-analysis module maintains a history of the usage, persistence, lateral movement, and system resource access through the respective listeners. Where the system has undergone some significant modifications, the FSM model listeners lead to state transition.

It is also on the move in setting the limits of the pre-encryption phase of the crypto-ransomware life cycle. The Dynamic Pre-encryption Delineation of boundaries technique was put forward. To prevent data sufficiency issues during attack feature extraction at pre-encryption, an annotated TF-IDF method was proposed. To demonstrate the improvement that was made by the DPBD-EF method, a large-scale experimental study was to be carried out. Otherwise, unless this is stated, which is again in the context of the present study, malware and symmetric encryption should be treated like this. [20].

Some of the tools include the penetration test, which maintains a journal of the process by the attacker. It does not dilute or dissipate its impact. The honey pot operation is non-partisan and is designed to appear to be the actual world scene and to make the enemy come into the trap. Honeypots are set up to seem to the attackers as an operating system and are attacked by the attackers. It gathers data on the attacker, following the actions of the attacker, and gives the data needed. Honeynet will not focus on blocking an attack, and its main agenda will be to capture evidence on the attack and the attackers [21]. These are two categories of honeynet: production and research honeynet. Production Phishing emails may be used to gather as much information as possible with the hope of learning more about the attacker and reducing the impact of the threat to the organization. Knowing this, one can better understand what kind of failing points are present within the current environment, and hence, a more resilient method of defense can be developed. Since the Honeypot has no usable resources, all the received or sent traffic to the Honeypot is suspect [22]. The Honeypot access is considered unauthorized access. To establish a high-security system, the honeynets can be employed to harvest high-value information [23]. The traditional approaches to evaluation include either fixed or/dynamic approaches to the existence of a ransomware application or not. These are the research activities that may be integrated into a real-time ransomware detection technique. It is not problem-free, however, e.g., inability to detect new, variant, or unknown Ransomware. These kinds would need signed patches prepared by a trusted party, and would not otherwise be prepared beyond on-demand as a response to attacks. A second investigation on real-time ransomware detection within a PC setting suggests the monitoring of key files, and after the application is authorized, it is assumed safe and permitted [25]. Vital files will not, however, be accorded privileges to audited programs. To avoid the occurrence of some new risks and ensure that the user's intent is similar, they can apply this strategy to an Android environment. As a result, this work of literature suggests a solution to quickly identify new, variant, or unknown Ransomware in an Android environment [26].

These and some other attacks indicate that the cryptographic tool usage is a resource that needs to be greatly controlled [27] and its own management and auditing needs, in general-purpose computer environments. When drawing a comparison between the different results, the fact that the random forest-based approach to malware exposure is vastly superior to the one that depends on KNN and logistic regression is undoubtedly evident. The suggested ML models demonstrate similar or even the same high performance in detection as methods of deep learning. In situations where it was scraped off publicly accessible repositories, the data were sampled by using sandbox analysis. The sampling datasets were used to create machine learning models. The RARI measure is selected in the AMOEBA ML module that calculates the significance of each indicator. AMOEBA is highly recognized for Ransomware, thus AMOEBA does not need to employ as much SSD space to hold the data, merely to store as few backup pages in one logical page as possible. The recovery itself can be done in a very simplistic way because they [29] can translate the state of all the backup pages into the legal state. In situations where features are numerous, one might not be able to produce the most suitable features because the search space is vast. DNAactRAN is an application that employs MOGWO and BCS [30] to extract the desired features in the obtained dataset [30].

3. Primitives

3.1. Ransomware vectors

The protocol standards on which RDP is based are extensions of the existing standards. Presentation data can be transmitted using a multichannel capable protocol with serial device connectivity, license evidence, and encrypted data using various virtual channels such as keyboard and mouse channels. The communications between the Terminal Server and Client are carried out on the RDP. RDP is encrypted in TCP. The misconfigured RDP ports are left open to the Internet when the network is to be accessed, and cyber threat actors (CTAs) employ them. Through RDP, lateral travel across a network, escalate, steal sensitive information, steal passwords, and install all kinds of malware can all be done. As the CTAs are hooking up to a real network service and, therefore, get the same functionality as any other remote user, this common attacker trick would aid them in avoiding detection. The CTAs scan the Internet to detect open ports of the RDP services, including the Shodan search engine, and use brute force password cracking techniques to infiltrate vulnerable networks. Dark web markets are also broken by the means of RDP credentials.

A number of classifications of Ransomware are employed in executing cyberattacks to require ransoms. Crypto and locker are the most significant ones. Criminals have introduced two other forms of Ransomware, and they are: double extortion and Ransomware as a service. Social engineering is often employed to compromise confidential data of the sources, in particular, usernames and passwords, and credit card numbers, through deceit. Social engineering is the process whereby an attacker pretends to be a person they are trusted, and exploits the Internet, messaging service, or text victim to their detriment. Malware can also infect the target, as they have been tricked into using an infectious link. It may result in the Ransomware being used to lock the computer or display personal data.

On the one hand, hackers use the developed systems to probe the security vulnerabilities. The scan will also be able to give the attacker information on which kind of software is on the system, whether it is up-to-date or not, and whether it is a vulnerable software package. Without knowing all this, the machine will not be sufficiently prepared against an attack. The attacker can also run malicious commands on the targeted system in case the attack is successful. The haphazard states of the system supplied by ransomware programs and benign programs are precisely defined and modeled through a formal system, a finite-state machine. It is modeled as a finite state machine, and a collection of listeners and decision-support modules is utilized to identify a ransomware attack. The decision-making module, in turn, is defined by the state-change listener, which observes any change of the system and, in its turn, triggers an alarm in case the FSM transitions towards the probable ransom attack. When the notification is made, the decision-making module attempts to halt the processes in question and notify the user that they are supposed to do additional things. Ransom is not a mystical means of concealing file encryption and ransomware initiatives.

One contains bugs that can cause serious issues, such as stealing data, and bugs that can cause system failure. The minor ones can result in incorrect output or exception messages- bugs cause programs to act in a strange manner. Virtually any software contains minor (or severe) bugs. Even though it is impossible to write bug-free code, it is invariably possible to detect and remove any serious bugs that are likely to put security at risk.

