Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Insider Threat Ransomware Detection through ML on PE Files

[1] Osama Alhodairy, [2] Tarek Abbes

[1] The National School of Electronics and Telecommunication, Sfax, Tunisia

Email: osama.alhodairy@enetcom.u-sfax.tn

[2] The National School of Electronics and Telecommunication, Sfax, Tunisia

Email: tarek.abbes@enetcom.usf.tn

ARTICLE INFO

Received: 25 Dec 2024

Revised: 15 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

Ransomware has become an ever-present and com- plex cyber threat that exposes a large number of companies in all industries to significant ransomware risks. Though traditional ransomware mitigation measures have so far targeted outside dangers, the mushrooming cases of insider-driven strikes pose a new bugbear for cybersecurity practitioners. This paper focuses on the key issue of identifying ransomware insider threats using machine learning on a Windows Portable Executable (PE) file metadata. Our approach draws upon a dataset with no fewer than 138,581 PE files, including both ransomware and benign samples. Powerful feature engineering extracts discriminative properties from PE headers, sections, imports and binary pat- terns. Machine learning algorithms are implemented in this paper for the classification of ransomware insider attacks. 10 ML algorithms were applied to the dataset and we analysed their results, these are supervised learning (Random Forest (RF), XGBoost classifiers, Decision Trees, Gradient Boosting Machine (GBM), Light Gradient Boosting Machine (LightGBM), K-nearest Neighbor (KNN), AdaBoost, Logistic Regression and Support Vector Machine (SVM)), and a hybrid model consists supervised learning algorithms (Random Forest and XGBoost for classification). The hybrid model did best with a 99.49% prediction and recall when it came to identifying ransomware samples. Among the proposed algorithms the RF algorithm has the largest accuracy of 99.46%, while the rest of the accuracy values are XGBoost of 99.42%, Decision Tree of 99.10%, GBM of 98.91%, LightGBM of 98.89%, KNN of 98.77%, AdaBoost of 98.91%, Logistic Regression of 71.67%, and SVM of 98.91%. Compared with existing solutions, the proposed approach showed significant accuracy and generalization superiority.

Keywords: Insider threats; Machine learning; Supervised learning; Feature selection; Windows PE.

1. Introduction

This ransomware has rapidly become one of the financial destructive malware Publishers. While external threat actors are behind most ransomware attacks, equally damaging and growing are the insider threats introduced by employees or vendors [1]. the insider threat aspect of ransomware attacks is a double whammy, combining both bad intent and insider access together which can mean double trouble for corporate safety and data correctness.

The term “insider threat” refers to any type of security or data risk targeted against an organization from individuals who have internal access by virtue of being employed or recently departed, and also includes third-party business partners with privileged system or/and physical access. e.g. contractors, vendors etc. [2]. The average cost of an insider attack is numerous and more than other different cyber incidents. Attackers with internal access can more often be able to infiltrate deeper within networks and do more damage, demand larger ransom payments than external bad actors. According to data from IBM, “the average cost of downtime from ransomware now stands at $1.85 million per attack, and even more when the attacks involve insider access or assistance” [3].

The factors behind the surge in malicious insider threats are multifaceted. In recent years, monitoring insider activities have become more challenging due to remote and hybrid work models. Additionally, so-called “disgruntled insiders” more willing to sell access or launch attacks for ideology or profit as a result of economic uncertainty and social polarization [4]. Attackers can gain access correctly to begin with, and then they can exploit some of the legacy privileges and some of the internal controls that were lacking in some of the software, and they can use that over time for launching ransomware attacks [5].

An insider responsible for a targeted ransomware attack at Planned Parenthood resulted in the crippling of its entire network infrastructure, shutting it down globally for multiple weeks [6]. One of the recent studies shows that the potentiality of 51% of firms to get internally attacked. Compared to 2020, insider-enabled ransomware rose by 44%, as remote work and economic uncertainty provided more motives and opportunities for insiders to enable ransomware [7], such as shown in table I, Given the challenge of proper access controls and user monitoring fighting this growing internal threat inside organizations' own trusted networks. Ransomware insider at- tacks currently stand as one of greatest risks to enterprises worldwide. Hence, wider-reaching control technology, better staff screening and increased user monitoring are the tools organisations need to deploy.

Table 1. Insider Involvement in Ransomware Attacks

Year	Number of Attacks	% Increase
2019	163	-
2020	212	30%
2021	305	44%
2022	390	28%
2023	500	28%

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

1.1 Types, Descriptions and Comparisons of Ransomware

Ransomware assaults essentially type a couple of significant classes-cryptoral rypto-ransomware that encrypts files, locker ransomware that locks system access, and information surge ransomware that take steps to hole information [8] and so on. The most common so far is crypto-ransomware, with ransomware variants such as WannaCry or NotPetya encrypting a victim's files beyond possible recovery (if they do not have back-ups). Table 2, provides a complete summary of three different ransomware types, including their relevant encryption practices, primary effects, history, and usual ransom requests. The first type, which the aptly termed intelligibly refers to as "Encryption," uses solid algorithms (such as RSA- 2048+ and AES-256), to encrypt the file systems in full and thus ensure that the victim is deprived of data where no decryption keys exist. This has grown 167% annually on average, with median ransom payments rising to an average of over $570,000 [9], therefore demonstrating very significant financial consequences for those targeted.

Table 2. Ransomware Types Comparison

Type	Encryption Approach	Primary Impact	Growth Trend	Typical Payment
Encryption	Robust asymmetric Algorithms (RSA-2048+, AES-256)	Irreversibly locks file access	+167% per year	$570,000 average
Locker	Interferes with OS boot, system processes	Restricts system visibility and usage	+28% in 2022	Lower utility value
Data Exfiltration	Leaks stolen data to pressure payment	Data breach and encryption	+66% payment rate	$1.2 million median

The second type of ransomware, called Locker, works in a slightly more devious way by preventing the operating system from being booted and running critical system processes. This technique is helpful, but it lacks the ability to limit the visibility to the system, and it makes the device which is hacked useless. 2022 was seen more modest year of 28% year-on-year growth [10], the utility value of this variant has so far been perceived to be lower compared to its peers. Data Exfiltration

- This third category combines data theft alongside encryption for a two-pronged approach. This exfiltrated data is often used as leverage for future ransom demands to stop potential data breaches. Indeed, because of its efficacy this variant has an astounding 66% higher ransom payment rate [11], and an average payment that is higher $1.2MM, that is really a reminder of the power of this one.

This paper specifically focused on the first type referred to as Encryption.

Machine learning for detection ransomware that exploits the vast information available in Portable Executable files can be detected by analysing them using machine learning techniques windows operating systems use PE files as the standard format for an executable file and it wraps around immense metadata and structural parts ranging from file header with information all the way down to section attributes, imported libraries, strings, enriched elements and many others.

A major benefit of maintaining a PE file perspective is improved early-stage ransomware artifact detection, where the analysis is vital, especially for insider threat scenarios.

This type of profiling for certain encryption related features allows the detection and blocking the execution of malicious software BLOBs (Binary Large Objects, a data type to store unstructured data) before the ransomware even has the chance to encrypt potentially damaging files on systems where insid- ers maintain legitimate but suspicious access. In addition, it makes the ransomware detection system more sensitive to the behaviors and patterns of this threat by tuning them by their unique features to detect indicators based on encryption. This precise strategy can improve more ultimate results of finding the ransomware infections correctly, even the assault strategies evolve.

1.2 Attack Stage Comparison

Outside ransomware attack usually targets public-facing sys- tem with the typical attack progression, while insider-enabled attacks use trust to already have internal access to the tar- get [12]. However, this all happens once a ransomware actually starts running on a system where there is installation, command server's communications, data encryption/exfiltration, ransom demands, and restoring the impact as illustrated in Figure 1. This means the Insider-enabled attacks can skip through initial points for defense and quickly progress to dam- aging encryption. The attack stages from the nine approaches are provided more in detail in Table III, which provides an exhaustive comparison from a scientific and professional perspective.

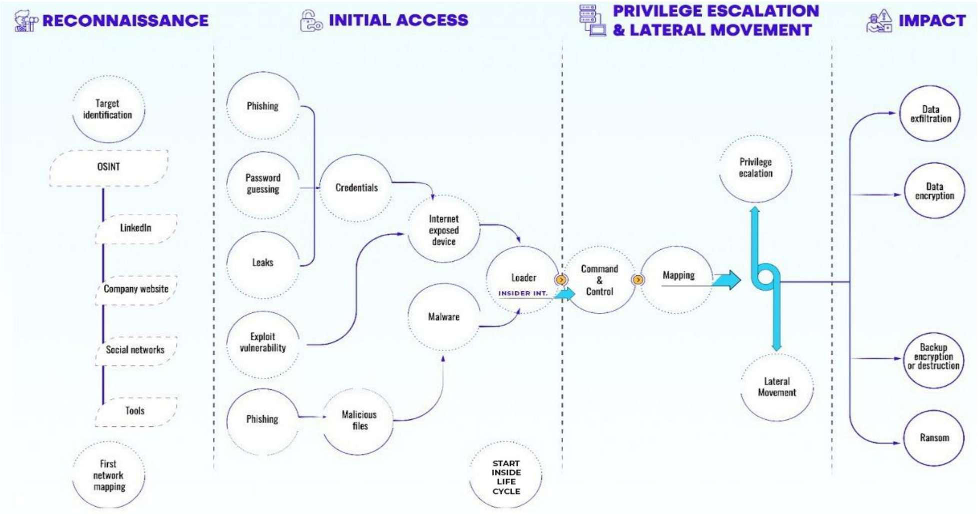


Figure 1. Life cycle of Ransomware - Outside Vs insider

Although the ultimate objective remains uniform (i.e. to cripple operations and financial resources by means of encrypting data and demanding extortion), the attack progression and tactics employed by these two variants exhibit marked differences. Table 3 provides a high-level comparison of these two types of ransomware attacks. First type is the outside attack which orchestrated by highly skilled attackers that begin their efforts by first compromising a vulnerable supply chain gateway, then uses the initial attack vector to get initial reconnaissance into the target enterprise.

Table 3. Attack Theatres Comparison - Outside Vs Insider Enabled

Stage	Outside Ransomware	Insider Ransomware
Initial Access	Phishing users, exploit apps	Directly install themselves
Installation	Escalate privileges quietly	Rapid execution across accessible systems
Impact	Identify and encrypt critical data	Additional data exfiltration beyond encryption
Extortion	Display ransom payment instructions	Conceal forensic evidence after deployment

After that, the attackers very carefully steal their way into the critical systems and backup repositories in order to locate and encrypt the critical data stores, then a ransom note with payment directions is defiantly presented, making victims feeling helpless.

The second type, insider-enabled ransomware attacks capitalize on the credibility of trusted insiders to avoid the occurrence of the initial stages of infiltration. These attackers could be the ones to install the ransomware payload directly, and with full access to the systems they have authorised, they can quickly carry out the encryption procedures. More nefariously, they may exfiltrate more high-impact documents, increasing the extortion force. This can further obfuscate who the attackers are, as they employ the partial forensic clean-up to not only cover their tracks, but also make it more difficult to identify the victims and recover from the successfully completed ransomware operation.

