Implement a System that can Detect Ransomware Attacks in Real-Time using Behaviour Analysis

MSc Research Project
MSc In Cyber Security

MD MASUDUR RAHMAN
Student ID: X23218291

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: MD MASUDUR RAHMAN …….……………………………………………………………………………………………

Student ID: X23218291 ……………………………………………………………………………………………………..……

Programme: MSc in Cybersecurity …………………………………………………………… **Year:** 2024 …………………………..

Module: MSc Research Project …………………………………………………………………………………………….………

Supervisor: Kamil Mahajan …………………………………………………………………………………………….………

Submission Due Date: 29-01-2025 …………………………………………………………………………………………….………

Project Title: Implement a System that can Detect Ransomware Attacks in Real-Time using Behaviour Analysis …………………………………………………………………………………………….………

Word Count: 6066 ……………………………………… **Page Count** 21 …………………………………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: MD MASUDUR RAHMAN ……………………………………………………………………………………………………

Date: 28-01-2025 ……………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Implement a System that can Detect Ransomware Attacks in Real-Time using Behaviour Analysis

Md Masudur Rahman
X23218291

Abstract

Ransomware remains a critical challenge in cybersecurity, requiring innovative methods for detection because of the sophistication of modern-day attack patterns. This paper deals with the development of a machine learning-based ransomware detection framework and investigates the efficacy of Logistic Regression, Random Forest, and Support Vector Machine. Featured are DebugSize and ExportSize, two of the most important features showing great dispersion across all ransomware files. In their performance, the Random Forest model performed better, realizing an accuracy of 99.67%, an AUC of 0.9994 close to perfect, and minimum false positives and negatives; it proves to be more reliable when put into practical use. The study has also found a manual prediction scenario for any instances in dynamic environments. Though this approach improves detection accuracy, challenges such as scalability and computation efficiency do prevail. Some of the future directions are lightweight models for IoT, privacy-preserving methods such as federated learning, and hybrid approaches incorporating behaviour-based systems for zero-day threats handling. This work lays a solid foundation for developing scalable and adaptive ransomware detection solutions.

Keywords: *Ransomware detection, machine learning, Random Forest, real-time detection, DebugSize, ExportSize, ensemble learning, cybersecurity, behavior-based systems, zero-day threats*

1 Introduction

1.1 Background

Ransomware attacks are among the fastest-growing cybersecurity threats, which have prospered in sophistication and scope to result in destructive financial and operational consequences across the globe. Also, ransomware attacks target people, organizations, and infrastructures that play a critical role and exploit vulnerabilities with the intent of encrypting data and asking for ransoms. In fact, various studies show that the traditional detection methodology that relies on static signatures has proven inadequate to deal with the rapid evolution and polymorphism of ransomware variants, making it necessary to have a more dynamic approach (Yu, et al., 2024). This research, therefore, focuses on the development of a system that will be able to detect ransomware attacks in real time, taking advantage of behavior analysis so as to bridge the gap left by the traditional methods of ensuring an effective and timely response.

1.2 Behaviour Analysis for Ransomware Detection

Behavior analysis offers the novelty in ransomware detection through observing system operation activities that are not within the set normal operation. Unlike signature-based methods, behavioral analysis will use machine learning techniques to comprehend typical system behavior and recognize abnormalities. This has been identified as promising for combating complex strains of malware that bypass signature-based detection methods (Limer, et al., 2024). This research will contribute to the emerging field of cybersecurity by demonstrating an effective model for ransomware detection through a combination machine learning, data analysis, and real-time data processing.

1.3 Research Question and Objectives

The primary research question addressed in this study is: How can behavior analysis automatically detect ransomware attacks in real time using machine learning?

1.4 Research Objectives

- To develop a system architecture that collects and analyzes endpoint data in real time.
- To train machine learning models to detect behavioral anomalies indicative of ransomware.

With such a focus on these objectives, this study contributes to establishing a framework that can be translated into an actionable ransomware detection system in near real-world environments. Besides this, the real-time nature of the system aims at reducing potential damages to a much more negligible rate, as it ensures quick isolation of threats, leading to minimal operational interruptions and loss of data.

1.5 Structure of the Report

The rest of the report is organized as follows:

- **Literature Review**: Provide a detailed view of the related work in literature to ransomware detection methods, in fact, studies the existing approaches and challenges and critical reviews of the existing solutions for the same.
- **Methodology**: Details the methodology of the research, including methodology of evaluation and testing, data collection, and statistical analysis techniques.
- **Design Specification**: Design Specification-describes the architecture of the proposed system; the design of key modules such as data collection and/or analysis will be outlined.
- **Implementation**: How the proposed solution was implemented, with all the data transformation, model development, and tools used, is presented.
- **Discussion**: Presents analysis of results and findings through experimental data and visual aids such as graphs and charts. Also, a critical discussion of the implications from an academic perspective and practical point of view is included.
- **Experiments/Case Studies**: Explains each of the experiments done during the evaluation; includes strengths, limitations, and suggestions on how to improve.

- **Conclusion and Future Work**: Summarizes the research question and objectives, key findings, and their implications. Also discusses possible future research directions and opportunities for commercialization.

2 Related Work

2.1 Introduction

For this reason, ransomware detection can be considered among the most important fields in cybersecurity research. In this paper, a review of existing detection methods will be discussed in detail, from signature-based methods to machine learning and behavior-based systems. This paper will critically assess the strengths and weaknesses of these approaches, pointing out challenges proposed by advanced ransomware variants. It also discusses some ethical and privacy concerns related to ransomware detection and specifies the lacuna in the existing solutions, thereby highlighting the need for real-time and dynamic Detection Systems. This review synthesizes the related work and points out the urgency for innovative approaches against the ransomware threat landscape that keeps on evolving.

2.2 Signature-Based Detection

Signature-based detection techniques, which are widely used in the detection of ransomware threats. These techniques mainly rely on predefined patterns and signatures derived from known malware samples.

Larocque et al. (2024) mentioned that APSE, using unsupervised learning algorithms, dynamically builds highly specific signatures in an autonomous way and improves conventional signature-based detection methods at multiple scales. Because of this, the new approach is able to answer the shortcomings of static signatures by making it possible to adaptively respond against new kinds of threats in real time. While the authors of this work consider that APSE has learned how to detect zero-day ransomware variants with high accuracy, APSE may face resource management in a computationally limited environment.

Anikolova et al. (2024) note that inherent rigidity in static signature-based detection methods is conducive to evasion techniques used by ransomware variants. They go on to discuss the BASE framework, emphasizing that the behavior-based approach is much more adaptable compared to the rigid static methods. Their paper, at the same time, underlines a historical role of signature-based detection in laying such a foundation for threat identification systems.

Aslan and Samet (2020) underlined that signature-based detection could detect quite fast, simple, and resource-effectively well-known threats but is criticized for detecting unknown malware since it relies on a static database. This makes it ineffective against adaptive ransomware. Their review aimed at underlining the necessity of hybrid methods to put a bridge between known and unknown threat detection.

Su et al. (2024) presents the DOSA framework, which couples signature mapping with dynamic analysis to counter the polymorphic behavior of ransomware. While DOSA might show the potential to evolve along with obfuscation patterns, Su et al. recognizes that static signature reliance remains a bottleneck in addressing highly novel ransomware strains.

Olabiyi (2024) alludes to the signature-based detection that has historical importance. While this has been effective in early stages of ransomware evolution, the paper identifies critical drawbacks that this technique faces, including high false-negative rates and vulnerability to obfuscation and encryption techniques.

While signature-based methods have been very instrumental in ransomware detection, their static nature limits their effectiveness against sophisticated ransomware variants. The latest frameworks, such as APSE and DOSA, which try to outstrip these challenges, still face difficulties in achieving comprehensive adaptability.

2.3 Machine Learning for Ransomware Detection

Machine learning has lately become a point of change in ransomware detection due to its analytic feature for patterns and the ability of adaptation with new threats. Both techniques are employed: supervised and unsupervised, which enable better detection and identification of new variants.

As per Ispahany et al. (2024), the supervised learning models do a pretty good job in classifying the known types of ransomware from labeled datasets. However, the authors raise concerns about dataset quality and limited feasibility in real-time scenarios. They emphasize the requirement for real-time detection models that integrate machine learning with adaptive frameworks to overcome these limitations.

Zahoora et al. (2022) propose an unsupervised deep learning model using a Contractive Auto Encoder (CAE) for feature extraction and a cost-sensitive Pareto Ensemble classifier (CSPE-R) for detection. This approach effectively mitigates false positives and negatives, making it particularly robust against zero-day ransomware. However, the authors note that the computational cost of such models can be a limitation for deployment in resource-constrained environments.