1) Statistical Analysis

Data collection is the process of gathering precise insights for study and using defined, recognised techniques to analyse and evaluate them.

2) Benign Application

In order to collect the benign applications, the Microsoft App Store is used, which is the official application store for desktops. The dataset consists of around 16,000 randomly collected applications from the entire genre, like games, education, tools, and books, collected from the period of May 2020 to May 2021. The hundred most popular applications are also downloaded.

3) Malicious Application

The malicious applications are collected from the malware research website. Around 32,000 malicious applications are downloaded from virusshare.com and later executed for feature extraction. These malware were generated from the period of May 2013 to March 2014.

4) Dataset Generation

A sizable collection of both good and bad applications is gathered. From a common collection of binary files, binary files were produced. Files with the ".dll" and ".exe" extensions make up most of the file format.

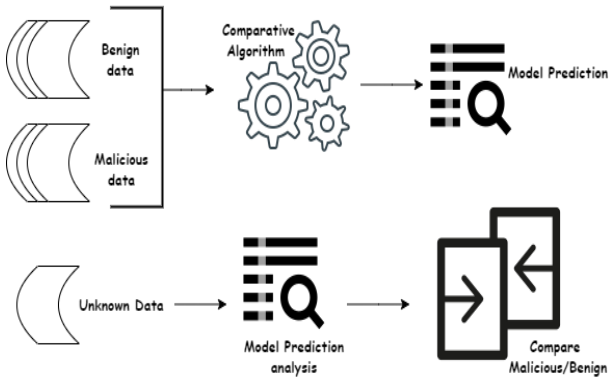


Fig. 1: Data Model Prediction with Analysis.

Malware and Ransomware each use 16000 samples, whereas benign samples use the same 16000 to avoid biased training and decreased accuracy. It is critical to distinguish between illusion and metadata management. Master data focuses on corporate entities, whereas statistics focuses on classification and categorization. Master data changes will always be managed as part of current business processes. However, a shift in reference data values may require a change in business procedures to accommodate the change. This is another difference between reference and master data. It is standard business practice to add existing clients or sales items, for instance. On the other hand, adding an additional product category (for example, "limited sales item") or adding an additional customer type (for example, "gold level client") will require adjustments to the business procedures that control such things.

The malicious dataset in this research (2013-2014) was selected because it was available and had already been determined to offer the feature sets it does. Nevertheless, we appreciate the fact that variants of Ransomware have been changing with time. In order to overcome this, subsequent research will look into using newer data sets such as CICIDS2017, which have more recent samples that are more reflective of the current ransomware climate.

5) Training Dataset

The data set obtained is then entered into a random data algorithm that has the desired training parameters. Preprocessing of data, also referred to as data organisation, is a requirement of data training. Relationships between data should be co-relationships to extract features. Besides, cleaning of the data is quite natural. This will assist us in identifying and incorporating any missing data and values in the data. The practical completion of the data training part will enable the effective prediction outcome to take place with appropriate filtering, cleaning, and scaling.

6) Information Retrieval

Because of the necessity to possess quicker calculations and reduced use of space, it is essential to extract the features accordingly. The data is contained in regular and malicious programmes. The Python file package will extract the features from that dataset and is more frequently used when working with apps. The majority of desktop applications' installer packages are in the form of an .exe, which can be unbundled by use of the file library. It helps absorb all the necessary features and create a CSV file that sorts out the data. Early innovations: As the requirement for a massive volume of data rose, there was a necessity to create data structures that would enable easier accessibility.

The index is among such data structures that allow for the retrieval of information more rapidly. Hierarchies have been classified by hand ranking indexes, ages old.

3.2. Voting ensemble learning

1) Ensemble Techniques

Most machine learning algorithms are the best way to combine predictions. This then makes two or more independent models based on the trained dataset. When the model is asked to make predictions on new data, they (models can be attached to a voting classifier that will average sub-model predictions. The weight can be placed on sub-model predictions, but it isn't easy to do so by hand or even heuristically while defining the classifier. The more complex (advanced) stacking (stacked generalization) is not yet available in Scikit-Learn, but it can be applied to learning how to reduce the predictions of the sub-model. Techniques can be classified depending on the types of learning methods, such as direct learning. Bagging enchanted members are gradually trained, and estimates are also gradually trained and will affect the training of the next member to that point.

2) Building the Dataset

The data collected is used to remove both benign and malicious programs in the prefile library. Hash value features, dibgue size, dibgue relative virtual address, major version of the image, and many more were recovered. The trained algorithm is trained on a dataset manufactured, including all parameters and layers. The number of features used in the algorithm training is 16. Training is done through the implementation of good and bad programs. Before achieving data changes, raw data can be assembled, a source of convenience, and identified label data can be chosen, and the technology used to sample can be chosen, and data can be separated.

3) Data Preprocessing

In model training and testing, predicting features is essential. Successful feature prediction leads to faster computations and reduced memory usage. Once the features are extracted, the whole dataset is stored as a CSV file. This is followed by data analysis to identify correlations among the dataset elements. Additionally, data cleaning should be performed to address any missing properties or values. After scaling, cleaning, and appropriate filtering, the data can be used for training and testing. Interval random data is divided into two parts: random, which means the sampling is random, and data, which refers to a set of decisions that are easier to remember than theoretical concepts. In the case of a dataset, a small portion is used as the training set and is divided into groups and subgroups. Moreover, the groups and subgroups are interconnected, and this is determined by the algorithm. This leads to the creation of a data collection comprising multiple datasets. Each piece of information is unique. Variables are randomly selected during each data formation and split. The remaining dataset, which is not used for training, is then used to predict information in sets, resulting in the optimal classification of data points. The information with the best forecasting capability is displayed as output. A set of labels (1 for malware and 0 for benign apps) is then used to determine the nature of each program. This reduces the uncertainty in classifications and splits the training set into two subsets with distinctly different labels at every node of the decision information.