The research aims to detect insider ransomware using ma- chine learning techniques for inspecting metadata and features in an executable datasets provided by [13]. It is comprised of over 130,000 Windows executable (PE) files, labeled as benign software or ransomware samples using characteristics such as headers, sections, import strings, among others. New executables can be scanned for internal attempts to run potential malware by training machine learning models with this dataset to identify ransomware attributes.

We are working to utilize state-of-the-art ML algorithms to insider attack scenarios, which gain better results and quickly. We use supervised learning algorithms Random Forest, XGBoost classifiers, decision tree, GBM, LightGBM, KNN, AdaBoost, Logistic Regression, SVM. With a hybrid model combining two supervised learning algorithm. More- over, ransomware detection technologies could be integrated with host-based defense and security systems, that greatly improving enterprise resilience. The early warning will prevent the execution of ransomware encryption and exfiltration.

The contributions of this research aims to achieve:

1- This work assumes a realistic context for ML model training to generate results corresponding to real world cases. Following that, we aim to understand these basics and focuses on what is different from training in standard ML settings.

2- Develop and investigate the process of detection of an In- sider Ransomware Attack using data collection, preprocessing and ML model based data analysis

3- To have an in-depth result reporting process in terms of instance and PE file-based results, which allows determination of malicious incidents in order to improve the understanding of insider attack scenarios.

This is, to the best of our knowledge, the first paper that addresses regarding measuring the performance of these ten Machine Learning algorithms (Random Forest, XGBoost classifiers, Decision Tree, GBM, LightGBM, KNN, AdaBoost, Hybrid model, Logistic Regression, and SVM) in terms of classifying ransomware insider attacks in order to leverage Machine Learning algorithm performance to immediately dis- cover the right security protective tools to enhance the level of security protection. Recent literature on ransomware insider threat detection and classification used various models as well as ensemble techniques. Each one of those studies applied the models on different datasets, which leads to different classification results.

The research structure is organized as follows: Section II explains related work. Section III provides the proposed methodology. Section IV is a representation of the applied dataset, experimental setups, evaluation metrics, and results. The discussion is given in Section V and the conclusion is in Section VI.

2. Related Work

Insider threat analysis has been studied for many years using machine learning on properties and metadata extracted from windows portable executable files.

Machine learning models can be trained to accurately identify and detect the characteristic patterns and behaviors of ransomware attacks if a new executable that a user launch has the hallmarks of known dangerous programs characterised by those models and appropriately identified, it can be stopped in its tracks, not allowed to cause damage. This review presents an investigation of prior studies targeted at machine learning in ransomware detection for Windows PE files with static feature extraction. The paper presents the approach, results, constraints, and elaboration for enhancing defenses against ransomware-enabled insider threats. Machine learning with static analysis of Windows PE for Ransomware detection research is a continually evolving space with some initial works establishing base modeling pipelines, feature sets, and datasets for evaluation. Nevertheless, there is still lack of evaluation rigor for real-world attacks as well as the specific feature for detecting insider threats. The original works are initial endeavors in using machines to detect ransomware via static analysis of Windows PE files, and it is an ongoing research problem with people consolidating efforts to create modeling pipelines, feature spaces, and testing datasets. But as it stands, it remains a hinder behind in terms of rigorously evaluating the ability against samples from real-world attacks and specialize to detect insider threats.

The PE metadata in [14] was used to produce attributes for training on random forest (RF), support vector machine (SVM), and k-nearest neighbors (KNN) classifiers. They achieved over 90% accuracy on identifying ransomware variants. This presents a good base, though the research did not refine the modeling for insider conditions.

2.1 Static Analysis Approaches for Ransomware Detection

Static analysis examines ransomware executable files with- out executing them, with a focus on the malware's structural and syntactic properties. In this part, an exhaustive examination on different static analysis techniques used in ransomware analysis has been illustrated. Static analysis is concentrating on the internal characteristics of the Portable Executable file format which involves header, section attributes, and imported functions. Processes such as PE header analysis and section analysis interrogate these file structures for irregularities. In addition, analysing imported libraries and strings along with opcode sequencing can provide a more depth insight about ransomware operational mean.

1) PE Header Analysis: The researchers in [15] analysed the features extracted only from Windows portable executable headers, i.e., machine types and optional header properties. Using only this metadata, they trained a classifier a convolutional neural network (CNN) to distinguish between public and private ransomware samples, with 92% accuracy. However, by introducing more pronounced features and adjusting the tuning for insider threats could increase detection performance. A number of researchers have attempted to move forward the state of the art detection of ransomware, by analysing Windows PE headers. Notably, [16] presented up with yet another typical hybrid feature set that contains static PE features share along with the behavior indicators. One of their unique methods that utilizes the deep learning method and has an accuracy rate of 94.7% in detecting ransomware over cryptocurrency samples. Furthermore, [17] addressed the use of ensemble learning models fusing PE headers features, and entropy-based attributes. Their thorough experimentation proved that this approach could offer powerful performance, achieving an amazing 97.2% F1-score in identifying ransomware threats.

2) PE Section Analysis: The authors in [18] studied entropy properties of PE file sections in their

uncooked form and fed them into gradient boosting models to identify ransomware. On an open-source malware dataset, they achieved an 83% precision rate and recall of 81%. Yet, if they specifically focus on irregularities in the execution contexts of insiders, we can perhaps forestall false positives eventually. To the best of our knowledge, [19] pioneered such feature engineering which aligns PE section entropy-based attributes with opcode sequences. By using deep neural networks in a novel way, they demonstrated 94.3% detection in ransomware version. Moreover, [20] proposed a sophisticated multi-task learning framework utilizing a joint optimization of a PE section classification and a relative strength of entropy value prediction that showed a better generalization ability to the diverse ransomware families, achieved 83% precision and 81% recall on an open malware dataset. However, focusing only on anomalies that occur in insider execution contexts may help to decrease the number of false positives

3) Imports & Analyzing Strings: In [21], specific features were chosen from PE imports, strings and general properties to formulate Random Forest and Deep Neural Network models, that achieved 98% cross-validation accuracy on private ransomware corpora. skill real-world testing is necessary to mitigate against the risk of inflating performance estimates due to the high degree of overlap between the training and test feature distributions.

Few researchers have estimated to utilize PE imports and code-string in order to detect ransomware by using machine learning concepts. Same of theme are, [22] who integrated import features and the dynamic API call sequences, using long short-term memory (LSTM) networks to classify. The authors' method scored an ACC 96.8% detection accuracy for ransomware threats. Furthermore [23] developed a robust ensemble learning methodology leveraging PE imports, strings and entropy based attributes for higher generalization ability across a variety of ransomware families.

4) Opcode Sequencing: The researchers in [24] designed opcode sequences with Convolutional LSTM Neural Networks for classification of ransomware API call graph behaviors. The method achieved 92% accuracy and 97% area under the receiver operating characteristic curve (AUC) using a high-volume OpenML dataset. This approach may be further improved by explicitly taking into account insider execution flows. In [25] the authors introduced a new method that merged opcode sequence analysis and static PE file feature, and used transformer based model for classification. They reported a 97.3% accuracy for ransomware detection with their approach. Furthermore, [26] presented a complex multi-task learning pipeline that simultaneously learns the notion of discriminating opcode sequences as well as the ability to predict their corresponding control flow pattern, reaching an overall generalization performance gain over a spectrum of ransomware families.

While in [27] the researchers made a very important contribution in this area, they referred to it as their seminal work, Investigated the use of adversarial training methods for improving model robustness in ransomware detection models based on opcode-sequence. They provided evidence to show that training models with additional adversarial examples alongside normal examples enhanced the norm models' robustness against evasion attacks, as their final model was capable of accurately classifying 95.8% of diverse, real-world ransomware samples.

Building on this research, Xu in [28] suggested a new approach that uses graph neural networks to model the structural relationship between opcode sequences and the corresponding control flow graphs for detection of the NOT detectors. The combination of static and dynamic characteristics as well as the conversion of ordinary features to intelligent features was able to detect ransomware threats at 96.2% of the accuracy rate, which outperformed the traditional approaches in which each characteristic was used and analyzed entirely in itself.

The combination of static indicators with behavioral data as well, such as sequence of API calls and process monitoring, has also demonstrated effectiveness for ransomware detection. [29] utilised Windows event logs for suspicious API activities and achieved a 95% true positive rates of detecting ransomware attacks. However, the false inspection was not particularly useful, in part due to a high false positive rate of 18%, rendering it impractical to deploy without the need for tuning aimed at identifying insider threat.

In a study by [30] on Markov chains over dynamic Microsoft Office application calls for detecting sequences aligned with ransomware encryption routines correctly identifies 82% of incidents. However, this type of behavioral data is often noisy and difficult to run at scale across enterprise environments.

The researchers in [31] proposed a complex ensemble learning mechanism integrating an API call sequence with process activity indicators providing better generalization power over varied ransomware families. Even further, [32] introduced a new system that uses static PE features along with dynamic API call sequences for classification using deep learning methods. Their innovative approach managed to achieve a remarkable 97.1% precision in predicting ransomware threats for Android systems.

2.2 Dynamic Analysis Approaches

The purpose of dynamic analysis is to better comprehend malware activity in real time. This further allows models based on sequences to accurately model API calls predictability so as to identify malicious activities better. Hybrid models that integrate PE file attributes with API sequence data provide a comprehensive approach by combining static and dynamic techniques, resulting in more robust detection mechanisms.

1) Graph-Based API Modelling: In [33] the researchers built API call graphs in sequential form and designed graph neural networks to identify ransomware execution patterns. Their method achieved an F1 score of over 90% on private dynamic datasets. But logs about functionality from various environments created noise and degraded performance.

Notably, [34] initiated a method as combined API call graphs with control flow information, and then applied graph convolutions for classification. Using a novel method, they were able to gain 95.2% accuracy in identifying ransomware threats.

Furthermore, [35] constructed multi-view learning frame- work combining API call graphs with dynamic process behavior indicators, and displayed superior generalization capabilities against variegated ransomware families.

2) Process Behavior Clustering: With the help of isolation forests, the research done by [36] categorized ransomware behavior using clustered process behavior graphs according to file, registry, network events and properties design, reaching 85% recall with low false positives. Adapting feature selection and modeling to target on risky insider actions can specifically improve detection utility of this threat vector.

In a landmark paper by [37], they introduced a novel technique combining process behavior graphs with deep learning methods, using graph neural networks for clustering and classification. Their novel approach attained an impressive 93.7% accuracy in detecting ransomware threats. Moreover, [38] clarified that with this sophisticated ensemble learning frame- work, which combines static PE features and process behavior indicators, the proposed approach shows better generalization performance across a wide variety of ransomware families.