Islam 2024 overviews various machine learning models on the detection of ransomware attacks. Major gaps in scalability and real-world applicability are investigated. The paper insists that dynamic analysis should be included in the solution to increase the detection rate by tuning up the supervised models to handle the polymorphic variants of ransomware.

Urooj et al. (2021) emphasize dynamic analysis in machine learning-based frameworks. Their contribution overviews the synergistic use of deep learning with dynamic analysis techniques in the detection of ransomware on heterogeneous platforms, including IoTs and cloud environments. Though such techniques are highly effective, the authors show how it remains challenging to obtain high-quality datasets which are sufficient for such robust training.

Specifically, deep models and unsupervised techniques in machine learning, which offer significant benefits concerning the detection of advanced ransomware. However, there is still a need for further research directions on how best to reduce computational cost, dataset dependence, and scalability issues for optimal practical implementation.

2.4 Behaviour-Based Detection Systems

Behaviour-based detection systems provide runtime identification of ransomware by focusing on the extracted operational characteristics and inherent behavioural patterns of malware. These are very good at identifying new variants of ransomware, which cannot be done by traditional static signature-based models.

Loco et al., 2024; proposed the Adaptive Behaviour-Based Ransomware Detection system that relies on a dynamic flow signature to capture the runtime behavioural pattern of ransomware attacks. The model further relies on machine learning, which allows adaptation to zero-day threats and encrypted communications. ABRD exhibited high detection accuracy, scalability, and minimized false positives, hence becoming a robust tool in modern ransomware defence.

Garter et al., 2024; proposed the Adaptive Behaviour Profiling System, which couples dynamic behaviour modelling with network pattern profiling to find out ransomware activities. This ABPS had better accuracy and sensitivity by identifying clear behavioural and network indicators compared to conventional methods. The system also kept latency low while maintaining operational efficiency, practically proving its applicability in real-time applications in a number of sets of environments.

Cliford et al., 2024; introduced the BSAD framework for ransomware identification, which is dependent on dynamic pattern recognition and machine learning through behavioural anomalies. The BSAD method proved efficient in some of the most challenging kinds of ransomware detection, such as polymorphic ones with low latency and low resource utilization, thus making them suitable for both the cloud and on-premise methods of deployment.

Welderman et al. (2024) emphasized temporal behaviour modelling (TBM) as a foundational approach, focusing on time-based sequences of ransomware activity. TBM proved effective in distinguishing ransomware from benign software through sequential action analysis, achieving high accuracy and scalability despite obfuscation tactics.

These altogether point to the strengths of behaviour-based systems in adaptive ransomware detection to provide robust, scalable solutions to cope with the evolving ransomware threat landscape.

2.5 Comparative Analysis

Comparing the different ransomware detection methods, their performances differ so much in such performance metrics as accuracy, recall, and latency that it gives a view into the effectiveness of the approaches.

Azeem et al. (2024) assessed machine learning models for malware detection, finding that RF had the best performance with 97.68% accuracy. In their study, feature selection and dataset balancing were the prime factors for improving model performance. However, latency was one point of weakness, as RF-based systems face difficulties in real-time applications due to the presence of computationally intensive tasks.

Balantrapu 2021 performed a systematic review of machine learning algorithms for malware classification. Algorithms that fell under ensemble methods and neural networks showed high precision and recall. However, their computational overheads remained very inefficient. Simpler models, like Decision Trees, showed faster processing times but lost a little on accuracy and were thus suited to latency-critical environments.

Koyirar et al. (2024) proposed the framework for process memory analysis, which achieved high accuracy with low false positives by monitoring memory access patterns. This technique outperformed others in terms of real-time detection with minimum latency but had some scalability issues in a large-scale environment.

Ganfure et al., 2022; proposed the DeepWare model, which used deep learning and hardware performance counters to identify ransomware. DeepWare reached a recall of 98.6% and almost zero false-positive rate within a 100-ms detection window, outperforming traditional models such as OC-SVM and EGB. Its superior timeliness and accuracy make it particularly useful for zero-day ransomware detection, although its dependency on specialized hardware curbs widespread adoption.

In a nutshell, while the complex models, such as DeepWare, can achieve accuracy, even recall, simpler frameworks can provide a good balance of latency with respect to computational efficiency. This again creates the possibility of having context-specific ransomware detection strategies.

2.6 Limitations of Current Solutions

While many new methods for ransomware detection are being developed in a positive vein, significant gaps and challenges in the current deployments of real-time solutions mark complex and ever-changing cyber environments.

Malik et al. (2023) discussed some of the following limitations in malware detection in CPS: Most of the current techniques generate a high number of false alarms frustrating the critical infrastructures. They pointed out the gap between utilizing underused metaheuristic algorithms to improve the detection accuracy and reduce false alarms; the authors said that current solutions do not adapt to dynamic CPS configurations, an important issue.

Qureshi et al. (2024) have identified the gaps in IoT malware detection, including but not limited to an utter lack of comprehensive IoT-specific datasets and incorporation of an interdisciplinary approach toward a scalable and real-time detection system. Further, forensic techniques presently have a lacuna in handling anti-forensic tactics, which is turning them into poor analyzers and mitigators of ransomware threats effectively in IoT environments.

The weakness in the defence against ransomware on CPS has been indicated by Benmalek 2024, due to unique attack surfaces and safety-critical enforcements. The operational and technical particularities brought about by ransomware within industrial control systems and healthcare networks remain unaddressed by existing solutions. This paper pinpoints strong resilience strategies, especially with regard to CPS environments.

In this sense, Botacin et al. (2021) discuss greater pitfalls: malware research itself, relying heavily on closed-source solutions and private datasets, is very negative regarding the reproducibility of the results and limits their experimental validation. He thus proposes creating standards (programming methodologies and datasets), used in research when it comes to open datasets, able to enhance scientific robustness and practical utility.

The current ransomware detection systems have scalability, adaptability, and reproducibility issues. These reasons therefore hint at the need to visit novel approaches which may integrate together the interdisciplinary techniques with open-source methodologies and system-specific solutions for effectively addressing the emerging ransomware threats.

2.7 Ethical and Privacy Concerns

Ethical and privacy concerns in ransomware detection have become of prime importance, regarding data gathering and analysis for the protection of users' rights. Yaacoub et al. (2023) have discussed some challenges surrounding IoT systems in regard to ethical hacking. This is especially so because most penetration testing methods can lead to sensitive information leakage when insecure methods of penetration testing are performed. Thus, they call for the need for strong frameworks that must provide a balance between systems security and reduced danger of data exploitation.

Wafula (2022) explores the data privacy risks of Kenyan SMEs and reveals that insufficient frameworks of privacy risk assessment increase the vulnerabilities. In this respect, his study calls for complete implementation of standard frameworks such as OCTAVE-Small that balance the protective measures against compliance parameters. Wafula also identifies that in analyzing a ransomware incident, SMEs have legal requirements to protect customer data against unauthorized breach disclosure.

In this regard, Aidonojie et al., 2024; explore, within that line, the legal intricacies facing data privacy in a world of automated systems but with specific focus on Nigeria. They note that incomplete legislation in data protection raises vulnerability to unauthorized access and breach of data. Thus, the authors raise a call for enhanced legislation that would mitigate the inadequate regulatory provisions given for privacy compliance vis-à-vis data collection and processing.

Butt (2023) provides general awareness about user-centric awareness of the risks of privacy. He confirms that the main educational tool, such as the "RansomAware" game, lowers human error, hence making them vulnerable to data breach. According to him, aware users and customers can avoid compromising sensitive data on ethical grounds pertaining to cybersecurity.

2.8 Summary of Related Work

The literature review underlines significant progress in ransomware detection, starting from the signature-based approach to machine learning and behaviour-based systems. While meaningful, these signature-based approaches have limited adaptability for dealing with sophisticated variants of ransomware. Deep unsupervised machine learning models show promises in offering solutions against zero-day threats but at the same time bring scalability and computational efficiency challenges. The behaviour-based systems show dynamic capabilities for finding new ransomware variants, while critical optimization demands lay at the basis of their resource-wise wide applicability. Moreover, ethical and other privacy challenges complicate the collection and analysis of data. These findings justify the need for robust, real-time, and behaviour-based systems because the evolution of ransomware has changed.

3 Research Methodology

The research methodology has been designed to give a systematic approach in the development and evaluation of a real-time ransomware detection system using behaviour analysis. It has several important steps for rigorous data preparation, model development, and its evaluation. The detailed procedure is outlined below, organized under dataset handling, preprocessing, exploratory data analysis, machine learning model development, and its evaluation and implementation tools.