4) Learning in a cluster

This experiment, conducted by the Machine Learning Group, aims to estimate Ransomware and compute the test accuracy or aggregate of the two models. This further enhances the accuracy of the individual machine learning models. It demonstrates that the null hypothesis statements of the two machine learning models are incorrect when an ensemble method is applied. The deep learning model achieved over 80 percent accuracy in identifying ransomware PE files compared to non-ransomware files. However, the accuracy dropped below 80 percent when testing the null hypothesis. When an ensemble technique is used to identify Ransomware in an application, selecting examples with 80% accuracy is more effective. In this section, the false and alternative hypotheses are defined and tested. The null hypothesis suggests that the accuracy cannot improve when two deep learning models are combined. However, using another hypothesis to combine these two deep learning algorithms resulted in increased precision. The ensemble deep learning model is characterized by objective quantitative data and is measured by the accuracy of the ensemble deep learning model. The ensemble model consists of two models that extract different attributes from the given PE file, with the numerical forms of the two models not being similar. The input to both models is created by returning the 50 most frequent opcodes and converting them into a percentage frequency vector. The second model uses feedback in the form of 100x100 pixel 8-bit grey-scale images, which are hashes of the number of words present in the photos. The PE file contains words that represent the numeric values of the pictures. To ensure the accuracy of the test, approximately 16,000 PE file samples are used in each forecasting category. The main challenge of the project is the design of the machine learning and its validity. The software is developed with the purpose of training and testing deep learning models. The overall objective of the project is to develop an ensemble model of deep learning.

5) Random data prediction

A model is developed once all required characteristics are defined and the dataset training is complete. The parameters for random data at each level of each decision tree are set, allowing the classification of programs, with a completely new program serving as new input to predict whether it is malicious or benign. When a new application is input, the Python PE File module is used to extract relevant features, which are then compared to the model created during the previous training phase. In this section, each feature is analyzed, its behaviors are learned, and all results are compiled as one array, after which they are used for prediction. Based on the properties gathered from the input application, it is categorized as either malware or benign. If the prediction outcome is 1, the application is classified as malware; otherwise, it is considered benign.

6) Cluster Learning Prediction

Ensemble learning is done with two possible models, and they are built with the aid of the Convolution Neural Network technique. Two of them include the string model and the Opcode model. Both models are described below:

String Model: In this case, the words of the raw bytes of the PE files are taken to obtain the numerical vector to be processed further out of the raw bytes of the PE files, which are just input samples. The UTF-8 characters are filtered too, by reading the bytes in PE files, and these characters are clustered together to form words separated by spaces. To extract the words, a Python code was employed to carry out the above operation. Due to the Natural Language Processing technology, these words are transformed into numerical values by the use of a hash. Here is the hashing trick, a method of converting characters to single numbers. However, there is yet another method of achieving this process, referred to as common-bag-of-words, that was not chosen based on its inefficiency in terms of memory storage and time. There is no necessity to follow the line between the corpus of texts and numerical presentation, computational, and time demands. Since, in this case, we are speaking about the process of big-scale analysis, the Natural Language Processing technology was used to normalize the words, and additional padding was done over the stream of words. With this, a promise of something better in doing the right can be proven.

Opcode Model: The second model used in the creation of the neural network is the Opcode Model. The data used in the study is analysis-specific. It picks the 50 most frequent opcodes, such as add, sub, etc., and fully disassembles the input PE file by calling the Capstone Disassembly Engine, which is another point of importance in the Capstone framework based on the MC component of the LLVM architecture. Capstone disassembler is thus one of the most suitable disassemblers when it comes to decompiling the unprocessed bytes of PE files. The mapping is unveiled from the assembly code present in the PE code of the files used in extracting features when they are used in the past, since it is more precise. A sample gives a sheet with the values of the percentages of occurrence of each of the opcodes as a percentage of the frequency of occurrence of each opcode in the top50 sheet. As a result, a bag of words representation is created that is a 50-opcodes vector. This was better fitted in the dataset than the UTF-8 strings model due to the vocabulary size control. In order to understand the mechanism of the strategy, to compare the UTF-8 strings approach, and to compare both, the frequency of the Opcodes is visualised as a frequency histogram. That is, both the Opcode model and the string model, despite being designed by constructing a convolutional neural network into a deep learning ensemble, are first modeled. The average of the two models is considered a final ensemble model.

4. Proposed Method

Here, we state the process that the proposed method is going to follow in a brief manner. This approach can be divided into three stages, i.e., preprocessing, feature selection, and classification. During the initial stage, we will carry out L1-Norm maximization, which transforms the raw data into CSV data, followed by feature selection using the PCA method. A combination of these techniques enhances the prediction of features by eliminating a majority of the features of the data. The third phase is when we follow the combination of the RDVL method and the ensemble method. The process is divided into three pipelines: Bagging, Boosting, and Voting. We draw the architecture diagram of the proposed method, as shown in Figure 2 below.

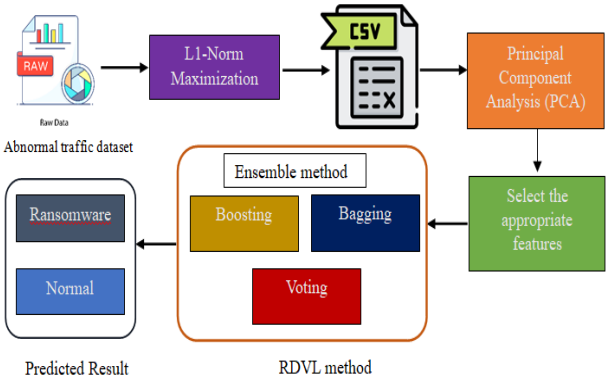


Fig. 2: Architecture Diagram of the Proposed Method.

When applied as a preprocessing method for the training and testing in the context of ransomware detection, our PCA with L1-Norm Maximization compared favourably. It improves feature prediction by removing most of the data's attributes and condensing it simultaneously, with essential information retained in a lower dimension. This makes the detection more accurate because all the vital features are kept, while models experience an enhanced ability to detect ransomware-related anomalies. Also, by reducing the feature space, PCA increases the rate at which a learning algorithm is trained and tested for a machine learning model. It also helps to organize memory space efficiently due to the creation of concise datasets saved in CSV format for further use and ad-hoc real-time applications. The result also shows that the PCA preprocessed performs better when combined with other ensemble learning methods like Bagging, Boosting, Voting, and Random Forests. Bagging has low data complexity, allowing multiple models to train on subsamples, which enhances stability and reduces variance. Boosting has the advantage of concentrating on misclassified cases using rich features, but voting integrates predictions from various models better, owing to relevant and straightforward features.