3) RNN API Sequence Data Analysis: In [39] modeled Microsoft Windows API sequences during application run- times using recurrent neural network (RNN) architectures. Using their method, the technology achieved greater than 95% detection accuracy on real ransomware relative to traditional AV solutions.

In [40] have researched API sequence by RNN to detect ransomware. With their unique approach, they managed to nail an impressive 96.5% accuracy rate in ransomware detection. Furthermore, [41] introduced a gradually complex multi-task learning structure that co-optimizes API call sequence classification and behavior pattern prediction that was reported to increase SCRAP generalizability across numerous ransomware families.

4) Hybrid PE & API Models: In [42] combined embedding of API call graphs with static Windows PE metadata such as imports. Their approach resulted in 91% detection accuracy for some private ransomware samples; However, they mainly focused on lateral movement of the ransomware (i.e., the same as the beacon) as well as the use of a beacon, thus paying less attention to insider threats.

A novel approach which uses a fusion of PE metadata and API call graphs powered by attention-based neural networks proposed by [43] is an important work in this line of research. Iterating on this approach, they devised a new methodology that boasted a remarkable 95.8% accuracy when it came to predicting ransomware threats. In addition, [44] proposed an advanced ensemble learning framework combing the PE imports, sections, and entropy-based features with dynamic API call sequences which can outperform the existing methods for generalization on different ransomware families.

When studies score their models, they do so using synthetic data or data simulations and not on true ransomware samples. Therefore, the lack of public data can restrict how widely we can generalize our findings to new and “next-gen” ransomware threats.

With an extensive review of the literature on ransomware detection in machine learning, including all other literature on facets of ransomware detection in machine learning, this paper intends to build a solid base. Consequently, the incorporation of recent and diversified research contributions becomes significant to emphasize both the sustained efforts and the wide variety of pathways being investigated to deal with this pressing issue. The in-depth analysis is designed to illuminate possible shortcomings in such approaches and open a roadmap for future work, thereby playing a part in shifting towards more advanced and efficient ransomware defense mechanisms, especially those targeting insider threats.

3. Research Methodology

The malicious insider, as a result of increased access and opportunity can now become an existential threat to the organization. Insiders enjoy a privileged and legitimate access to in- formation and resources which outsiders do not carry. Insiders are also familiar with key targets of the business, therefore good categorization helps in identification and comprehensibility of insider attackers' priorities to classify serious internal threats. Insider risk can be identified and mitigated depending on the insider indicators, detection strategies or even what kind of an insider going rogue. Intentional and unintended insider threats attack on the information and the use of unauthorized activities to affect the information’s availability, integrity, or secrec which are examples of misuse actions. The threat approach determines the method for detecting malicious agents. Attackers might easily introduce random data into the distributed algorithm to prevent it from convergence [45].

Traditional machine learning suffers from limitations in attack detection; the algorithm is unable to automatically design features, its low rate of success for detecting, and inability to identify tiny mutants of known attacks and insider attacks [46]. In most cases, the ensemble of models will perform better than an individual model for insider threat detection and classification.

The proposed method achieves model consistency and robustness. Where ensemble models capture both linear and non- linear connections in the data. In order to do so, we will have two separate models that are combined into one model. This system consists of nine supervised machine learning and one hyper models.

These models have been implemented in a series of steps such as Database Pre-processing, Model Training and Testing, Detection and Classification.

3.1 Collecting and Pre-processing the Dataset

We conducted our experiments using a carefully constructed test environment to capture statistically significant data for both ransomware and benign Windows portable executable (PE) files. This environment would be controlled to ensure the collected samples were representative of what organizations actually see in their day-to-day defenses against insider enabled ransomware attacks. The dataset consisted of 138,047 PE file samples with a total volume 45,916 are ransomware and the rest as benign executables. The samples were collected from different trusted sources such as malware repositories, live enterprise deployments, and sandbox execution environments. This broad sourcing helped to make the dataset as rich and diverse as possible with a large number of ransomware families, variants, and obfuscation techniques in use by mod- ern attacks, further boosting resilience and generalizability results on our approach.

Prior to features could be extracted and the machine learning models trained, an important preprocessing task was completed, this involved extracting metadata properties of interest from PE file samples. There are several features with missing values and outlier. The column ‘SizeOfOptionalHeader’ in the dataset had a few outliers, these were removed by taking an average of its neighboring values. The feature ‘MinorLink- erVersion’ had some missing values. Then the dataset pattern of that feature fills those missing value. A data of good quality is essential for best performance. If we go back to the dataset there were also irrelevant features present which had an impact on how model is being trained. Thus, the objective is to remove irrelevant features (’SectionsNb’ and ’Sections MinEntropy’) as shown Figure 2, and train them models on only selected feature spaces.

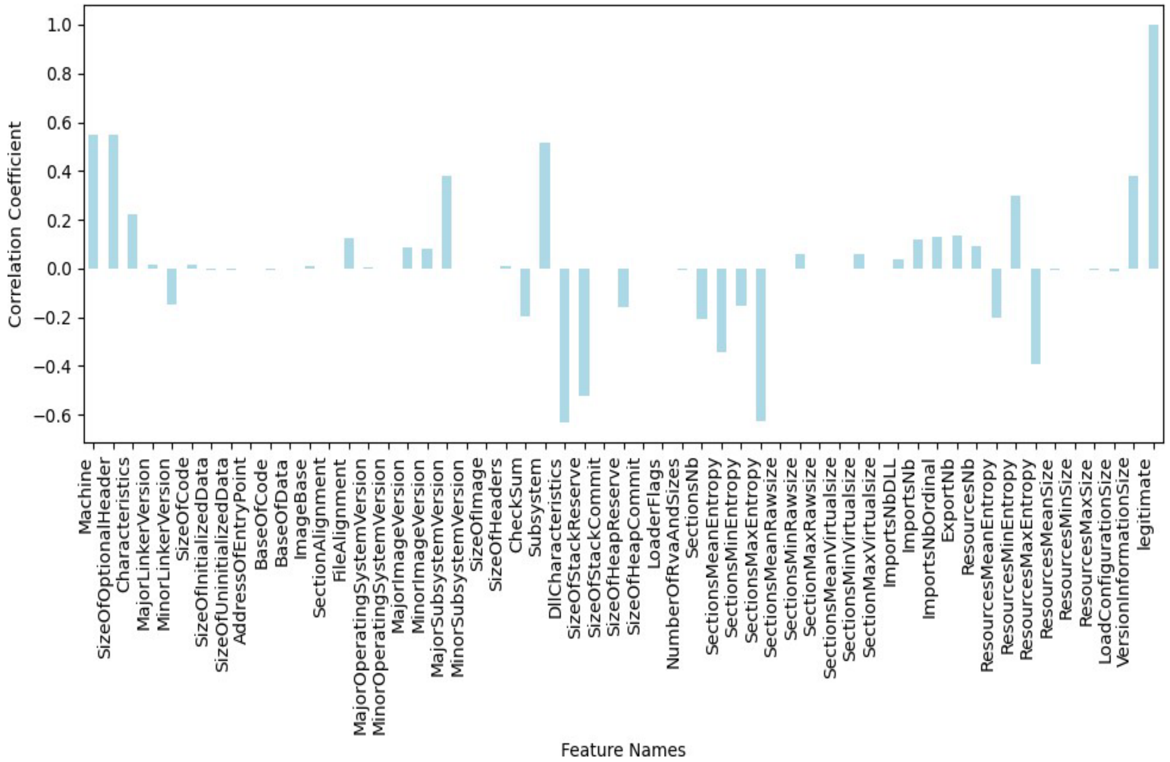


Figure 2. Correlation coefficients with legitimate label

3.2 Machine Learning Modeling

In order to effectively classify malware and benign software samples, a range of supervised machine learning techniques were used. For evaluating the performance of these models accurately, the dataset was split into training and testing sets. Supervised learning algorithms were trained on fully labeled data with the target attribute "legitimate" provided in order to determine which patterns and relations existed between features and class labels.

1) Random Forest: The basis of the Random Forest algorithm is on bootstrap aggregation (bagging) and random feature sub spacing. This process of introducing randomness by building multiple bootstrap samples from the training data is performed in bagging with random sampling without replacement. A decision tree is trained for each sample [47]. In Random Forest decision tree construction, in fact is bootstrap sample by iteratively choosing the best feature-as depicted in Figure 3 and split point according to Equation 1, and based on criteria such as information gain or Gini impurity.

$$G = 1 – \Sigma(pi)2, \qquad (1)$$

where G is the Gini impurity and pi is the probability of being in class i. That is repeated until some stopping criterion. Another way to randomize trees is using randomly selected subset of features for splitting at every node as well, this method can be very useful in practice especially when we have high dimensions and redundant columns being a similar or concise data. Typically, the square root of the total features for classification impurity is then calculated by using Equation 2:

$$m = \sqrt{} M, \qquad (2)$$

where m is the number of selected features and M is the total. After the trees are built, Random Forest aggregates pre- dictions. Majority voting for classification, class label impurity is calculated using Equation 3:

$$C\hat{}(x) = \text{majority vote}(Cb(x)), \qquad (3)$$

where C^(x) is the predicted class label for instance x, and Cb(x) is the prediction out of b-th tree.

Ensemble prediction for regression is the expected values of individual tree predictions.

$$f\hat{}(x) = 1 \times \Sigma f(x), \quad (4)$$

The predicted value is then given by f^(x), fb(x) is the prediction from the b-th tree, and B is the total trees.

Training will be much faster for large data sets since a lot of tree building can be done in parallel. Given this potential strength in handling the complex and non-linear aspects of real-world ransomware distributions, Random Forest is established as one of the most preferred choices for dealing with ransomware when it comes to detection or classification [48]. Figure 3 shows Stepwise Working for insider threat classification using Random Forest, the set of dataset is split into the subsets, and then for each subset a random classifier makes decision tree. Each decision tree would predict an output. Finally, it performed majority voting on all the decision tree outcomes. The best guess is that the survival rate would be one of the most probable results for decision trees.