3.1 Dataset Selection and Description

The dataset used was obtained from Kaggle, a repository for machine learning. This dataset consists of extracted features from executable files, which included both benign and ransomware samples. Some key features of the dataset were:

- **Size**: The dataset is approximately 62,485 rows and 18 columns, thus providing a good pool of data for training and testing.
- **Features**: Included attributes such as *DebugSize*, *ExportSize*, *DllCharacteristics*, and *BitcoinAddresses*.
- **Target Variable**: The column 'Benign' is a binary target variable, where 1 is benign and 0 is ransomware.

The dataset was selected due to its diversity and relevance for behaviour-based detection, providing a rich set of features for modelling ransomware patterns.

3.2 Data Preprocessing

Data preprocessing was an important step in ensuring the quality and consistency of data for machine learning models. This consisted of several sub steps:

Data Cleaning

- Verified the dataset for missing values using df.isnull().sum() and found no null entries.
- Dropped superfluous columns, which include ID and Timestamp; this was irrelevant, as those attributes did not contribute toward a behavior-based analysis.

Categorical Encoding

- Coded categorical variables with low cardinality using Label Encoding.
- Dropped high-cardinality categorical variables to avoid noise in the model.

Outlier Detection and Cleaning

- Applied the Z-score analysis to remove the far-out outliers. It has been important for noise reduction and hence improved performance of the model.

Feature Engineering

Added new features that improve model explainability:

- Debug_Export_Ratio: Ratio of DebugSize to ExportSize to detect the files with special characteristics.
- Stack_Resource_Ratio: Quotient of SizeOfStackReserve to ResourceSize for anomaly detection.

Feature Scaling

- Standardized the numerical features by using StandardScaler, ensuring that all features were normally distributed with improved convergence at model training.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) provided insights into the dataset's structure, patterns, and relationships. Key findings included:

Data Distribution

- Distributions for certain features such as DebugSize and ExportSize were visualized using histograms that showed the differences between benign and ransomware samples.
- Box plots outlined some important features that were presenting outliers and were then treated in the pre-processing.

Correlation Analysis

The heatmap of the correlation among numeric features was plotted, helping identify which variables to consider in training a model.

Behavioral Insights

Attributes like BitcoinAddresses showed clear patterns in ransomware files, which further validated their inclusion in the feature set.

3.4 Machine Learning Model Development

In this connection, machine learning models have been presented to classify files by their behavioural attributes. Training and evaluation of these three models have been done:

Logistic Regression

- Chosen as a baseline model because of simplicity and interpretability.
- Set to 1000 for maximum number of iterations to ensure convergence.

Random Forest

- Selected for its strong ensemble learning and resistance to overfitting.
- Set to 100 estimators to provide a reasonable balance between accuracy and computational efficiency.

Support Vector Machine

Used for its capability in handling non-linear decision boundaries. RBF kernel applied for better performance on high-dimensional data.

3.5 Evaluation Metrics and Analysis

These performance evaluation metrics were used to comprehensively judge the performance of machine learning models, which gives a comprehensive understanding of their capability and limitations. Accuracy is used as a primary measure representing the proportion of correctly classified samples and providing a general indication of model performance. In complementarity with accuracy, precision was used to assess the model's capability in identifying ransomware instances while reducing false positives. Recall, or sensitivity as it is sometimes referred to, is the measure for detecting all ransomware files-it has a sense of associated risk attached through false negatives. F1 score: the harmonic mean of precision and recall-is always more useful in situations with higher added value because of unbalancing in one or either set; thus, providing consistently a better estimate indicator about model performance. More AUC (Area Under the Curve) has allowed for its various class-change thresholds and essentially aggregating the capability referring correctly to the class by it.

The evaluation also included two critical visual aids for better interpretability: the confusion matrix and the ROC-AUC curve. The confusion matrix gave a detailed breakdown of model predictions into true positives, true negatives, false positives, and false negatives. This showcased some model-specific weaknesses, including tendencies to generate false alarms or miss the detection of ransomware. The ROC-AUC curve complemented the confusion matrix by considering the trade-offs between true positive rates and false positive rates. In contrast, the ROC-AUC curve showed the robustness of the model performance for all possible thresholds and helped to find optimal thresholds for classification. All these metrics and analyses put together provided a holistic evaluation framework that allowed ensuring the models were thoroughly assessed in terms of both accuracy and practical applicability regarding ransomware detection in real-time scenarios.

3.6 Implementation Tools

The research relied on a combination of programming languages, libraries, and hardware configurations to implement the methodology:

Programming Language

Python served as the primary language for its extensive library support in data analysis and machine learning.

Libraries

- Data Handling: pandas and numpy for data manipulation.
- Visualization: matplotlib and seaborn for generating plots.
- Model Development: scikit-learn for training and evaluating machine learning models.

3.7 Experiments and Case Studies

Case Study 1: Feature Analysis

- Analyzed the distribution across both DebugSize and ExportSize for benign and ransomware labels.
- This demonstrated large variance within these features for ransomware files, underpinning that their importance in detection is of great significance.

Case Study 2: Model Comparison

- The performance of three chosen models compared for selection of the best model.

Case Study 3: Manual Prediction

- Tested models using manually input feature values to simulate real-world scenarios.

3.8 Limitations and Challenges

The proposed methodology achieved success, despite a few identified limitations. Some of these are the features in the dataset limiting the exploration of novel ransomware behaviours and imbalanced classes, which are required to be handled with much care in order to prevent model bias. Scalability remains a challenge for resource-intensive models such as Random Forest in real-time deployment on resource-constrained systems. Moreover, while the models performed well on the dataset, their generalization to a completely new environment needed further testing to be robust. These limitations outline the future enhancement that has to be done, which includes more diverse datasets, lightweight models, and further testing across different real-world scenarios.

3.9 Summary

The approach of this research methodology has been holistic towards the construction of a ransomware detection system by means of behaviour analysis. In systematic fashion, it does data preprocessing, feature engineering, trains robust models, and applies in-depth evaluation metrics to create an effective framework. Challenges pertaining to scalability and constraints regarding the dataset were noticed, yet findings underline the great potential that can be offered by the behaviour-based system for real-time ransomware detection. The next step in such research work shall involve enhancing model adaptability and exploring lightweight solutions for more general application scenarios.