4.1. Principal component analysis (PCA) based on L1-norm maximization

L1-Norm Maximization as a preprocessing PCA technique is aimed at making Machine Learning (ML) efficient in training and testing while reconstructing feature prediction. This technique is instrumental in ransomware detection from datasets consisting of abnormal traffic, since large and high-dimensional datasets can be hard on the computational resources. L1-norm maximization enables PCA to concentrate on high-variance features while remaining sparse and easily interpretable.PCA-L1 Norm Maximization enhances the precision of the model's identification of anomalies reliably and with improved accuracy. After this feature selection step, the dataset is saved as a CSV file after dimensionality reduction to save significant time and memory during training and testing. The data preprocessing method significantly enhances the efficiency of the normalization process, decreasing model convergence time and reducing disk space while maintaining comparable detection rates. The original PCA is formulated as an L2-norm maximization problem in which principal components are computed to maximize the variance of the data. In the case of L1-Norm PCA, the measure of focus is changed to the sum of absolute values of the projections since it favors sparse components. Through equation 1, we find the L1-Norm PCA general optimization problem,

$$\max_{u} \sum_{x=1}^{n} \left| u^{T} i_{x} \right| \text{ subject to } \|u\|_2 = 1 \tag{1}$$

Let us assume, Ii_xs ththe hx − theh data point in the dataset, u as a weight factor which represents the direction of the principal component, n as the total number of data samples, and $\|u\|_2$ to ensure u is a unit vector. The optimization objective is to maximize the L1-Norm (absolute sum) of the projected data points onto the principal component direction u, as defined by this equation. In contrast to conventional PCA, L1-Norm PCA generates sparse principal components that are easier to understand and are resistant to outliers. Afte comthe $(u_1, u_2, \ldots, u_k)$ principal components, we transform the data into a lower-dimensional space through equation 2,

$$v_x = U^T i_x \quad (2)$$

Let us assume, $U = (u_1, u_2, \dots, u_k)$ as the matrix of the top k, k as principal components, and v_x as The transformed feature vector in the reduced space. Using this equation, the original dataset is projected into the subspace spanned by the top k sparse principal components. This treatment preserves the most essential information for ransomware detection while decreasing the dataset's dimensionality. Equation 3 can also be used to apply the L1-Norm constraint to the optimization problem to enforce sparsity in the primary components.

$$\max_u \sum_{x=1}^{n} |u^T i_x| \text{ subject to } \|u\|_1 \leq \lambda \quad (3)$$

Let us assume, $\|u\|_1 = \sum_{y=1}^{d} |u_y|$ as the L1-Norm of u, and λ as a regularization parameter, which is used to control the sparsity of u. This variant of the optimization problem restricts the L1-Norm of u, imposing a sparsity restriction on the principal components. The parameter λ controls the trade-off between the amount of variance collected and sparsity. Once the sparse principal components have been calculated, the most essential characteristics are chosen based on their contribution to the principal components. By Equation 4, we illustrated feature importance F,

$$F_y = \sum_{k=1}^{K} |u_{yk}| \quad (4)$$

Let us assume, u_y as the feature weight, k as the principal component, and K as the total number of principal components used. This equation measures each feature's significance according to how much it contributes to each of the primary elements. Higher relevance features are given priority for additional examination. Equation 5 saved the dataset in a compact format as a CSV file Z, following feature selection and dimensionality reduction.

$$Z_o = (v_1, v_2, \dots, v_n) \quad (5)$$

Let's assume Z_o as the output of CSV data. In this equation, each row in the CSV corresponds to the transformed feature vector vx. This equation effectively trains and tests ransomware detection models by storing the preprocessed dataset in a structured manner. The reduced dimensionality guarantees faster calculations and less memory utilization. ML models are trained and tested using a condensed dataset. By following equation 6, we evaluate the accuracy of the method,

$$A = \frac{1}{n} \sum_{x=1}^{n} 1(\hat{j}_x = j_x) \quad (6)$$

Let us assume, $\hat{j}_x$ as the predicted label for the sample v_x, j_x, where 1 is the indicator function. This equation measures the effectiveness of the preprocessing and feature selection pipeline in enabling accurate ransomware detection. The PCA method based on the maximization of the L1-Norm solves the problem of creating many irrelevant components while throwing away all dimensions that are sparse through the L1-Norm.

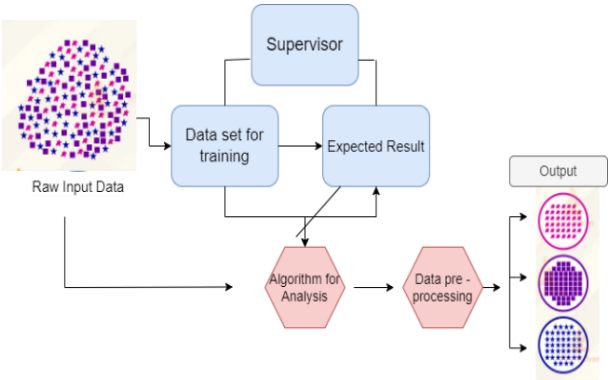


Fig. 3: Data Preprocessing with Raw Input.

PCA-L1 Norm Maximization provides an intermediate sparsity of principal components, making it easy for the user to interpret results between the two extreme poles of full sparsity provided by Lasso and the absence of any component sparsity, such as in the ordinary PCA. The preprocessing pipeline removes critical features and produces a low-dimensional cell based on it, formats the output data set in CSV format, and provides improved identification of Ransomware in the data sets of abnormal traffic. Every equation in the process is defined to choose the best features, improve computation time, and still give high detection rates.