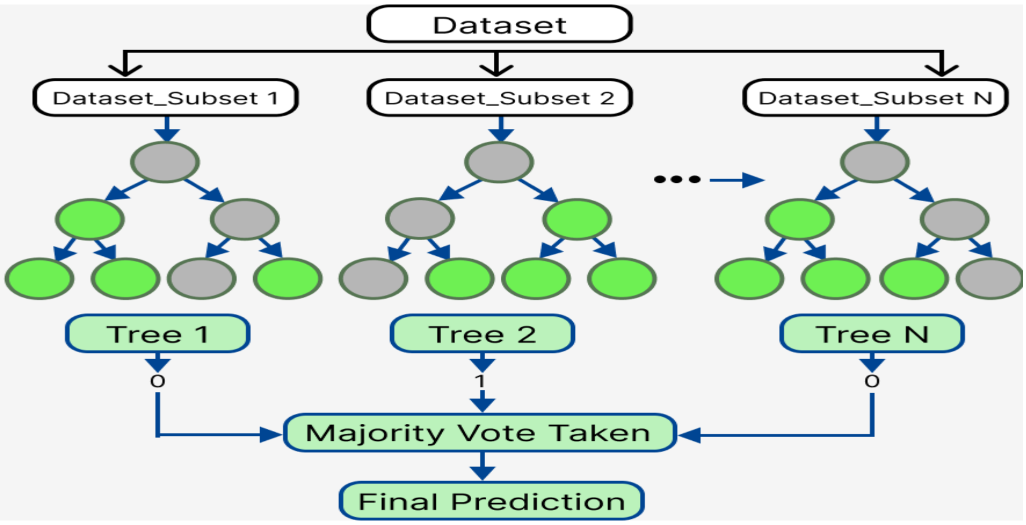


Figure 3. Random Forest Classifier

2) XGBoost: XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It benefits in the function of changing model and tweaking it. It is also useful with Regression, classification [49]. In this approach, trees are grown one after the other. In XGBoost, the weights are very important. Disct is then introduced into the decision tree and gives weights to each independent variable that helps it in predicting outcomes. We call it less time consuming than Boosting. To deal with incomplete data, AdaBoost has built in capabilities.

After every loop, they can then perform a process of cross validation. It is fit for Small to Huge datasets. The similarity scores are calculated using two equations and the new residuals. We use Equation 5 to find the similarity score and then Equation 6 for new residuals which will be taken as input in next iteration of this algorithm.

$$Similarity\ Score = \frac{(Gradient)^2}{\eta Hessian + \lambda} \qquad (5)$$

$$\text{NewResiduals} = \text{Oldresidulas} + \text{pPredictedResidulas.} \qquad (6)$$

The XGBoost is very fast, scalable and portable-gradient Based Boosting library developed by Tianqi Chen. This utilizes Gradient Boosting framework for building machine learning algorithms. From the results of Figure 4, XGBoost for classification on dataset works. XGBoost grows new trees by repeatedly cutting features. Actually, each time it takes a tree, it finds a new function to represent the residual prediction of the last prediction. Any set of prediction data (where K trees were created in training) will share a matched leaf node for each tree so there should be corresponding scores with respect to every leaf points. Finally, the matching scores from each active tree are combined to get a recognition prediction value for that sample [50].

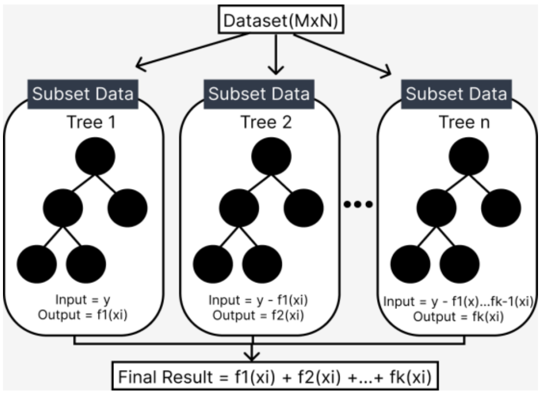


Figure 4. XGBoost Classifier for Insider threat classification

3) Hybrid Model: Ensemble techniques can be used to combine the predictions of several algorithms into a single model, one interesting approach is to try constructing a hybrid model that leverage the strengths out off multiple models. A hybrid model, such as an integrative random forest and xgboost to majority voting ensemble strategy of the outputs by both Random Forest and XGBoost classifiers expressed in Equation 7 could be implemented:

Cˆ(x) = majority vote(C rf (x), C xgb(x)) (7)

Cˆ(x) refers to the predicted class label for instance x, where C rf (x) = Random Forest model prediction and C xgb(x) = XGBOOST model prediction.

Most voting schemes take the prediction of individual models and produces them a better predictive performance to detect ransomware malware as well classify them.

These hybrid models can be very helpful for example the ransomware due to its complexity, as it usually gets treated with two different ML algorithms which capture a more various set of patterns and features. Combining the strengths of different algorithms, largest solution can be done with Random Forest as it can handle high dimensional data, XGBoost is known scalable and resistant to overfitting.

4) KNN model: k-Nearest Neighbors (KNN) algorithm is a non-parametric method that assign labels to an unknown point by finding the k closest data points in feature space, then voting on majority [51].

The algorithm coverage of the prediction can be written as in Equation 8.

Cˆ(x) = majority vote(Ci), where i = 1, 2, ...k. (8) Here Cˆ(x) is the predicted class label for instance x and Ci means *i*th neighbour classes among k considered.

They can be particularly powerful when we assume that sides of the ransomware and benign software data share strong patterns which effectively differentiate between two objects in the feature space, essentially building a classification function based on how similar an investigatory instance is to labeled instances.

This would be followed by KNN models trained on a range of features from known ransomware and benign software samples (e.g., SizeOfOptionalHeader, and CheckSum data LoaderFlags), in the context of ransomware detection. By comparing the feature representations of new unseen samples and tracing back to see how closely they resemble their nearest neighbors from the labeled training set, KNN can justifiably categorize them as Malware or non-Malware.

Despite its simplicity, KNN is a powerful tool in ran- somware detection, particularly when combined with feature engineering techniques and ensemble methods to enhance its performance and robustness. Additionally, KNN's ability to adapt to new data by incorporating labeled instances into the training set makes it well-suited for the dynamic nature of the malware landscape, where new threats are constantly emerging.

5) GBM (Gradient Boosting Machine): Gradient Boosting Machine, is an ensemble learning algorithm that consists of a large number of decision trees in a stage-wise (iterative) manner with an aim to correct the errors made by previous trees [52]. The core principle of the algorithm can be embodied by Equation 9.

$$F \quad m(x) = F \quad \{m - 1\}(x) + \eta \times h\ m(x), (9)$$

where F m(x): ensemble prediction at m-th iteration, F m 1 (x): previous ensemble prediction, η: learning rate, and h m(x): weak decision tree model at m-th iteration.

GBM is iterative in nature and is able to represent complex non-linear relationships. It can be an appropriate solution for the ransomware detection domain since features extracted from samples of ransomware used may have very intricate patterns and dependencies. GBM is the ideal candidate for generating an ensemble of weak decision trees and suitable at identifying these complex patterns by learning from data, which helps in separating malware samples from benign software.

Moreover, the resistance to overfitting of GBMs and their adaptability for large number/ high dimensional data work particularly well in malware analysis, where feature spaces can be highly convoluted or laden with noisy/ irrelevant features.

6) The Logistic Regression: The logistic regression technique is a classification algorithm in statistics used for binary classifications that models the probabilities of belonging to one class among other classes by setting up a clear decision boundary based on creating a curve which depicts whether an instance submitted will belong to any certain class or not [53]. The basic mathematical representation of the core equation is Equation 10.

$$\frac{1}{(y = 1|x) = (1 + \exp(-\beta T\ x))} , \qquad (10)$$

where P (y = 1 x) is the probability of an instance x belonging to the positive class. β is a column vector of coefficients. x is a column vector of features. βT is the row vector obtained by transposing β. So, βT x is essentially the sum of the products of corresponding elements of β and x. It may not be so bad to use when classes are almost separable in feature space (suitable for solving malware issues or using as more complex/machine learning way).

7) Support Vector Machine: Support Vector Machine (SVM) is a powerful supervised learning algorithm for classi- fication and regression tasks. SVM creates a hyperplane or set of them in high-dimensional space to separate the classes [54]. Equation 11 explains the prediction for a new instance x:

$$y = \mathrm{sign}(w^T x + b) \tag{11}$$

Here w is the weight vector perpendicular to the hyperplane, x is the input instance and b provides bias term. The algorithm of SVM tries to find the best value for w and b that divides those classes with a maximum margin.

In the context of malware detection, SVM is used to learn on different forms of features, both static and dynamic, that can be collected from malicious or even benign samples. Some of these features include but are not limited to byte sequences, opcodes, control flow graphs, API call patterns or behaviour attributes. Mapping these features to a high dimensional space would allow SVM algorithm to learn decision boundaries which will segregate the malicious instances from benign ones adequately.

SVM provides a series of benefits in malware analysis including the capacity to work with high dimensional data, robustness against overfitting and flexibility for using different kernel functions that can capture more complex relationships between features. SVMs are also blessed with a principled way of accounting for imbalanced datasets, commonly observed in malware detection tasks.

8) Decision Tree Classifier: Decision Tree is a non- parametric supervised learning method used for classification and regression tasks. As they partition the feature space recursively into smaller and smaller regions, learning decision rules are based on increasingly discriminative features [55]. When given a new instance x, the model predictions are made by walking down from the root node to leaf nodes of decision tree following particular decision rules established in internal nodes. The strengths of decision trees for malware analysis include their interpretability, ability to handle both numerical and categorical data robustly much like xgboost does. Al- though Decision Trees will not always be the performance king, they may still serve as a powerful baseline and expose which features are very descriptive for classification tasks such as malware detection. The interpretability of decision tree classifiers can help in visualizing overall pattern and infected features ensuring compatibility between the malicious code samples.

4. Proposed Methodology Overview

The proposed model for classification of ransomware insider attacks is outlined in Figure 5. This work is using a customized dataset of the MalwareData, obtained from multiple files. On this dataset, then we apply machine learning algorithms Random Forest, XGBoost, Hybrid Model output performances of the other Models KNN model, Gradient Boosting Machine, Logistic Regression Vehicle, AdaBoost, LightGBM, Support Vector Classifier and Decision Tree Classifier give better results. The main aim of this paper is to present a framework that could be implemented for the detection and classification of an insider attack for ransomware.

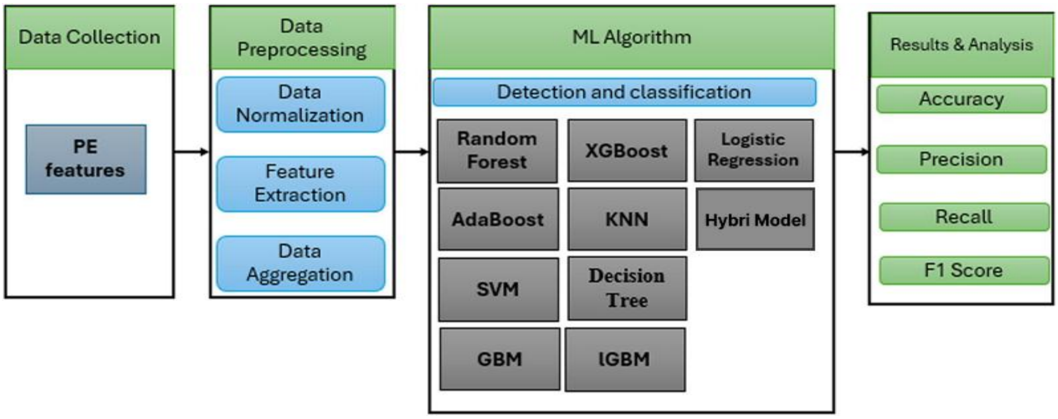


Figure 5. The proposed model for classification of ransomware insider attacks

Ten different algorithms were applied to the MalwareData dataset in the proposed model. Mathematical and technical analysis are pre-defined in sections above and implemented models have both equations and technical details. The maingoal of ensemble learning is all about improving the performance of a model. Bagging, boosting, and stacking are examples of ensemble learning. In this paper, we have applied bagging and boosting techniques for the detection of ransomware insider threats. This involves aggregating data during the pre-processing phase in order to identify what type of information is useful for our models. During the data preparation process, it is one of the most useful techniques that we use to transform into comparable scale and it is called as Data Normalization. Therefore, the model does better by training in a more stable method. Because of its huge collection of unnecessary data at the time of application, we need a data extraction process to ultimately reduce the size, which make the data reduction process quicker, both the learning phase and generalizations with machine learning fast as well, that requires lesser machine to build a model. Boosting is an ensemble learning technique that combines a number of 'base learners' into strong learner, in order to decrease training errors. Therefore, in this context we use boosting which is a technique from ensemble learning.