4 Design Specification

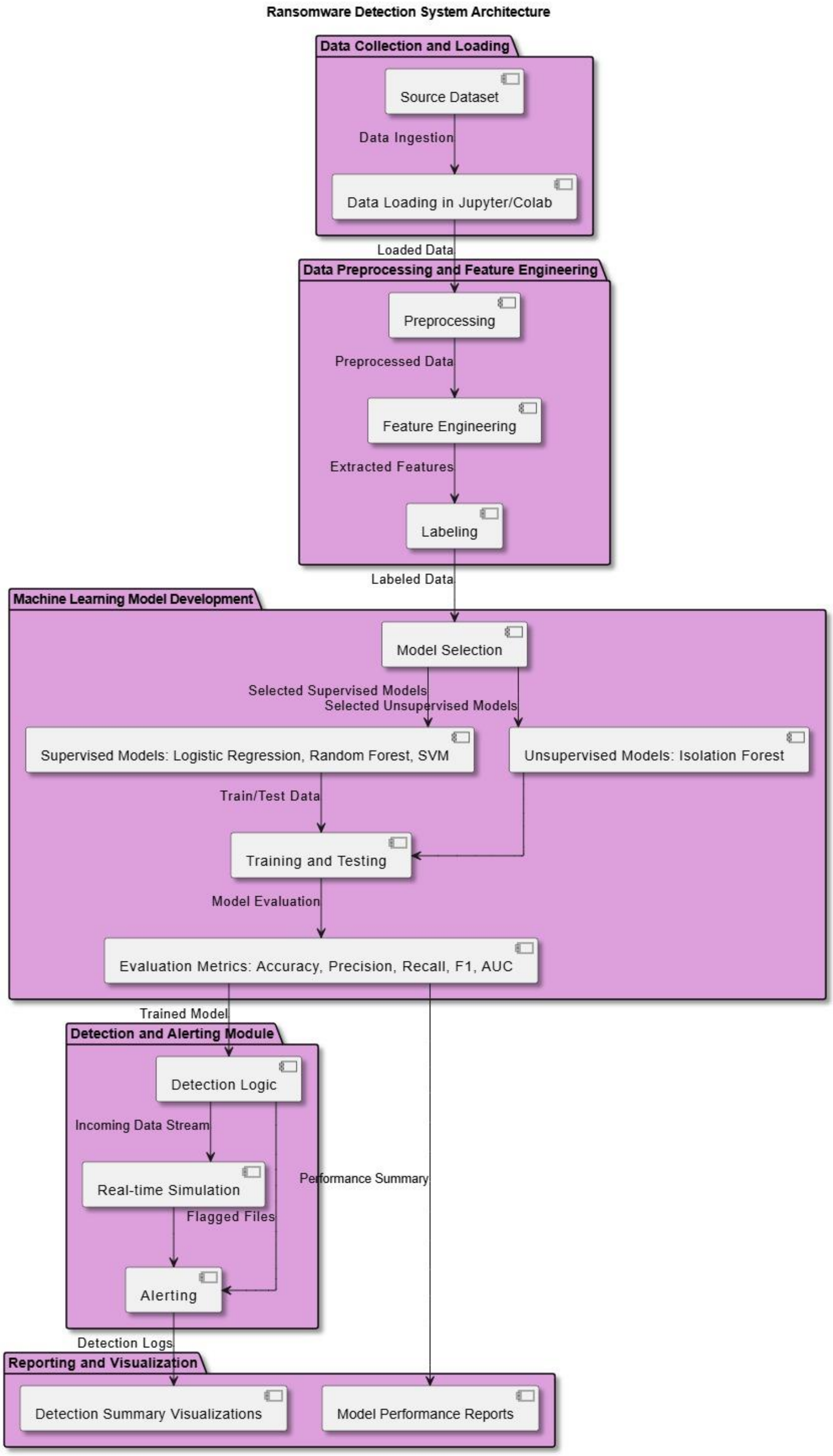

Ransomware Detection System Architecture
Data Collection and Loading
Source Dataset
Data Ingestion
Data Loading in Jupyter/Colab
Loaded Data
Data Preprocessing and Feature Engineering
Preprocessing
Preprocessed Data
Feature Engineering
Extracted Features
Labeling
Labeled Data
Machine Learning Model Development
Model Selection
Selected Supervised Models
Selected Unsupervised Models
Supervised Models: Logistic Regression, Random Forest, SVM
Unsupervised Models: Isolation Forest
Train/Test Data
Training and Testing
Model Evaluation
Evaluation Metrics: Accuracy, Precision, Recall, F1, AUC
Trained Model
Detection and Alerting Module
Detection Logic
Incoming Data Stream
Real-time Simulation
Performance Summary
Flagged Files
Alerting
Detection Logs
Reporting and Visualization
Detection Summary Visualizations
Model Performance Reports

Figure 1: Architectural Diagram

The proposed ransomware detection system is designed in a modular architecture to facilitate efficient real-time detection using behavior-based analysis. This architecture consists of four major components: Data Collection and Loading, Data Preprocessing and Feature

Engineering, Machine Learning Model Development, and Detection and Alerting Module, with an additional focus on Reporting and Visualization for result interpretation.

It begins with the process of Data Collection and Loading, which includes the ingestion of datasets with behavioral attributes of benign and ransomware samples from sources such as Kaggle. The data gets loaded into computational environments like Jupyter Notebook or Google Colab, preparing it for preprocessing. The Data Preprocessing and Feature Engineering stage cleans and optimizes the data by handling missing values, removing outliers, and engineering new features such as Debug_Export_Ratio and Stack_Resource_Ratio for capturing anomalous behaviors effectively. Labeling ensures the target variable is correctly encoded for machine learning models.

The supervised models to be used during the Machine Learning Model Development phase are Logistic Regression, Random Forest, and Support Vector Machine (SVM), which will classify the files as either benign or ransomware. An unsupervised Isolation Forest model is included for identifying zero-day ransomware anomalies. Their robustness is ensured by checking them with metrics such as accuracy, precision, recall, F1 score, and AUC.

The trained models are then deployed to the Detection and Alerting Module, where the incoming data streams are analysed for ransomware patterns. Flagged files trigger alerts and generate detailed logs for further action. The Reporting and Visualization component provides summaries of detections through visual aids like confusion matrices and ROC-AUC curves, along with detailed performance reports.

It gives scalability, adaptability, and thus practical application in dynamic cybersecurity environments through the modular architecture-what is quite critical for the effective countering of known and emerging ransomware.

5 Implementation

The considered implementation of the ransomware detection system represents the translation of the designed architecture into a functional and effective solution. First of all, the preparation of data was performed, where raw data was pre-processed by cleaning out outliers, handling missing values, and performing feature engineering, such as Debug_Export_Ratio and Stack_Resource_Ratio. Further, the labelled dataset was divided into training and testing subsets to proceed with model training and testing.

The developed and trained machine learning models are supervised ones, including Logistic Regression, Random Forest, and Support Vector Machine-SVM, using pre-processed data. Moreover, an unsupervised Isolation Forest model was developed for anomaly detection that represents potential zero-day ransomware. All the models were evaluated for their performance using accuracy, precision, recall, F1 score, and AUC for ensuring robustness. The Random Forest model was the most efficient in yielding the highest accuracy and reliability in real-time detection scenarios.

These trained models were then embedded into a real-time detection framework, enabling the system to scan incoming data streams for possible ransomware behaviour. The flagged files generated alerts supported by detection logs that can be analysed further. Implementation was done using Python programming, taking advantage of the various libraries such as pandas for data manipulation, numpy, scikit-learn, and matplotlib for modelling and visualization. Outputs of the system are performance reports and visualizations of detection for interpretation by stakeholders.

6 Evaluation

Various experiments have been conducted to validate the performance, efficiency, and robustness of machine learning models deployed in the ransomware detection system. The different key performance metrics considered for this study are as follows: accuracy, precision, recall, F1-score, and AUC. Graphical illustrations of the results included the confusion matrices, ROC curve, and distribution plots of these models to have an in-depth study of each one of them.

6.1 Experiment / Case Study 1: Feature Analysis

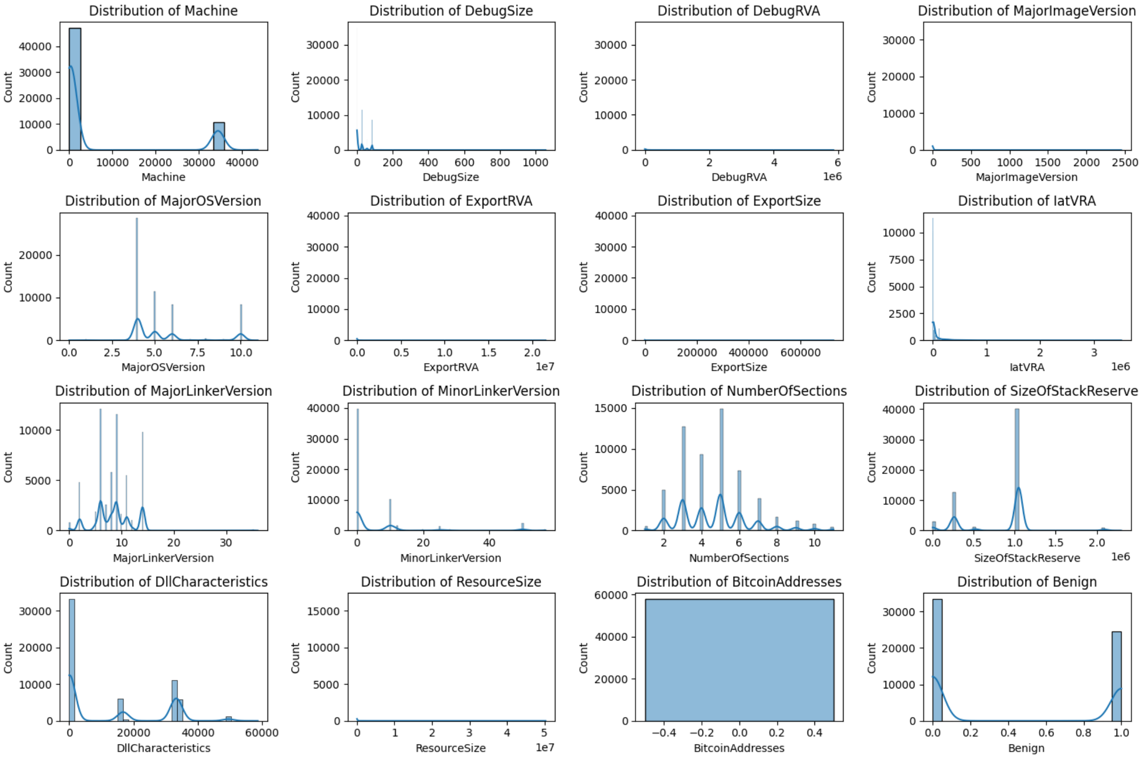


Figure 2: Distribution of Features

Feature analysis has been done to understand the distribution of the features DebugSize and ExportSize across the benign and ransomware labels. Using distribution plots, it could be observed that these variables are highly variant for ransomware files compared to their benign versions. In the case of DebugSize, benign files had a focused range of values, while ransomware files showed higher and wider peaks, which was indicative of anomalous behavior. Similarly, in the case of ExportSize, the values to be higher and more variant for ransomware thus reinforcing its usefulness in distinguishing malicious files.