4.2. Ransomware detection via voting and learning (RDVL) method

The identification of Ransomware from anomalous traffic is a significant concern, and the RDVL algorithm utilizes ensemble learning approaches to develop the model's precision and redundancy. Ensemble learning is a way of computing the power from various Voice Recognition models and using them to compute the results when combined, leading to enhanced accuracy compared to single models. The RDVL algorithm employs three key techniques for merging results from various models: Bagging (Bootstrap Aggregating), Boosting, and Voting. Specifically, we have the bagging technique, where several like models are learned at the same time, and each model is trained with a different sub-sample derived from the given training set. This way, variance and overfitting are also tackled because bags are created by averaging or aggregating the detection results. In boosting, several models, usually of the same kind, are trained iteratively, and every model tries to reduce the errors that occurred in the previous model. This kind of iteration increases accuracy because it concentrates on the complex samples to reduce bias and increases the model accuracy of the ransomware detection. By combining the results of multiple models, often of different types, voting works with the simplest of statistical methods, such as averaging in the case of regression-based

methods and simple classification voting, where the contending models vote to select the majority result. This technique makes the model less sensitive to noise since several models are involved, improving the overall probability computation. Using the ensemble learning techniques described, the RDVL algorithm can identify Ransomware based on anomalous network traffic. Bagging makes predictions stable, boosting makes them accurate, and voting uses different models; hence, ransomware detection is both thorough and effective. Through equation 7, the bagging process, which is used to aggregate predictions from base models for regression and classification,

$$\widehat{A}(i) = Z_o\left(\frac{1}{T}\sum_t^T A_t(i)\right) \tag{7}$$

Let us assume $\widehat{A}$ as the final aggregated prediction, i as the input data, T as the total number of models in the ensemble, and $A_t(i)$ as the prediction of the t model for i. Several models trained on bootstrapped subsamples of the dataset are combined in bagging for regression tasks. The predictions are then averaged to achieve a more consistent result and lessen the impact of outliers in the ransomware detection procedure. By following, we classify the majority voting through equation 8,

$$\hat{j} = \arg\max_c \sum_{t=1}^T 1(A_t(i) = c) \tag{8}$$

Let us assume c as a class label (for instance, Ransomware or regular traffic), ( )one as an indicator function, which returns one if $A_t(i) = c$ else 0, and $\hat{j}$ as the final predicted class. Bagging uses majority voting for classification, choosing the class label that most models predict as the final output. Pooling the power of several models trained on marginally different datasets guarantees robustness in ransomware detection. The weight is then predicted using a boosting method that successively combines several weak models, each improving on the errors of the one before it by using equation 9.

$$\widehat{A}(i) = \sum_{t=1}^T \alpha_t A_t(i) \tag{9}$$

Let us assume α_t as the weight allocated to the RDVL t method, proportional to its accuracy, $A_t(i)$ as the prediction by the RDVL t method for i, and T as the total number of models. This equation ensures that models that perform better at higher weights help minimize errors from earlier rounds. By focusing on difficult-to-categorize traffic patterns, this iterative approach is particularly effective in improving ransomware detection. By following, we perform the weight update rule for binary classification through equation 10,

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-e_t}{e_t}\right) \tag{10}$$

Here, we assume that the weighted error rate of the model is. This equation calculates the weight α_t for each model based on its error rate e_t. Models with lower errors receive higher weights, allowing the boosting algorithm to focus on improving weak models in the ensemble. Then we perform a voting process through equation 11, which is used to combine predictions from multiple models (e.g., Naïve Bayes, KNN, SVMs) to achieve a consensus.

$$\hat{j} = \text{mode}\{A_1(i), A_2(i), \ldots, A_T(i)\} \tag{11}$$

Here, we use mode to predict the class labels with the highest frequency by the RDVL method. In hard voting, the majority class across all model predictions is selected as the final output. This is suitable for ransomware detection where diverse models contribute to a robust final decision. By following in 12, we perform a soft voting equation to compute the weighted probabilities,

$$\hat{j} = \arg\max_c \sum_{t=1}^T \alpha_t Q_t\,(j = c|i) \tag{12}$$

Let us assume that αt is assigned the weight to the RDVL t model based on its accuracy, and Q is the Probability of the class. In this equation, the RDVL method outputs class probabilities, and the final prediction is determined by the class with the highest weighted likelihood. This ensures a nuanced and reliable prediction by considering the confidence levels of individual models. To train the ensemble, the input dataset G through features $I = \{i_1, i_2, \ldots, i_g\}$ and $j = \{j_1, j_2, \ldots, j_n\}$ is employed. By following, we transform the feature space through equation 13,

$$I' = \Phi(I) \tag{13}$$

Let denote I as the transformed feature space, and -dimensionality reduction function (e.g., PCA or feature selection). This equation will guarantee that irrelevant or redundant characteristics are removed, it will reduce the complexity of the computation, and enhance the capability of the model to concentrate on the most important ransomware detection characteristics of abnormal traffic information 14.

$$\mathcal{L}_{RDVL} = \sum_{t=1}^T \alpha_t \mathcal{L}_t \tag{14}$$

Let us assume α_t as the weight for the RDVL t model, and $\mathcal{L}_t$ as the loss function for the t model. This equation makes sure that models with high precision put more effort into arriving at the final forecast. This assists the RDVL algorithm in approaching the optimal solution for detecting Ransomware. Here, the RDVL algorithm combines bagging, boosting, and voting approaches; each model improves ransomware detection from stranger traffic. Bagging is used to decrease the variance, boosting is used to minimize the bias, and lastly, the idea of voting is used to take advantage of model diversity. These equations express the level of ransomware detection with high accuracy, a minimum number of false alarms, and reasonable time consumption by the algorithm. In Figure 4 below, we illustrate the flowchart diagram of the RDVL method.

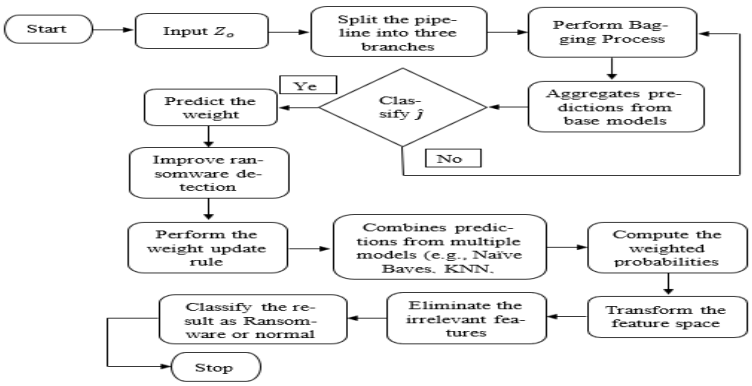


Fig. 4: Flowchart Diagram of the RDVL Method.