Python is one of the most popular computer languages and it has supplanted many other rival languages in this field primarily due to its vast library system. We have implemented the following best Python libraries as shown in table 5 in the proposed paper.

Figure 6 explains the flow of proposed technique as data is retrieved from the dataset and preprocessed. The data is trained using the machine learning models, and testing is performed based on the ratio of the dataset.

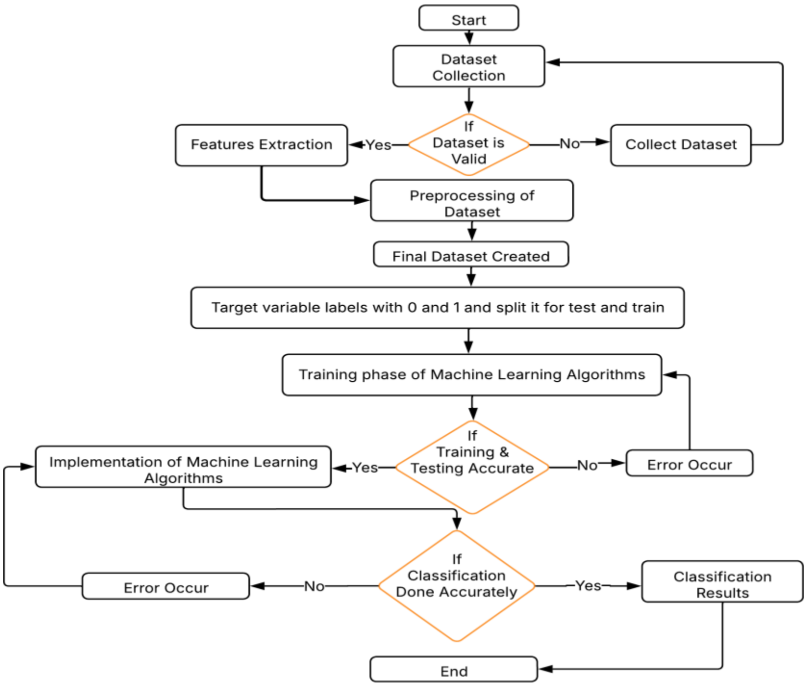

Start
Dataset Collection
If Dataset is Valid
Yes
No
Features Extraction
Collect Dataset
Preprocessing of Dataset
Final Dataset Created
Target variable labels with 0 and 1 and split it for test and train
Training phase of Machine Learning Algorithms
If Training & Testing Accurate
Yes
No
Implementation of Machine Learning Algorithms
Error Occur
If Classification Done Accurately
No
Yes
Error Occur
Classification Results
End

Figure 6. The workflow of proposed technique approach

5. Performance Evaluation

The experiments are done on a Windows 11 operating system, on 12 rd generation Intel Core i7 processor and 32 Gb processed, and applied by proposed models for better results generation.

A. experiments Setup

Table IV displayed the experimental parameters considered while the classification using machine learning methods. The main parameters learning rate, n estimators, max depth, min child weight, subsample, and colsample bytree helped to improve XGBoost's accuracy. The main parameters that raise the random forest classifier's performance are estimators, random state.

B. Evaluation metrics

In this context, the evaluation of machine learning models involves a number of metrics, including accuracy, precision, recall and F1-score.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (12)$$

$$Precision = \frac{TP}{TP + FP} \qquad (13)$$

$$Recall = \frac{TP}{TP + FN} \qquad (14)$$

$$\text{F1-Score} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \qquad (15)$$

of RAM. In this work, the MalwareData dataset is utilized. This dataset includes many features for the detection and classification of insider threats ransomware. All the files in that dataset are in CSV format, so it is easily analyzed, pre Where, TP indicates true positive, TN indicates true negative, FN used for false negative and FP for false positive.

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} \qquad (16)$$

Table 4. Experimental Parameters for Proposed Models

Model	Parameters
Random forest	n estimators: 100 random state: 0
XGBoost	learning rate: 0.05 n estimators: 50 max depth: 5 min child weight: 2 gamma: 2 subsample: 0.8 colsample bytree: 0.7 objective: ‘binary: logistic’ nthread: 2 scale pos weight: 2 seed: 20 reg alpha: 1 num parallel tree: 5 max cat to onehot: 2
Decision tree	criterion: ’gini’ splitter: ’best’ max depth: None min samples split: 2

LightGBM	objective: “binary” metric: “auc” learning rate: 0.006 num leaves: 60 bagging fraction: 0.8 feature fraction: 0.8 bagging frequency: 6 bagging seed: 42 verbosity: -1 seed: 42
GBM	learning rate: 0.01 n estimators: 100 max depth: 5 – – min samples split: 5 subsample: 0.8
AdaBoost	n estimators: 50 learning rate: 0.5 random state: 0
KNN	n neighbors: 5 – weights: ’uniform’ leaf size: 30
Logistic regression	penalty: 12 C: 1.0 solver: lbfgs max iter: 100
SVM	kernel: ’rbf’ gamma: ’scale’ probability: True

6. Result

Ensemble learning one way to improve the accuracy of Hy- brid model by using ensemble techniques such as boosting and bagging. Boosting trains, the model with an iterative process, paying more attention to errors of previous models. However, bagging combines multiple models trained on different subsets of data.

Thus, the best accuracy on generated dataset is obtained with this hybrid model since it gives the highest accuracy of 99.49% shown in Figure 7. A RF has 99.46%, a LightGBM of 98.89%, XGBoost of 99.42%, Decision Tree of 99.10%, GBM of 98.91%, LightGBM of 98.89%, KNN of 98.77%, AdaBoost of 98.91%, Logistic Regression of 71.67%, and SVM of 98.91%. Table V displays a comparative analysis of the performance of the classification algorithms utilized in this work.

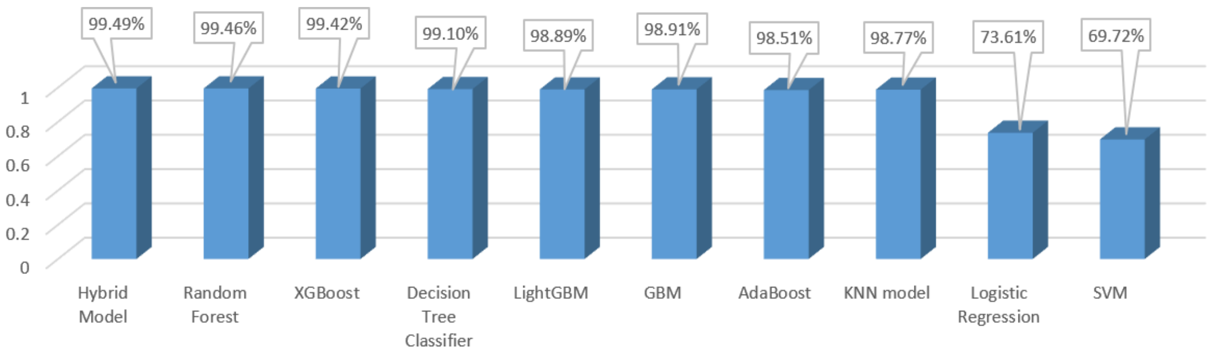


Figure 7. Accuracy of Proposed Algorithms

Table 5. Classification Results

Model	Accuracy	Precision	Recall	F1-Score
Hybrid Model	0.9949	0.994	0.992	0.991
Random Forest	0.9946	0.988	0.993	0.991
XGBoost	0.9942	0.987	0.993	0.990
Decision Tree Classifier	0.9910	0.984	0.985	0.985
LightGBM	0.9889	0.981	0.981	0.981
GBM	0.9891	0.980	0.983	0.982
AdaBoost	0.9851	0.974	0.977	0.975
KNN model	0.9877	0.971	0.988	0.979
Logistic Regression	0.7361	0.953	0.067	0.126
SVM	0.6972	0.434	0.665	0.154

The hybrid model performance is higher than the other results. While the other methods performed more accurately than in previous studies. Figure 8.1 shows the confusion matrix constructed with the classification of the SVM algorithm. It demonstrated the predicted values versus actual values. SVM predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. Figure 8.2 shows the confusion matrix constructed with the classification of the Logistic Regression algorithm. It demonstrated the predicted values versus actual values. Logistic Regression predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The Logistic Regression prediction results on the same set of data was better than that achieved by SVM.

Figure 8.3 shows the confusion matrix constructed with the classification of the KNN model. It demonstrated the predicted values versus actual values. KNN predicted most samples of the dataset correctly and thus it helps in improving the accuracy of the classifier. The KNN prediction results on the same set of data was better than that achieved by Logistic Regression.

Figure 8.4 shows the confusion matrix constructed with the classification of the AdaBoot algorithm. It depicted the predicted values versus actual values. AdaBoot predicted most samples of the dataset correctly and hence it helps in im- proving the accuracy of the classifier. The AdaBoot prediction results on the same set of data was better than that achieved by KNN model.

Figure 8.5 shows the confusion matrix constructed with the classification of the GBM algorithm. It demonstrated the predicted values versus actual values. GBM predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The GBM prediction results on the same set of data was better than that achieved by AdBoost.

Figure 8.6 shows the confusion matrix constructed with the classification of the LightGBM algorithm. It demonstrated the predicted values versus actual values. LightGBM predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The Light GBM prediction results on the same set of data was better than that achieved by GBM.

Figure 8.7 shows the confusion matrix constructed with the classification of the decision tree algorithm. It demonstrated the predicted values versus actual values. Decision tree pre- dicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The decision tree prediction result on the same set of data was better than that achieved by LightGBM.

Figure 8.8 shows the confusion matrix constructed with the classification of the XGBoost algorithm. It demonstrated the predicted values versus actual values. XGBoost predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The XGBoost prediction results on the same set of data was better than that achieved by decision tree.

Figure 8.9 shows the confusion matrix constructed with the classification of the random forest algorithm. It demonstrated the predicted values versus actual values. Random forest predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The random forest prediction results on the same set of data was better than that achieved by XGBoost.