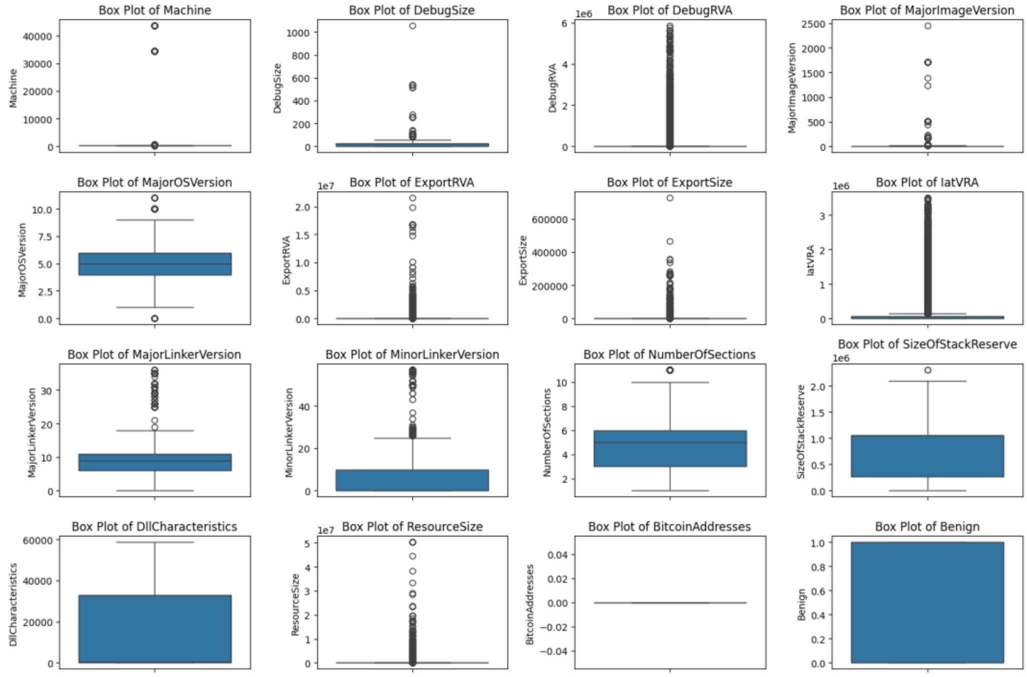
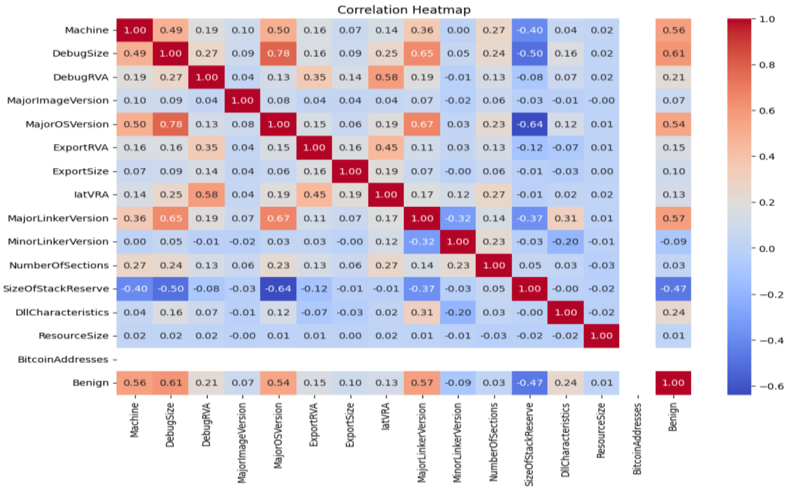


Figure 3: Box-Plot Distribution

Boxplots highlighted significant outliers within the ransomware group for features such as ExportSize and DebugSize. These are very important contributors to our models in finding ransomware. Finally, a correlation matrix showing strong relations among the features of DebugSize and DllCharacteristics with regards to ransomware behavior. It was really helpful in deciding on those given sets of features to include in model training for detection and further improving the performance.



	Machine	DebugSize	DebugRVA	MajorImageVersion	MajorOSVersion	ExportRVA	ExportSize	latVRA	MajorLinkerVersion	MinorLinkerVersion	NumberOfSections	SizeOfStackReserve	DllCharacteristics	ResourceSize	BitcoinAddresses	Benign
Machine	1.00	0.49	0.19	0.10	0.50	0.16	0.07	0.14	0.36	0.00	0.27	-0.40	0.04	0.02		0.56
DebugSize	0.49	1.00	0.27	0.09	0.78	0.16	0.09	0.25	0.65	0.05	0.24	-0.50	0.16	0.02		0.61
DebugRVA	0.19	0.27	1.00	0.04	0.13	0.35	0.14	0.58	0.19	-0.01	0.13	-0.08	0.07	0.02		0.21
MajorImageVersion	0.10	0.09	0.04	1.00	0.08	0.04	0.04	0.04	0.07	-0.02	0.06	-0.03	-0.01	-0.00		0.07
MajorOSVersion	0.50	0.78	0.13	0.08	1.00	0.15	0.06	0.19	0.67	0.03	0.23	-0.64	0.12	0.01		0.54
ExportRVA	0.16	0.16	0.35	0.04	0.15	1.00	0.16	0.45	0.11	0.03	0.13	-0.12	-0.07	0.01		0.15
ExportSize	0.07	0.09	0.14	0.04	0.06	0.16	1.00	0.19	0.07	-0.00	0.06	-0.01	-0.03	0.00		0.10
latVRA	0.14	0.25	0.58	0.04	0.19	0.45	0.19	1.00	0.17	0.12	0.27	-0.01	0.02	0.02		0.13
MajorLinkerVersion	0.36	0.65	0.19	0.07	0.67	0.11	0.07	0.17	1.00	-0.32	0.14	-0.37	0.31	0.01		0.57
MinorLinkerVersion	0.00	0.05	-0.01	-0.02	0.03	0.03	-0.00	0.12	-0.32	1.00	0.23	-0.03	-0.20	-0.01		-0.09
NumberOfSections	0.27	0.24	0.13	0.06	0.23	0.13	0.06	0.27	0.14	0.23	1.00	0.05	0.03	-0.03		0.03
SizeOfStackReserve	-0.40	-0.50	-0.08	-0.03	-0.64	-0.12	-0.01	-0.01	-0.37	-0.03	0.05	1.00	-0.00	-0.02		-0.47
DllCharacteristics	0.04	0.16	0.07	-0.01	0.12	-0.07	-0.03	0.02	0.31	-0.20	0.03	-0.00	1.00	-0.02		0.24
ResourceSize	0.02	0.02	0.02	-0.00	0.01	0.01	0.00	0.02	0.01	-0.01	-0.03	-0.02	-0.02	1.00		0.01
BitcoinAddresses																
Benign	0.56	0.61	0.21	0.07	0.54	0.15	0.10	0.13	0.57	-0.09	0.03	-0.47	0.24	0.01		1.00

Figure 4: Correlation Heatmap

The feature importance assessment for ransomware detection brought forth important facts, which find further support in the correlation heat map and distribution analyses of DebugSize and ExportSize. Figure 4 represents the strong correlations of the features DebugSize and ExportSize with the "Benign" label, hence their predictive relevance in the ransomware versus benign file classification task. DebugSize was correlated at 0.61 with the benign status, showing the effect on classification accuracy.