It starts with the initiation of the Z o. Traffic data network communications are collected and analyzed as inputs, both standard and Ransomware traffic. The workflow is divided into three major branches: Bagging, boosting, and voting techniques. Several models are trained using reduced training sets, and an averaging or a voting system is used to combine the results. The models are developed sequentially, and there is an understanding that the later models will learn from the mistakes of the earlier models. Single models and a combination of models are constructed, combining outputs of the models that are combined using voting methods or probabilistic ratios. A combination of the results of bagging, boosting, and voting methods provides a single prediction. The final prediction of the unified prediction categorizes the network traffic as either Ransomware or Normal Traffic.

5. Result and Discussion

An RDVL voting selector is a machine learning method that gathers information by examining several techniques. It forecasts performance by looking at the class with the highest likelihood of throughput. Voting classifier supports alternative input formats. A meta-estimator of this type is frequently used to reduce the variance of the estimator (such as a decision tree) by adding randomization to the building of a black-box estimator. The Estimator Code is all about the Voting ensemble learning methods. In this, panda and numpy is used for the model selection along with the datasets and their metrics. The data file has both the training data collection and the trained data collection. Samples will be tested with the trained datasets using sklearn.

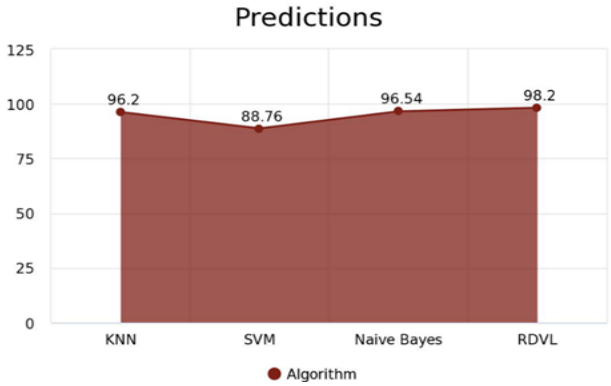


Fig. 5: Prediction Analysis.

The suggested work's primary goal is to improve accuracy, which it accomplishes well in that it gains high accuracy using the RDVL voting classifier. In comparison to KNN, SVM, and Naive Bayes, which each use the same number of terminals, the RDVL obtains 98.12% time analysis out of a total of 96 techniques.

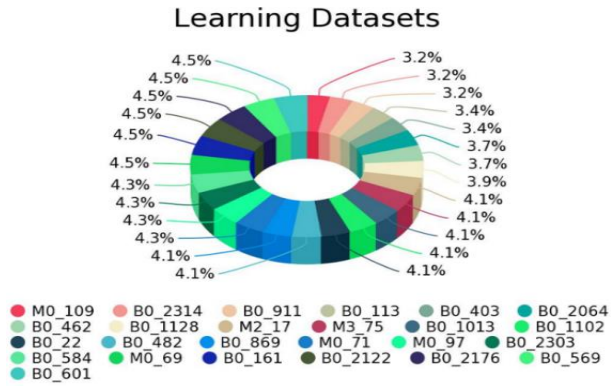


Fig. 6: Dataset with Random Subsets.

KNN, SVM, and Naive Bayes get 96.67%, 90.6% and 97.82% accuracy out of the same number of terminals, respectively. Figure 6. Analysis confirms that the suggested ensemble learning strategy provided a high level of vote accuracy. The proposed work's primary goal is to improve accuracy, which it accomplishes well in that it gains high accuracy using the RDVL voting. Stability is another key aspect of

bagging. Starting with a certain number, as the ensemble size grows, the error rate becomes nearly constant. Strength-based ensemble RDVL voting is less reliable than the remaining voting techniques, due to the thorough understanding of voting and its uses in machine learning.

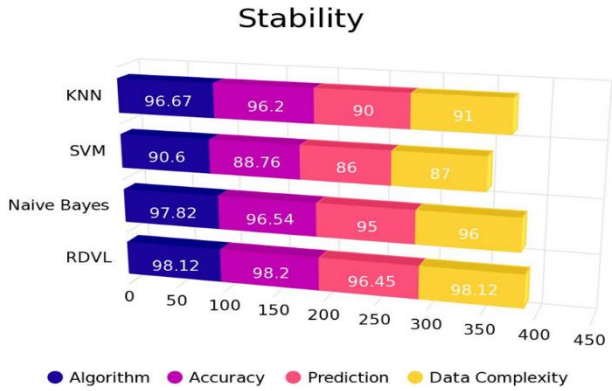


Fig. 7: Stability Aspects.

Figure 7. Illuminates the RDVL voting with the KNN, SVM, and Naive Bayes on the sensitivity metric in a graphical comparison. Constraints count variation is the focal point of the comparison.

Table 1: Comparison of Algorithms

Algorithm	Accuracy	Time Analysis	Data Complexity	Prediction
KNN	96.67	91	90	96.2
SVM	90.6	87	86	88.76
Naive Bayes	97.82	96	95	96.54
Voting	98.12	98.12	96.45	98.2

The predictive modeling can be significantly improved by combining machine learning models. Although this ensemble method might be a valuable choice for developing models, it should not be used as a default strategy because it is more expensive and does not consistently outperform individual models.

Table 2: Comparison Table for Data Prediction and Sensitivity

Algorithm	Data Prediction	Sensitivity
KNN	95	98
SVM	93	96
Naive Bayes	98.17	98.34
RDVL	97	96.34

Illustrates the comparison of the proposed system's accuracy using KNN, SVM, Naive Bayes, and RDVL voting classifier. The suggested classifier achieves a prediction accuracy of 98.2%; nevertheless, for the same number of terminals, KNN, SVM, and Naive Bayes achieve accuracy of 96.2, 88.76, and 96.54, respectively.