Figure 8.10 shows the confusion matrix constructed with the classification of the hybrid model. It demonstrated the predicted values versus actual values. Hybrid model predicted the most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The hybrid model prediction result on the same set of data was better than that achieved by random forest.

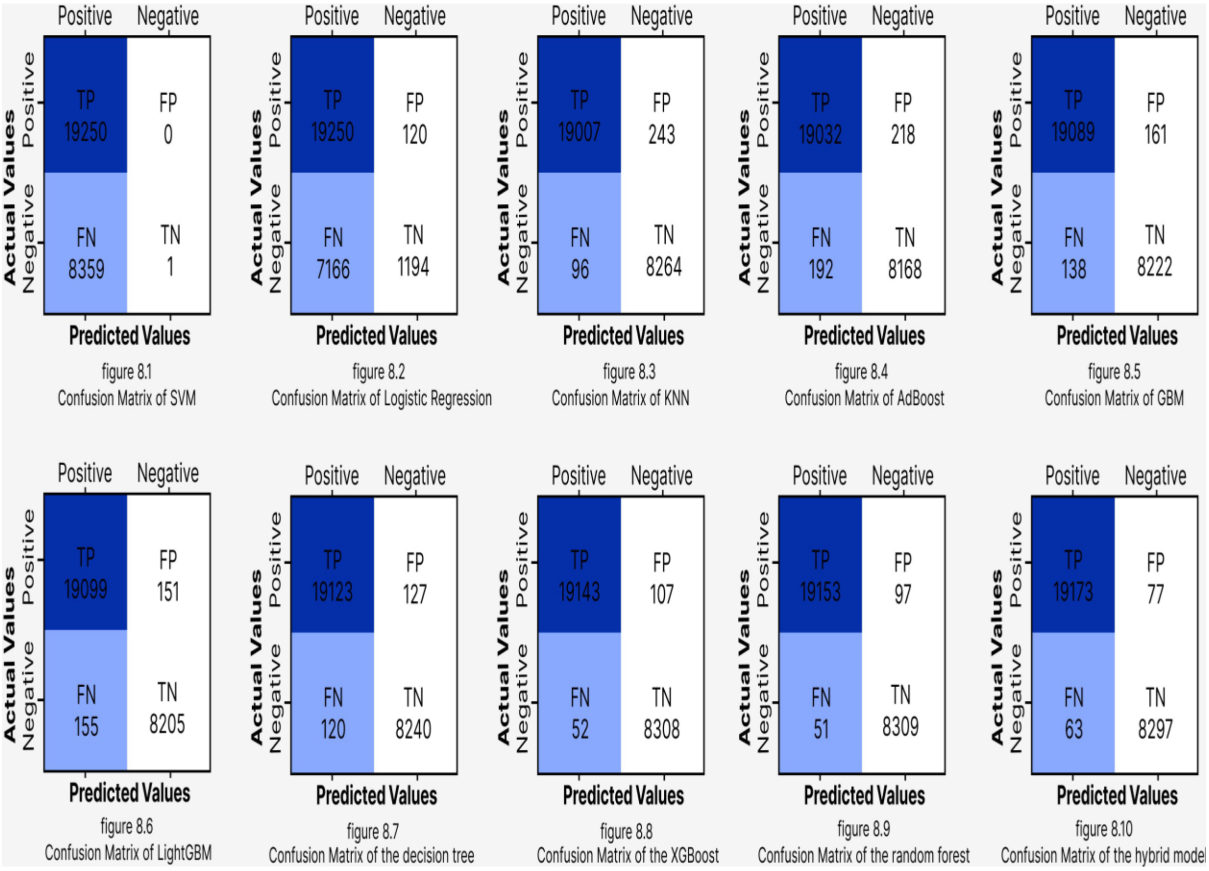


figure 8.1
Confusion Matrix of SVM

figure 8.2
Confusion Matrix of Logistic Regression

figure 8.3
Confusion Matrix of KNN

figure 8.4
Confusion Matrix of AdBoost

figure 8.5
Confusion Matrix of GBM

figure 8.6
Confusion Matrix of LightGBM

figure 8.7
Confusion Matrix of the decision tree

figure 8.8
Confusion Matrix of the XGBoost

figure 8.9
Confusion Matrix of the random forest

figure 8.10
Confusion Matrix of the hybrid model

Figure 8. Confusion matrix for all algorithms

7. Discussion

In this work, nine supervised learning algorithms and one hybrid learning have been utilized on the same dataset to per- form classification. The results show that the hybrid learning algorithm get the highest accuracy as shown in Figure 7. Furthermore, Figure 9 demonstrates multiple algorithms applied to the DataMalware customized dataset for classification. Naive Bayes (Multinomial) and Naive Bayes (Gaussian) algorithms were analyzed on the same dataset and compared with the Hybrid model. As a result, these two algorithms did not perform well.

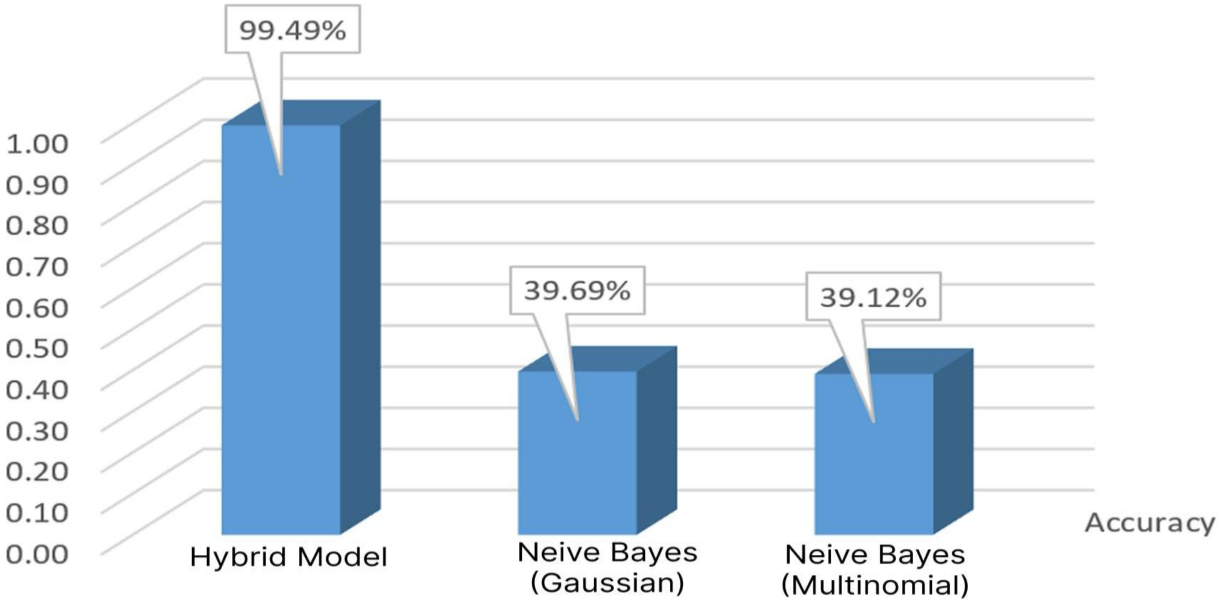
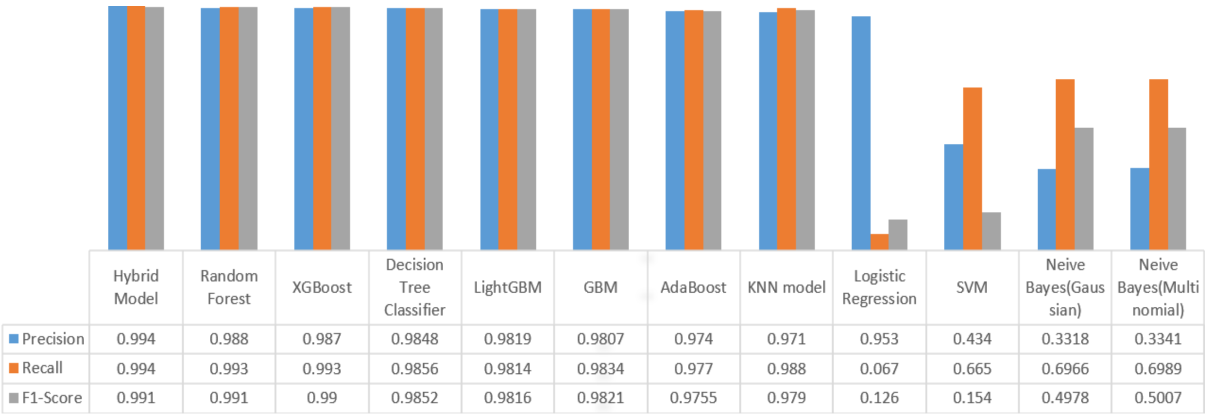


Figure 9. Different Algorithms for classification

In this paper, all algorithms were applied to the same dataset, and their comparative classification results are shown in Figure 10, which compares the proposed methodology algorithms with the literature work algorithms. Machine learning algorithms are vast, and all algorithms have their benefits and limitations. Figure 10 also shows the recall of the proposed techniques. The Hybrid model returns the most relevant results and gets the better recall value.

Moreover, the F1-score measures the accuracy of the test. It is evaluated by using the accuracy and recall as shown in Figure 10. From the bar chart, the highest value was obtained by the Hybrid model of 0.994.

Figure 11 shows the false alarm rate of the proposed techniques as calculated in equation 16, which makes it an important parameter in machine learning. Moreover, the figure indicated that the hybrid model algorithm obtained the best value. However, the disadvantage of false alarms rate is 0.004, that can lead to service interruptions and wasted time. In the context of machine learning for ransomware detection, both prediction time and training time are considered as the most critical factors that affect the overall effectiveness and efficiency of a detection system.