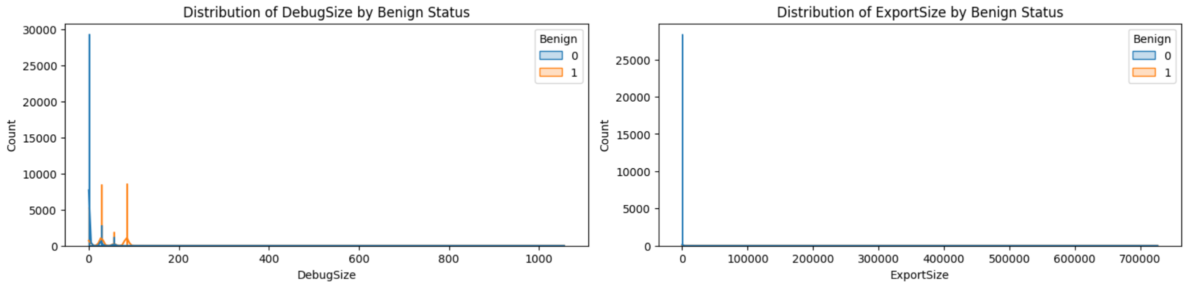


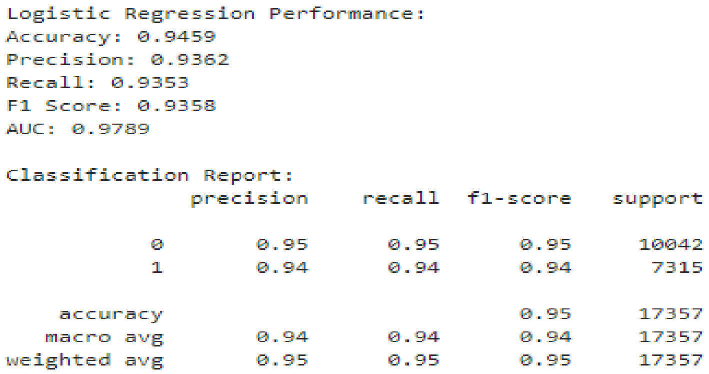
Figure 5: Distribution of DebugSize and ExportSize by Benign Status

Figure 5 depicts the distribution of features with respect to the benign and ransomware statuses for the features DebugSize and ExportSize. From the above plots, it is very evident that these features are highly variable in the case of ransomware files as compared to their benign versions. In the case of DebugSize, while it is concentrated in benign files, the range is higher in ransomware, which underlined the variability of this feature and its potential for detection. Similarly, in the case of ExportSize, the spread of values is greater in ransomware, and that also shows its relevance to the identification of malicious activities.

6.2 Experiment / Case Study 2: Model Evaluation and Comparison

The evaluation of machine learning models for ransomware detection provided critical insights into the performance and suitability of Logistic Regression, Random Forest, and Support Vector Machine (SVM).

Logistic Regression Classifier



```
Logistic Regression Performance:
Accuracy: 0.9459
Precision: 0.9362
Recall: 0.9353
F1 Score: 0.9358
AUC: 0.9789

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95     10042
           1       0.94      0.94      0.94      7315

    accuracy                           0.95     17357
   macro avg       0.94      0.94      0.94     17357
weighted avg       0.95      0.95      0.95     17357
```

Figure 6: Classification Report

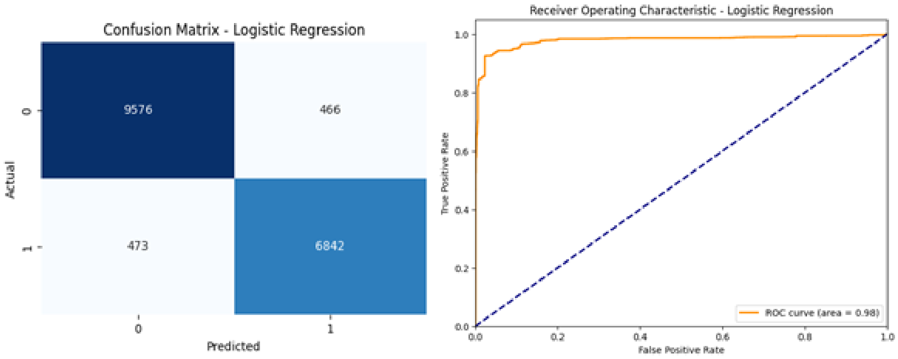
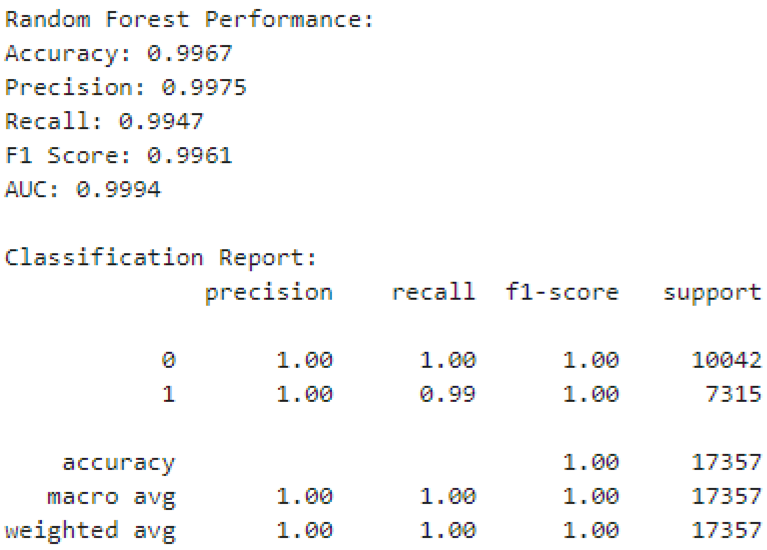


Figure 7: Confusion Matrix and ROC Curve

The Logistic Regression model achieved an accuracy of 94.59%, along with an AUC of 0.9789 that made it reliable for the detection of a basic level. However, performance showed weaknesses, especially in managing false negatives, according to the confusion matrix. So, this weakness may mean that such a model, while performing rather consistently, is not performing so well with more complex patterns of ransomware and perhaps fails to detect critical threats. The ROC curve for Logistic Regression showed its strong performance across different thresholds, but the sensitivity was not as robust as with the rest of the models.

Random Forest Classifier



```
Random Forest Performance:
Accuracy: 0.9967
Precision: 0.9975
Recall: 0.9947
F1 Score: 0.9961
AUC: 0.9994

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     10042
           1       1.00      0.99      1.00      7315

    accuracy                           1.00     17357
   macro avg       1.00      1.00      1.00     17357
weighted avg       1.00      1.00      1.00     17357
```

Figure 8: Performance Matrix of RF Model

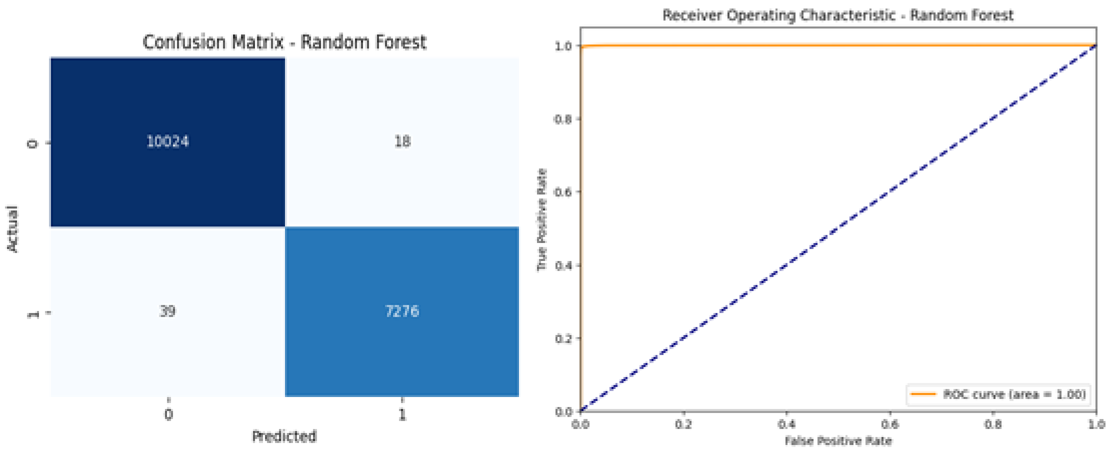
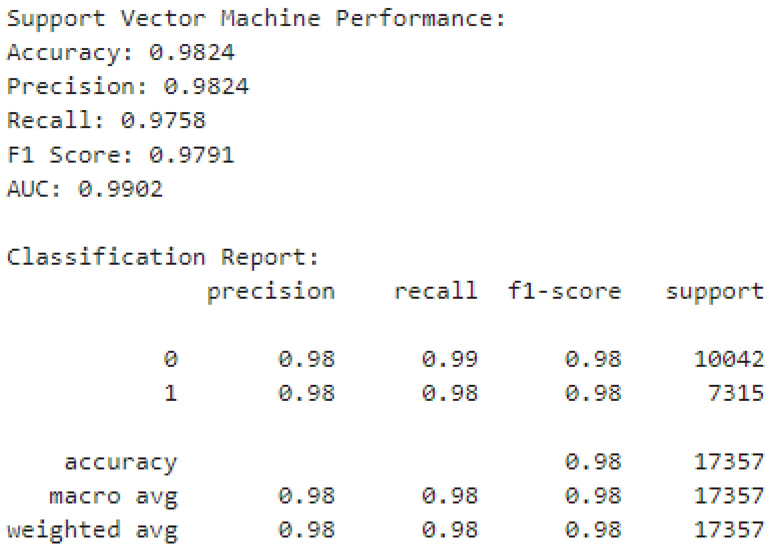