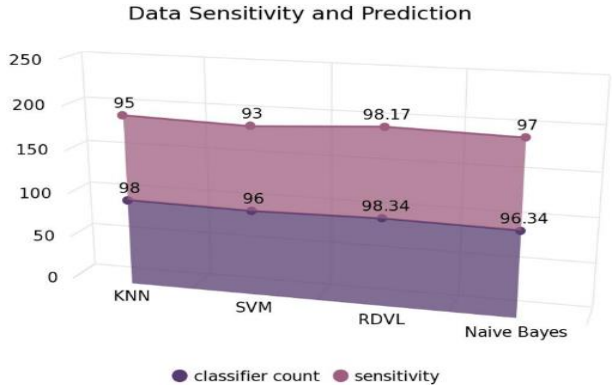


Fig. 8: Data Sensitivity and Prediction.

The number of correctly identified positives by the classifiers is referred to as the sensitivity metric. The ensemble classifier-based RDVL centering approach achieves 98.17% sensitivity. The current classifier performs worse than the proposed classifiers for a given node count. As a result, this comparison shows that the suggested approach achieves better performance than the current classifiers.

The RDVL has limitations, although it is effective. The computational overhead of real-time deployment because of processing vast amounts of network traffic is one of the major concerns. Also, the ransomware attacks against zero-day vulnerabilities have not been evaluated in full against RDVL. This is what will be researched later in the work, where optimization strategies can be discussed, like model pruning and distributed processing, to make it much more efficient in the context of real-time applications.

6. Conclusion

The harmless and malicious applications in our physical world are tested against the combating ransomware applications that affect the desktop. The behavioral attributes of harmless and threatening applications are looked at to identify every infection. The data extracted is then trained using machine learning, using a random data classifier and Voting Ensemble Learning using RDVL. The RDVL data management method achieves a prediction value and accuracy of 98.2 and 98.12 when compared to the other methods. Ransomware detection is designed with the help of data creation using computational methods and Ensemble detection, and will make reporting about monitoring files and testing Ransomware more productive. The regularity of scanning these files with our method will go a long way in detecting in advance and taking relevant action. A statistical method can be employed in the future to prevent the challenge of over-fitting and enhance the correctness of the research. To detect benign apps, more benign apps can be incorporated to research the different types of apps that are frequently involved. In the future, app distribution channels will be supported with the help of RDVL in combating malware.
Future studies can examine how the RDVL can be applied to the detection of mobile Ransomware, where device-specific properties and actions can pose their own challenges. Moreover, the compatibility of RDVL and zero-trust architectures may increase its resilience in environments with a high level of security and provide better results in detecting Ransomware at various access points and networks. In addition, research into whether RDVL has the capacity to support Ransomware as a service (RaaS) models and its ability to support IoT networks would offer more information about how scalable it is and how well it can be used.

References

[1] Shweta Sharma, C. Rama Krishna, Rakesh Kumar, RansomDroid: Forensic analysis and detection of Android Ransomware using unsupervised machine learning technique, Forensic Science International: Digital Investigation, Volume 37, 2021, 301168, ISSN 2666-2817. https://doi.org/10.1016/j.fsidi.2021.301168.

[2] Akhtar, M. S., & Feng, T. (2022). Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. Symmetry, 14(11), 2308. https://doi.org/10.3390/sym14112308.

[3] Zahoora, U., Khan, A., Rajarajan, M., Khan, S. H., Asam, M., & Jamal, T. (2022). Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier. Scientific Reports, 12(1), 1-15. https://doi.org/10.1038/s41598-022-19443-7.

[4] Li, Guan, Shaohui Wang, Yanbin Chen, Jie Zhou, and Qihang Zhao. "A hybrid framework for ransomware detection using deep learning and monte carlo tree search." (2024). https://doi.org/10.31219/osf.io/cjyvb.

[5] Alomari, E. S., Nuiaa, R. R., Alyasseri, Z. A., Mohammed, H. J., Sani, N. S., Esa, M. I., & Musawi, B. A. (2022). Malware Detection Using Deep Learning and Correlation-Based Feature Selection. Symmetry, 15(1), 123. https://doi.org/10.3390/sym15010123.

[6] F. Khan, C. Ncube, L. K. Ramasamy, S. Kadry and Y. Nam, "A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning," in IEEE Access, vol. 8, pp. 119710-119719, 2020. https://doi.org/10.1109/ACCESS.2020.3003785.

[7] Ramkumar, M., et al. "Identifying cancer risks using spectral subset feature selection based on multi-layer perception neural network for premature treatment." Computer Methods in Biomechanics and Biomedical Engineering 27.13 (2024): 1804-1816. https://doi.org/10.1080/10255842.2023.2262662.

[8] S.H. Kok, A. Azween, NZ Jhanjhi, Evaluation metric for crypto-ransomware detection using machine learning, Journal of Information Security and Applications, Volume 55, 2020,102646, ISSN 2214-2126. https://doi.org/10.1016/j.jisa.2020.102646.

[9] Manabu Hirano, Ryo Hodota, Ryotaro Kobayashi, RanSAP: An open dataset of ransomware storage access patterns for training machine learning models, Forensic Science International: Digital Investigation, Volume 40, 2022, 301314, ISSN 2666-2817. https://doi.org/10.1016/j.fsidi.2021.301314.

[10] Sakthivel, S., and B. Dhiyanesh. "A privacy-preserving storage security for spatial data in dynamics cloud environment." 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). IEEE, 2013. https://doi.org/10.1109/ICCCNT.2013.6726759.

[11] Gowtham Ramesh, Anjali Menen, Automated dynamic approach for detecting Ransomware using finite-state machine, Decision Support Systems, Volume 138, 2020, 113400, ISSN 0167-9236. https://doi.org/10.1016/j.dss.2020.113400.

[12] Ilker Kara, Murat Aydos, the rise of Ransomware: Forensic analysis for windows based ransomware attacks, Expert Systems with Applications, Volume 190, 2022, 116198, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2021.116198.

[13] S.H. Kok, Azween Abdullah, NZ Jhanjhi, Early detection of crypto-ransomware using pre-encryption detection algorithm, Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 5, 2022, Pages 1984-1999, ISSN 1319-1578. https://doi.org/10.1016/j.jksuci.2020.06.012.