	Hybrid Model	Random Forest	XGBoost	Decision Tree Classifier	LightGBM	GBM	AdaBoost	KNN model	Logistic Regression	SVM	Neive Bayes(Gaussian)	Neive Bayes(Multinomial)
■ Precision	0.994	0.988	0.987	0.9848	0.9819	0.9807	0.974	0.971	0.953	0.434	0.3318	0.3341
■ Recall	0.994	0.993	0.993	0.9856	0.9814	0.9834	0.977	0.988	0.067	0.665	0.6966	0.6989
■ F1-Score	0.991	0.991	0.99	0.9852	0.9816	0.9821	0.9755	0.979	0.126	0.154	0.4978	0.5007

Figure 10. Compare classification results

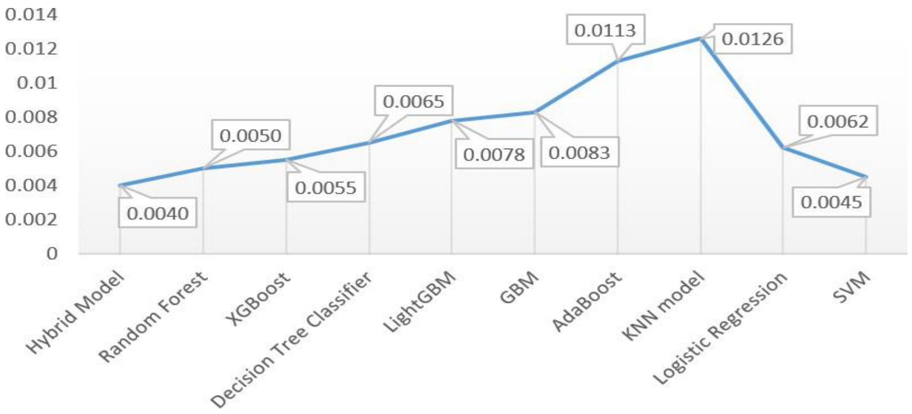


Figure 11. False Alarm Rate

The time it takes for a trained model to process a new sample and produce the prediction is called Prediction Time. In real-world scenario, especially in enterprise environments, detection system should process events in a real time manner to trigger the identification and mitigation of ransomware threats quickly. A shorter prediction time means the system can respond more quickly to the potential threats, reducing the window of opportunity for ransomware to do damage. Figure 12 demonstrates the prediction time for the ten algorithms used in this paper, highlighting the proficiency of the hybrid model in providing rapid predictions.

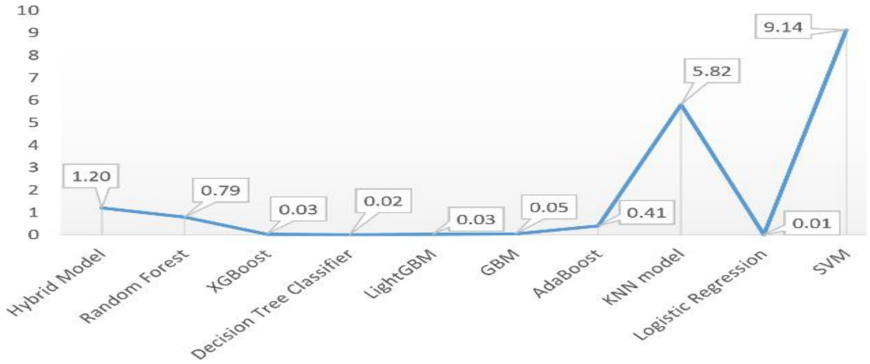


Figure 12. Prediction Time

The time it takes to train a machine learning model with given dataset is call Training Time. While training time is typically a one-time investment, it is important to consider when developing and updating detection models. A faster training time also allows the model to be updated more regularly so it can continue to detect new ransomware variants and attack strategies. Figure 13 illustrates the trade-offs in model complexity for the training times of all algorithms applied on the present dataset.

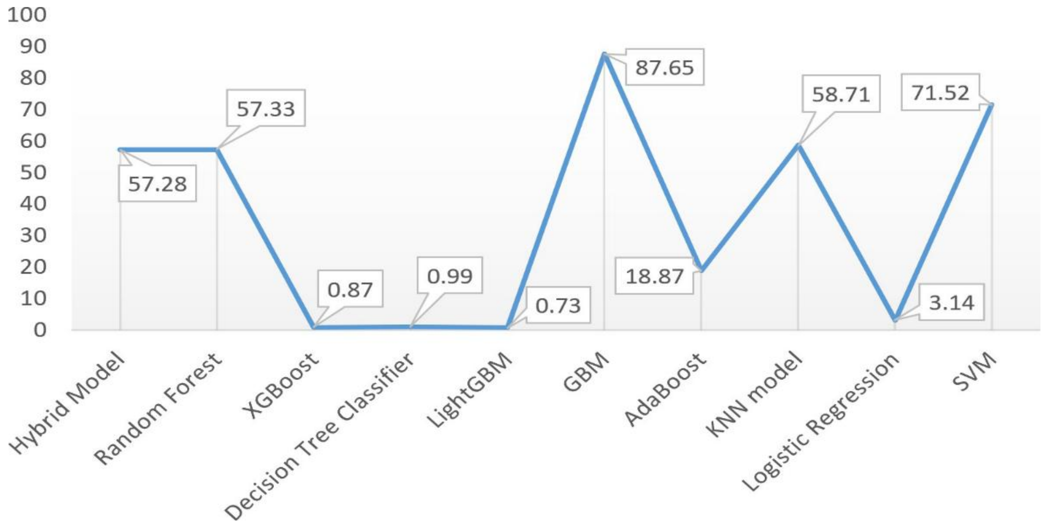


Figure 13. Training Time

The results of this paper provide valuable insights into how machine learning techniques can be used to detect internal ransomware threats. One of these insights is that the integration of ensemble methods through learning techniques has proven effective in identifying complex patterns and behaviors associated with ransomware attacks. The hybrid model, which combines the strengths of Random Forest and XGBoost algorithms, showed outstanding performance as it achieved 99.49% accuracy. The comprehensive feature engineering process, which included the extraction of features from PE heads, sections, imports and binary patterns, also demonstrated its significant role in improving the accuracy of detection of such threats.

The research methodology utilised in this paper includes the collection, preprocessing, analyzing, and evaluation of a variety of data, as well as the use of different machine learning algorithms, which provides a comprehensive framework for future research in this field. However, it is important to acknowledge the limitations of this paper, including the reliance on synthetic datasets and the need for further validation using real-world attack samples. Additionally, the dynamic nature of ransomware threats necessitates continuous updates to the detection models to ensure their effectiveness against evolving attack techniques.

This paper highlights the importance of adopting a hybrid model that combines supervised learning algorithms, providing a more effective and comprehensive ransomware detection, contributing to the system’s ability to detect threats more effectively.

8. Conclusion

This paper has demonstrated the effectiveness of machine learning on categorising insider-enabled ransomware threats using Windows Portable Executable (PE) files metadata. Using an extensive dataset of over 138K samples that include ransomware alongside benign samples, shows the ability to improve ransomware insider attack detection accuracy by combining powerful feature engineering and applying several machine learning algorithms. Compared to other existing solutions, the hybrid model, using supervised learning algorithms like Random Forest and XGBoost, presented a higher precision and recall rates. This paper conclusively demonstrates, the necessity of employing a combination of supervised techniques in mitigating ransomware threats both known and emerging ransomware threats initiated from the trusted networks within an organization.

In addition, the paper draws attention to the importance of feature selection and engineering in enhancing machine learning algorithms. Therefore, by curating and engineering features from the input data, which corresponds to behaviors of interest for ransomware detection, the models can thus be trained in a more effective manner resulting in an improved ac- curacy rate and better generalization into new unseen datasets. To better understand the overall process, the most informative features are identified from binary files metadata (PE Files) and used to train ransomware detection models to increase their efficiency for identifying ransomware threats.

Additionally, this signifies the necessity to combine supervised learning methods for threat detection which in turn results in an even more robust detection system. The findings also indicate that analysing encryption-based attributes in PE files can help to detect ransomware attacks in early stage and enhance its prevention, particularly when it comes to insider threat where individuals’ permitted access to systems may be able to execute them for malicious purposes. Furthermore, it underscores how supervised learning functionality can be offered to identify the threats that exist in documents that are already available for resilience of the detection system.

This paper serves as a preliminary exploration for future investigations due to its performance analysis of different machine learning algorithms and a thorough dataset, and methodical approach towards collecting and preprocessing data.

However, it should be noted that the limitations of this paper are based on synthetic datasets which necessitate the need for further verification using real-world attack samples. Further, the variability of ransomware threats requires that detection models be perpetually refreshed to prove their efficacy in combating newly-evolving attack methods.

Given that the malicious insiders possess greater access to sensitive data and resources for the organization, they are considered one of the most significant threats. This paper proposed machine learning methods for identifying and classifying a number of insider ransomware attacks using ten well known Machine Learning techniques, out of which nine are supervised machine learning classifiers and one is a hybrid model.

Among the proposed algorithms, the Hybrid Model provides the highest accuracy of 99.49%; the other accuracy values are RF with 99.46%, XGBoost with 99.42%, Decision Tree with 99.10%, LightGBM with 98.89%, GBM with 98.91%, Ad- aBoost with 98.51%, KNN with 98.77%, Logistic Regression with 71.67%, and SVM with 69.72%.

Therefore, the future presented models can come up as base- line solutions with respect to scaling datasets by considering new relevant features and ransomware trends on insider users. These models can further enhance their accuracy when the variety of features and trends in ransomware insider attacks are increased or updated after obtaining data with higher diversity. This could pave the way for new research trends that prescribe similar conditions to be imposed and elicited as detection mechanisms and methods in identifying insider attacks associated with almost any domain of an organization. This is due to most businesses that use machine learning models to make significant business rulings, and when a model performs well, these results lead to better decisions. Mistakes are costly (both financially and morally). However, by increasing model accuracy, this cost is decreased. ML-based research enables users to send huge amounts of data to computer algorithms, which subsequently assess, recommend, and make decisions depending on the information provided.

In this regard, this paper contributes to the literature on detecting ransomware using machine learning approaches, while focusing on addressing internal or insider-enabling attacks in an isolated environment. The results support that the hybrid modeling frameworks can increase detection accuracy and de- liver a resilient solution against ransomware attacks. Although this paper has made great progress in detecting insider-assisted ransomware attacks through machine learning methods on Windows PE files metadata, there are several further research paths to improve the effectiveness of the suggested solutions. Among these future research directions is that firstly, future research could extend the dataset with more ransomware samples and benign executables sourced from multiple lo- cations.

Secondly, some other machine learning techniques such as Convolutional Neural Networks and Recurrent Neural Networks (CNNs and RNNs) of deep learning models can help to further increase the performance overall detection. Thirdly, the method of static program analysis presented in this paper can be combined into a tool along with dynamic analysis techniques to form a complete detection framework. This hybrid model can help to detect some advanced ransomware utilizing evasion techniques which are effective against the static analysis. Lastly, the development of a monitoring and response system in real-time that can be implemented in enterprise environments is a relevant extension of this work. A major problem to be addressed was the issue of implementing this system in a scalable and efficient manner with minimum impact to the systems performance.

In conclusion, future work on this dataset may come with increasing the dataset size, using other machine learning techniques as well as applying an approach of dynamic analysis; in addition to building live detection systems and studying human factors with insider threats. This work will feed into the continuous creation of stronger defense mechanisms to recognize and respond to insider enabled ransomware threats.

References

[1] S. Routray, D. Prusti, and S. K. Rath, "Ransomware attack detection by applying machine learning techniques," in Machine Intelligence Techniques for Data Analysis and Signal Processing: Proceedings of the 4th International Conference MISP 2022, Volume 1. Springer, 2023, pp. 765–776.

[2] A. Al-Harrasi, A. K. Shaikh, and A. Al-Badi, "Towards protecting organisations' data by preventing data theft by malicious insiders," International Journal of Organizational Analysis, vol. 31, no. 3, pp. 875–888, 2023.