Figure 9: Confusion Matrix and ROC Curve

Random Forest came out to be the best, which gave an accuracy of 99.67%, precision of 99.75%, recall of 99.47%, and AUC of 0.9994. The confusion matrix of Random Forest showed very few false positives and false negatives; hence, it is highly suitable for real-time ransomware detection scenarios. Also, the ROC curve showed almost full distinguishability between ransomware and benign files, hence the most reliable model to deploy.



```
Support Vector Machine Performance:
Accuracy: 0.9824
Precision: 0.9824
Recall: 0.9758
F1 Score: 0.9791
AUC: 0.9902

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98     10042
           1       0.98      0.98      0.98      7315

    accuracy                           0.98     17357
   macro avg       0.98      0.98      0.98     17357
weighted avg       0.98      0.98      0.98     17357
```

Figure 10: Performance of SVM Model

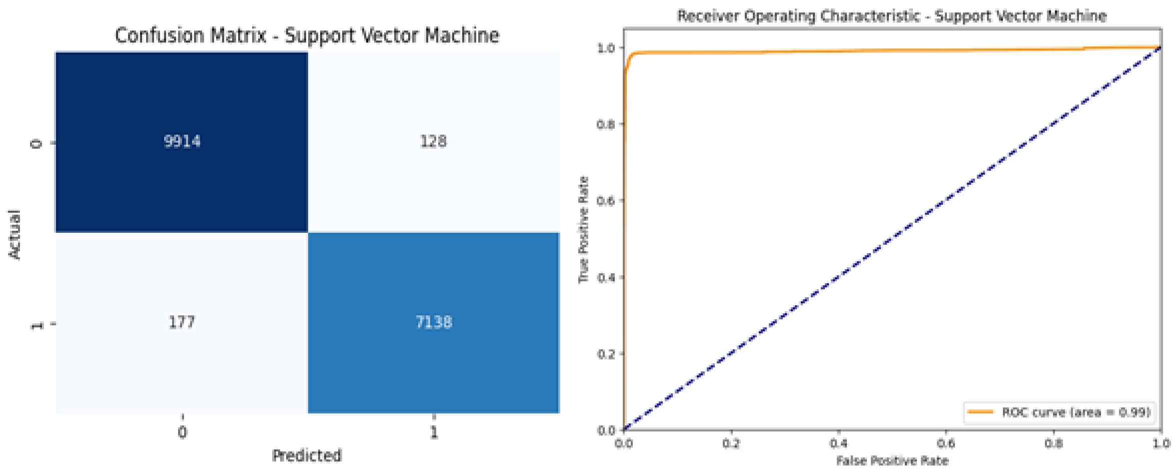
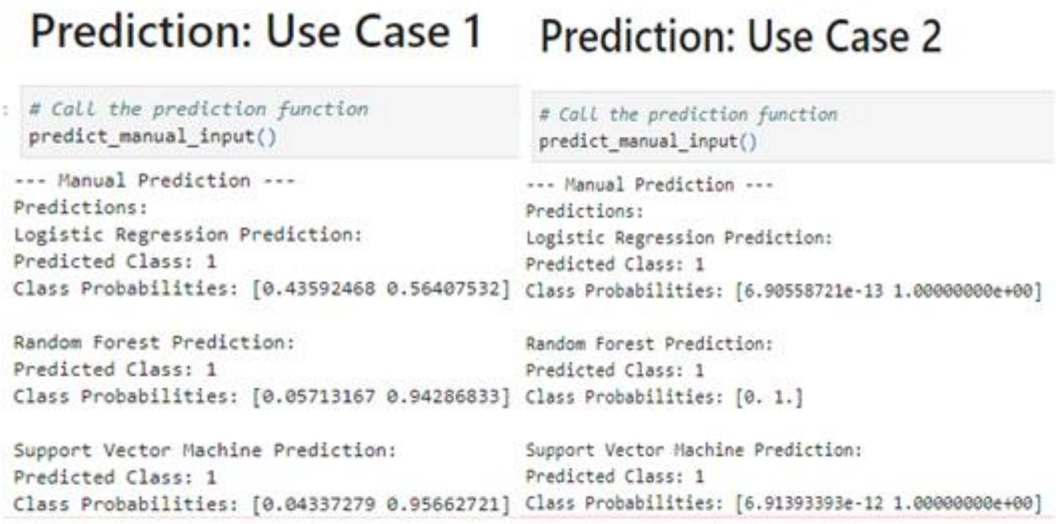


Figure 11: Confusion Matrix and ROC Curve

SVM yielded the following performances: an accuracy of 98.24%, a F1-score of 97.91%, AUC of 0.9902. Despite giving very good results, with a much higher false-positive rate compared to Random Forest, as revealed by its confusion matrix, this would seriously limit its applicability in sensitive environments where possible false alarms could lead to operational disruptions.

The comparative analysis underlined the Random Forest model for ransomware detection as superior, with an optimal balance between accuracy, precision, and recall, and minimum false positives and negatives. This therefore justifies its adoption for real-world implementation in robust cybersecurity frameworks.

6.3 Experiment / Case Study 3: Manual Prediction



Prediction: Use Case 1

```
# Call the prediction function
predict_manual_input()
```

```
--- Manual Prediction ---
Predictions:
Logistic Regression Prediction:
Predicted Class: 1
Class Probabilities: [0.43592468 0.56407532]

Random Forest Prediction:
Predicted Class: 1
Class Probabilities: [0.05713167 0.94286833]

Support Vector Machine Prediction:
Predicted Class: 1
Class Probabilities: [0.04337279 0.95662721]
```

Prediction: Use Case 2

```
# Call the prediction function
predict_manual_input()
```

```
--- Manual Prediction ---
Predictions:
Logistic Regression Prediction:
Predicted Class: 1
Class Probabilities: [6.90558721e-13 1.00000000e+00]

Random Forest Prediction:
Predicted Class: 1
Class Probabilities: [0. 1.]

Support Vector Machine Prediction:
Predicted Class: 1
Class Probabilities: [6.91393393e-12 1.00000000e+00]
```

Figure 12: Sample Prediction

These were tested by trained models using manually input values for feature values to simulate real-world application scenarios. This has, in essence, looked at real ad hoc data processing at the system level and predictive ability. Each model generated predictions that support each with probabilities for whether they were benign or ransomware.

- Random Forest was giving the best predictions all along, consistent with its robust performance during model evaluation.
- Logistic Regression and SVM made reasonable predictions, sometimes failing to recognize borderline cases.

Manual testing of the predictions was done to verify that the system was ready for deployment into the real world, enabling it to be adaptive and accurate when receiving data inputs that are unexpected or user-provided.

6.4 Discussion

This discussion integrates findings from this research with insights from the literature, underlining the progress made in ransomware detection while considering the gaps and limitations noted in previous studies.

This work also agrees with the literature in highlighting the weaknesses of static signature-based detection methods. As Larocque et al. (2024) indicate, although APSE-like techniques extend traditional techniques by embedding dynamic signatures into them, they are nevertheless limited by resource requirements and challenges of adaptability.

This research enhances these observations with the demonstration of fixed-pattern detection limitations in the handling of polymorphic variants of ransomware. Besides, the outcome of the behaviour-based system discussed in Loco et al. 2024 will share resonance with this article's emphasis on runtime behaviour analysis for identifying zero-day ransomware threats with high precision.

This research supports the transformational role of machine learning, as highlighted by Ispahany et al. (2024). The use of supervised and ensemble methods in this work, most notably Random Forest, has indeed been able to prove their claims of high accuracy and adaptability in real-world applications. Nevertheless, issues in the literature regarding dataset quality and computational overheads (Zahoora et al., 2022) have also been encountered, underlining the need for scalable solutions. The research extends this discussion by assessing and comparing the models for practical deployment, which is a lesser-explored area in previous research. The integration of manual prediction further bridges theoretical insights into real-world applicability by showing adaptability in dynamic environments. This study will add to the evolving landscape of ransomware detection through the validation of findings, addressing critical gaps, and reinforcing the need for robust, scalable, and real-time detection systems.