[14] M. Basnet, S. Poudyal, M. H. Ali and D. Dasgupta, "Ransomware Detection Using Deep Learning in the SCADA System of Electric Vehicle Charging Station," 2021 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America), Lima, Peru, 2021, pp. 1-5. https://doi.org/10.1109/ISGTLatinAmerica52371.2021.9543031.

[15] C. Prasanth, R. P. Kumar, A. Rangesh, N. Sasmitha and D. B, "Intelligent Loan Eligibility and Approval System based on Random Forest Algorithm using Machine Learning," 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023. https://doi.org/10.1109/ICIDCA56705.2023.10100225.

[16] B. A. S. Al-Rimy et al., "A Pseudo Feedback-Based Annotated TF-IDF Technique for Dynamic Crypto-Ransomware Pre-Encryption Boundary Delineation and Features Extraction," in IEEE Access, vol. 8, pp. 140586-140598, 2020. https://doi.org/10.1109/ACCESS.2020.3012674.

[17] I. Almomani et al., "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data," in IEEE Access, vol. 9, pp. 57674-57691, 2021. https://doi.org/10.1109/ACCESS.2021.3071450.

[18] Gulmez, S., Gorgulu Kakisim, A., & Sogukpinar, I. (2024). XRan: Explainable deep learning-based ransomware detection using dynamic analysis. Computers & Security, 139, 103703. https://doi.org/10.1016/j.cose.2024.103703.

[19] P. Bajpai and R. Enbody, "Memory Forensics Against Ransomware," 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, 2020, pp. 1-8. https://doi.org/10.1109/CyberSecurity49315.2020.9138853.

[20] M. Al-Janabi and A. M. Altamimi, "A Comparative Analysis of Machine Learning Techniques for Classification and Detection of Malware," 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 2020, pp. 1-9. https://doi.org/10.1109/ACIT50332.2020.9300081.

[21] Amos Loh Yee Ren, Chong Tze Liang, Im Jun Hyug, Sarfraz Nawaz Broh, NZ Jhanjhi, Year: 2020, A Three-Level Ransomware Detection and Prevention Mechanism, EW, EAI.

[22] Shaukat, K., Luo, S., & Varadharajan, V. (2023). A novel deep learning-based approach for malware detection. Engineering Applications of Artificial Intelligence, 122, 106030. https://doi.org/10.1016/j.engappai.2023.106030.

[23] Karthick, Mr K., et al. "A subset scaling recursive feature collection based DDoS detection using behavioural based ideal neural network for security in a cloud environment." Procedia Computer Science 215 (2022): 509-518. https://doi.org/10.1016/j.procs.2022.12.053.

[24] Per Håkon Meland, Yara Fareed Fahmy Bayoumy, Guttorm Sindre, The Ransomware-as-a-Service economy within the darknet, Computers & Security, Volume 92, 2020, 101762, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2020.101762.

[25] A. H. Mohammad, " Kale, Bukola & Aworo, Solomon & Anyangwu, Cynthia. (2022). Cyber-Attacks on Digital Infrastructures in HealthCare: The Secured Approach.

[26] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," in IEEE Access, vol. 8, pp. 124579-124607, 2020. https://doi.org/10.1109/ACCESS.2020.3006143.
[27] Y. Pan, X. Ge, C. Fang and Y. Fan, "A Systematic Literature Review of Android Malware Detection Using Static Analysis," in IEEE Access, vol. 8, pp. 116363-116379, 2020. https://doi.org/10.1109/ACCESS.2020.3002842.
[28] Ammal, S.G., Saranya, K. et al. Advanced Cloud-Based Prediction Models for Cardiovascular Disease: Integrating Machine Learning and Feature Selection Techniques. SN COMPUT. SCI. 5, 572 (2024). https://doi.org/10.1007/s42979-024-02927-w.
[29] Adarsh Kumar Singh, Gandharv Wadhwa, Mayank Ahuja, Keshav Soni, Kapil Sharma, Android Malware Detection using LSI-based Reduced Op-code Feature Vector, Procedia Computer Science, Volume 173, 2020, Pages 291-298, ISSN 1877-0509. https://doi.org/10.1016/j.procs.2020.06.034.
[30] R. Feng, S. Chen, X. Xie, G. Meng, S. -W. Lin and Y. Liu, "A Performance-Sensitive Malware Detection System Using Deep Learning on Mobile Devices," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1563-1578, 2021. https://doi.org/10.1109/TIFS.2020.3025436.
[31] Ashour, M. (2024). Zero-trust architectures in enterprise networks: A comprehensive framework for next-generation cybersecurity. Electronics, Communications, and Computing Summit, 2(3), 18–27.
[32] Poornimadarshini, S. (2025). Topology Optimization of Brushless DC Machines for Low-Noise and High-Torque Applications. National Journal of Electrical Machines & Power Conversion, 45-51.
[33] Van, C., Trinh, M. H., & Shimada, T. (2025). Graphene innovations in flexible and wearable nanoelectronics. Progress in Electronics and Communication Engineering, 2(2), 10–20.
[34] Wilamowski, G. J. (2025). Embedded system architectures optimization for high-performance edge computing. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 47–55.
[35] Jeon, S., Lee, H., Kim, H.-S., & Kim, Y. (2023). Universal Shift Register: QCA Based Novel Technique for Memory Storage Modules. Journal of VLSI Circuits and Systems, 5(2), 15–21. https://doi.org/10.31838/jvcs/05.02.03.
[36] Prasath, C. A. (2025). Adaptive filtering techniques for real-time audio signal enhancement in noisy environments. National Journal of Signal and Image Processing, 1(1), 26–33.
[37] Veerappan, S. (2024). A comparative study of NFC and UWB technologies for secure contactless payment systems. National Journal of RF Circuits and Wireless Systems, 1(1), 49–57.
[38] Rahim, R. (2025). Lightweight speaker identification framework using deep embeddings for real-time voice biometrics. National Journal of Speech and Audio Processing, 1(1), 15–21.
[39] Surendar, A. (2025). AI-driven optimization of power electronics systems for innovative grid applications. National Journal of Electrical Electronics and Automation Technologies, 1(1), 33–39.