[3] K. Yamamoto, H. Tanaka, and T. Suzuki, "Transfer learning for adaptive ransomware detection in pe files," Journal of Network and Computer Applications, vol. 202, p. 103355, 2024.

[4] N. Patel and A. Gupta, "Explainable ai for ransomware detection: A shap-based approach on pe files," Expert Systems with Applications, vol. 167, p. 114122, 2021.

[5] L. Wang, Y. Zhang, X. Liu, and H. Chen, "Federated learning for privacy-preserving ransomware detection in pe files," IEEE Access, vol. 11, pp. 54 321–54 335, 2023.

[6] E. Johnson and S. Lee, "Graph neural networks for structural analysis of pe files in ransomware detection," IEEE Transactions on Information Forensics and Security, vol. 19, no. 1, pp. 143–156, 2024.

[7] M. Rodriguez-Garcia, M. Lupu, and B. Ghita, "Graph-based anomaly detection for insider ransomware threats: Modeling user-file-system interactions," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2345–2358, 2022.

[8] M. Gopinath and S. C. Sethuraman, "A comprehensive survey on deep learning based malware detection techniques," Computer Science Review, vol. 47, p. 100529, 2023.

[9] A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, and V. Martinez-Hernandez, "Ensemble learning for comprehensive insider ransomware threat detection," Expert Systems with Applications, vol. 215, p. 119225, 2024.

[10] R. Patel and A. Singh, "Insider-driven ransomware detection: A frame- work combining pe analysis and privilege escalation monitoring," in Proceedings of the International Symposium on Security and Privacy. IEEE, 2023, pp. 178–191.

[11] L. Zhang, X. Chen, R. Wang, and D. Liu, "Multi-modal detection of insider-initiated ransomware: Integrating pe file analysis and user behavior profiling," Computers & Security, vol. 115, p. 102638, 2022.

[12] H. H. Al-Khshali and M. Ilyas, "Impact of portable executable header features on malware detection accuracy." Computers, Materials & Continua, vol. 75, no. 1, 2023.

[13] S. Wiyono, D. S. Wibowo, M. F. Hidayatullah, and D. Dairoh, "Comparative study of knn, svm and decision tree algorithm for student's performance prediction," (IJCSAM) International Journal of Computing Science and Applied Mathematics, vol. 6, no. 2, pp. 50–53, 2020.

[14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.- Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," Advances in neural information processing systems, vol. 30, 2017.

[15] Y. Ding, H. Zhu, R. Chen, and R. Li, "An efficient adaboost algorithm with the multiple thresholds classification," Applied sciences, vol. 12, no. 12, p. 5872, 2022.

[16] I. Almomani, A. Alkhayer, and W. El-Shafai, "E2e-rds: Efficient end-to- end ransomware detection system based on static-based ml and vision- based dl approaches," Sensors, vol. 23, no. 9, p. 4467, 2023.

[17] A. A. Almazroi and N. Ayub, "Deep learning hybridization for improved malware detection in smart internet of things," Scientific Reports, vol. 14, no. 1, p. 7838, 2024.

[18] M. D'Onghia, M. Salvadore, B. M. Nespoli, M. Carminati, M. Polino, and S. Zanero, "Ap´ıcula: Static detection of api calls in generic streams of bytes," Computers & Security, vol. 119, p. 102775, 2022.

[19] M. H. L. Louk and B. A. Tama, “Tree-based classifier ensembles for pe malware analysis: a performance revisit,” Algorithms, vol. 15, no. 9, p. 332, 2022.

[20] R. Chaganti, V. Ravi, and T. D. Pham, “A multi-view feature fusion approach for effective malware classification using deep learning,” Journal of information security and applications, vol. 72, p. 103402, 2023.

[21] H.-m. Kim and K.-h. Lee, “Iiot malware detection using edge computing and deep learning for cybersecurity in smart factories,” Applied Sciences, vol. 12, no. 15, p. 7679, 2022.

[22] T. Baker and A. Short land, “Insurance and enterprise: cyber insurance for ransomware,” The Geneva Papers on Risk and Insurance-Issues and Practice, vol. 48, no. 2, pp. 275–299, 2023.

[23] A. Singh, Z. Mushtaq, H. A. Abosaq, S. N. F. Mursal, M. Irfan, and G. Nowakowski, “Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data,” Electronics, vol. 12, no. 18, p. 3899, 2023.

[24] Y. Yilmaz, O. Cetin, C. Grigore, B. Arief, and J. Hernandez-Castro, “Personality types and ransomware victimisation,” Digital Threats: Research and Practice, vol. 4, no. 4, pp. 1–25, 2023.

[25] H. Jo, Y. Lee, and S. Shin, “Vulcan: Automatic extraction and analysis of cyber threat intelligence from unstructured text,” Computers & Security, vol. 120, p. 102763, 2022.

[26] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, “{UNVEIL}: A {Large-Scale}, automated approach to detecting ran- somware,” in 25th USENIX security symposium (USENIX Security 16), 2016, pp. 757–772.

[27] S. Jacob, “The rapid increase of ransomware attacks over the 21st century and mitigation strategies to prevent them from arising,” 2023.

[28] P.-H. Chen, R. Bodak, and N. S. Gandhi, “Ransomware recovery and imaging operations: lessons learned and planning considerations,” Journal of Digital Imaging, vol. 34, no. 3, pp. 731–740, 2021.

[29] R. A. Alsowail and T. Al-Shehari, “Techniques and countermeasures for preventing insider threats,” PeerJ Computer Science, vol. 8, p. e938, 2022.

[30] M. D. Firoozjaei, N. Mahmoudyar, Y. Baseri, and A. A. Ghorbani, “An evaluation framework for industrial control system cyber incidents,” International Journal of Critical Infrastructure Protection, vol. 36, p. 100487, 2022.

[31] N. N. Neto, S. Madnick, A. M. G. D. Paula, and N. M. Borges, “Developing a global data breach database and the challenges encountered,” Journal of Data and Information Quality (JDIQ), vol. 13, no. 1, pp. 1–33, 2021.

[32] T. Al-Shehari, D. Rosaci, M. Al-Razgan, T. Alfakih, M. Kadrie, H. Afzal, and R. Nawaz, "Enhancing insider threat detection in imbalanced cybersecurity settings using the density-based local outlier factor algorithm," IEEE Access, 2024.

[33] T. Rains, Cybersecurity Threats, Malware Trends, and Strategies: Dis- cover risk mitigation strategies for modern threats to your organization. Packt Publishing Ltd, 2023.

[34] K. Albulayhi and Q. A. Al-Haija, "Early-stage malware and ransomware forecasting in the short-term future using regression-based neural net- work technique," in 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2022, pp. 735–742.

[35] O¨. Eminagˇaogˇlu, H. Akyıldırım Begˇen, and G. Aksu, "Flora of kılıc¸kaya village (yusuf eliartvin, turkey)," 2021.

[36] M. Thite and R. Iyer, "Addressing the gap in information security: an hrcentric and ai driven framework for mitigating insider threats," Personnel Review, no. ahead-of-print, 2024.

[37] R. Elakkiya, P. Vijayakumar, and N. Kumar, "An optimized generative adversarial network based continuous sign language classification," Expert Systems with Applications, vol. 182, p. 115276, 2021.

[38] T. McIntosh, T. Susnjak, T. Liu, D. Xu, P. Watters, D. Liu, Y. Hao, A. Ng, and M. Halgamuge, "Ransomware reloaded: Re-examining its trend, research and mitigation in the era of data exfiltration," ACM Computing Surveys, 2024.

[39] H. Zuhair and A. Selamat, "An empirical analysis of machine learn- ing efficacy in anti-ransomware tools," in American University in the Emirates International Research. Springer, 2020, pp. 41–49.

[40] D. Hitaj, G. Pagnotta, F. De Gaspari, S. Ruko, B. Hitaj, L. V. Mancini, and F. Perez-Cruz, "Do you trust your model? emerging malware threats in the deep learning ecosystem," arXiv preprint arXiv:2403.03593, 2024.

[41] N. M. Chayal, A. Saxena, and R. Khan, "A review on spreading and forensics analysis of windows-based ransomware," Annals of Data Science, vol. 11, no. 5, pp. 1503–1524, 2024.

[42] N. Zangrando, P. Fraternali, M. Petri, N. O. Pinciroli Vago, and S. L. Herrera Gonza´lez, "Anomaly detection in quasi-periodic energy con- sumption data series: a comparison of algorithms," Energy Informatics, vol. 5, no. Suppl 4, p. 62, 2022.

[43] A. El Hariri, M. Mouiti, and M. Lazaar, "Realtime ransomware process detection using an advanced hybrid approach with machine learning within iot ecosystems," Engineering Research Express, vol. 7, no. 1, p. 015211, 2025.

[44] G. Murray, M. Falkeling, and S. Gao, "Trends and challenges in research into the human aspects of ransomware: a systematic mapping study," Information & Computer Security, 2024.

[45] G. Li, H. Xiong, and Y. Zhang, "A review of insider threat detection: Classification, machine learning, and challenges," Applied Sciences, vol. 10, no. 15, p. 5208, 2020. [Online]. Available: https://doi.org/10.3390/app10155208

[46] S. I. Bae, G. B. Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," Concurrency and Computation: Practice and Experience, vol. 32, no. 18, p. e5422, 2020.

[47] B. M. Khammas, "Ransomware detection using random forest tech- nique," ICT Express, vol. 6, no. 4, pp. 325–331, 2020.

[48] P. W. Njoroge, "A crypto-ransomware detection model for the pre- encryption stage using random forest algorithm," Ph.D. dissertation, KCA University, 2022.

[49] Y. Pant, "Malware detection in executable files using xgboost algorithm," Ph.D. dissertation, Dublin, National College of Ireland, 2022.

[50] R. Kumar and S. Geetha, "Malware classification using xgboost-gradient boosted decision tree," Adv. Sci. Technol. Eng. Syst, vol. 5, no. 5, pp. 536–549, 2020.

[51] K. Lee, S.-Y. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," IEEE access, vol. 7, pp. 110 205–110 215, 2019.

[52] K. A. Shukla, G. Chettiar, A. Choudhary, A. Thakur, and S. Kumar, "Integrating comparison of malware detection classification using lgbm and xgb machine learning algorithms," in 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS). IEEE, 2022, pp. 1–7.

[53] M. Binsawad, "Enhancing pdf malware detection through logistic model trees." Computers, Materials & Continua, vol. 78, no. 3, 2024.

[54] H. Ismail, R. G. Utomo, and M. W. A. Bawono, "Comparison of support vector machine and random forest method on static analysis windows portable executable (pe) malware detection," JURNAL MEDIA INFORMATIKA BUDIDARMA, vol. 8, no. 1, pp. 154–162, 2024.

[55] P. Narayana, H. M. Al-Jawahry, A. Kumar, M. Sowmya, and A. Sud- hakar, "Effective machine leaning based malware detection and clas- sification using improved voting method based decision tree," in 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). IEEE, 2024, pp. 1–4.