7 Conclusion and Future Work

It has proposed to address issues in ransomware detection by drawing on the advantages of rich machine learning techniques and featured analysis for developing a modern, real-time ransomware robust detection system. The findings have met the aims of this research by implementing and evaluating Logistic Regression, Random Forest, and Support Vector Machine. Among all, the most useful features checked include DebugSize and ExportSize, which served the most crucial role in setting ransomware apart from benign. The best among them all came out to be the Random Forest model with accuracy of 99.67%, very strongly robust, able to handle complex patterns of ransomware, as also further shown by confusion matrices and ROC Curves.

This research confirms that machine learning, especially ensemble-based models, works within the limitations of traditional signature-based methods. However, scalability, quality of dataset, and computational overhead are major challenges. These results have implications for cybersecurity and provide a way in which real-time ransomware detection systems can be practically deployed across a variety of environments.

However, further work might include the incorporation of adaptive behavioural-based analysis to detect zero-day ransomware. Lightweight models in general require less computational and other resources, which, if applied to constrained IoT devices, will ease the scalability issues; also, inclusion of federated learning will add benefits on the adaptiveness of the model with preservation of data privacy. The potential for commercialization involves a broad set of deployments for the protection against critical infrastructure that provides dynamic real-time defences. Furthermore, hybrid techniques for detection and adaptiveness at various platforms are possible for their extension to provide better leverage in the fast-evolving landscape of ransomware.

References

Yu, R., Li, P., Hu, J., Chen, L., Zhang, L., Qiu, X. and Wang, F., 2024. Ransomware detection using dynamic behavioral profiling: A novel approach for real-time threat mitigation. *Authorea Preprints*. https://d197for5662m48.cloudfront.net/documents/publicationstatus/230292/preprint_pdf/f746de9d3eef268001a8ee3ade56b351.pdf

Limer, A., Abramovich, R., Devereux, G., Ziemniak, P. and Dubois, F., 2024. Automated ransomware detection using dynamic behavior trace profiling. https://d197for5662m48.cloudfront.net/documents/publicationstatus/230017/preprint_pdf/91e2f10578ae839cc5cbae518e5592b1.pdf

LaRocque, A., Gross, G., Lindholm, F., Greco, P., Dupont, B. and Kruger, J., 2024. Effective ransomware detection using autonomous patternbased signature extraction. https://d197for5662m48.cloudfront.net/documents/publicationstatus/229707/preprint_pdf/70b85fc4e6e33e411b32c971b9a0182e.pdf

Anikolova, E., Martins, S., Rozental, D., Fontana, J. and Maier, P., 2024. Ransomware detection through behavioral attack signatures evaluation: A novel machine learning framework for improved accuracy and robustness. Authorea Preprints. https://d197for5662m48.cloudfront.net/documents/publicationstatus/230904/preprint_pdf/fa6b6818096cbf618ff3c195f5dfba09.pdf

Aslan, Ö.A. and Samet, R., 2020. A comprehensive review on malware detection approaches. IEEE access, 8, pp.6249-6271. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8949524

Su, L., Cheng, H., Li, L., Zhang, C., Wang, Y. and Zhao, J., 2024. A novel approach of ransomware detection with dynamic obfuscation signature analysis. https://assets-eu.researchsquare.com/files/rs-5375812/v1_covered_ff901b3a-d649-4355-ae9c-6b8bfb17cd84.pdf?c=1730778664

Olabiyi, W., 2024. Traditional Detection Techniques. https://www.researchgate.net/profile/Winner-Olabiyi/publication/384362928_Traditional_Detection_Techniques_Author_Winner_Olabiyi/links/66f5e69e553d245f9e3ac05b/Traditional-Detection-Techniques-Author-Winner-Olabiyi.pdf

Ispahany, J., Islam, M.R., Islam, M.Z. and Khan, M.A., 2024. Ransomware detection using machine learning: A review, research limitations and future directions. *IEEE Access*. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10521643

Zahoora, U., Khan, A., Rajarajan, M., Khan, S.H., Asam, M. and Jamal, T., 2022. Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier. *Scientific reports*, *12*(1), p.15647. https://www.nature.com/articles/s41598-022-19443-7.pdf

ISLAM, M.Z., 2024. Ransomware Detection Using Machine Learning: A Review, Research Limitations and Future Directions. https://researchoutput.csu.edu.au/ws/portalfiles/portal/480414522/480413277_Published_article.pdf

Urooj, U., Al-rimy, B.A.S., Zainal, A., Ghaleb, F.A. and Rassam, M.A., 2021. Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, *12*(1), p.172. https://www.mdpi.com/2076-3417/12/1/172

Loco, P., Alonso, S., Hartmann, G., Whitmore, J. and McLaughlin, E., 2024. Adaptive behavior-based ransomware detection via dynamic flow signatures. https://assets-eu.researchsquare.com/files/rs-5317374/v1_covered_be9cdf81-73e4-4f2a-b398-db48ab0bf9f8.pdf?c=1729743725

Garter, L., Johnson, C., Brown, A., Miller, W., Davis, M. and Martin, D., 2024. A Novel Approach of Ransomware Detection Using Dynamic Behavior Modeling and Network Pattern Profiling. https://d197for5662m48.cloudfront.net/documents/publicationstatus/230911/preprint_pdf/601aed5b1c31ba5b87514f0281e1dd3f.pdf

Cliford, T., Mendes, L., Olsson, F., Skarsgard, E. and Steinsson, C., 2024. A Novel Method of Ransomware Detection Using Behavior-based Sequence Anomaly Detection. https://d197for5662m48.cloudfront.net/documents/publicationstatus/229662/preprint_pdf/e32051fdd034eb3f7849fc882e8d9356.pdf

Welderman, G., Castellanos, R., Whitacre, A., Montague, F. and Starck, J., 2024. A robust system for ransomware detection using temporal behavior modeling.

https://d197for5662m48.cloudfront.net/documents/publicationstatus/230133/preprint_pdf/a19c320baa31023442d7ea07241e96ee.pdf

Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S. and Alzahrani, M., 2024. Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. *Heliyon*, *10*(1). https://www.cell.com/heliyon/pdf/S2405-8440(23)10782-1.pdf

Balantrapu, S.S., 2021. A Systematic Review Comparative Analysis of Machine Learning Algorithms for Malware Classification. *International Scientific Journal for Research*, *3*(3), pp.1-29. https://isjr.co.in/index.php/ISJR/article/view/251

Koyirar, W., Harris, B., Williams, J., Moreno, A. and Davis, E., 2024. Efficient ransomware detection through process memory analysis in operating systems. *Authorea Preprints*. https://d197for5662m48.cloudfront.net/documents/publicationstatus/226397/preprint_pdf/da5b9abf6a0b29e7e2634d0401d6460c.pdf

Ganfure, G.O., Wu, C.F., Chang, Y.H. and Shih, W.K., 2022. Deepware: Imaging performance counters with deep learning to detect ransomware. *IEEE Transactions on Computers*, *72*(3), pp.600-613. https://www.researchgate.net/profile/Gaddisa-Olani/publication/360432025_DeepWare_Imaging_Performance_Counters_with_Deep_Learning_to_Detect_Ransomware/links/627753822f9ccf58eb3701bd/DeepWare-Imaging-Performance-Counters-with-Deep-Learning-to-Detect-Ransomware.pdf

Malik, M.I., Ibrahim, A., Hannay, P. and Sikos, L.F., 2023. Developing resilient cyber-physical systems: a review of state-of-the-art malware detection approaches, gaps, and future directions. *Computers*, *12*(4), p.79. https://www.mdpi.com/2073-431X/12/4/79

Qureshi, S.U., He, J., Tunio, S., Zhu, N., Nazir, A., Wajahat, A., Ullah, F. and Wadud, A., 2024. Systematic review of deep learning solutions for malware detection and forensic analysis in IoT. *Journal of King Saud University-Computer and Information Sciences*, p.102164. https://www.sciencedirect.com/science/article/pii/S1319157824002532

Benmalek, M., 2024. Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges. *Internet of Things and Cyber-Physical Systems*. https://www.sciencedirect.com/science/article/pii/S2667345223000561

Botacin, M., Ceschin, F., Sun, R., Oliveira, D. and Grégio, A., 2021. Challenges and pitfalls in malware research. Computers & Security, 106, p.102287. https://www.sciencedirect.com/science/article/abs/pii/S0167404821001115