Intrusion Detection System for Ransomware in Network Traffic Using Supervised Machine Learning

Ziad Almulla, Moath Alamri, Mounir Frikha
Department of Computer Networks and Communications-College of Computer Sciences and Information Technology,
King Faisal University, Al-Ahsa, 31982, Saudi Arabia

***Abstract*—Ransomware is one of the most dangerous cyber threats today, as it can disrupt systems and cause serious financial losses. Traditional detection methods often fail to catch newer attacks because they can hide within normal network traffic. In this study, we used machine learning to detect ransomware based on network data. We tested four models Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting using the UGRansome dataset, both before and after balancing it with SMOTE. The Decision Tree model gave the best results, achieving 99.40% accuracy, 98.0% precision, 99.90% recall, and an AUC-ROC of 99.95%. We also found that protocol flags and network flow features played a key role in detecting attacks. Overall, using tree-based models with balanced data proved to be a simple and effective way to build a real-time ransomware detection system.**

***Keywords*—Ransomware detection; machine learning; IDS; network traffic analysis; real-time detection**

ABBREVIATIONS

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
API	Application Programming Interface
AUC	Area Under the Curve
BTC	Bitcoin
CFS	Correlation-Based Feature Selection
CNN	Convolutional Neural Network
DL	Deep Learning
DPI	Deep Packet Inspection
DT	Decision Tree
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GB	Gradient Boosting
HIDS	Host-based Intrusion Detection System
IDS	Intrusion Detection System
IoC	Indicator of Compromise
IP	Internet Protocol
MLP	Multi-Layer Perceptron
NIDS	Network-based Intrusion Detection System
PCA	Principal Component Analysis
PCAP	Packet Capture
RaaS	Ransomware-as-a-Service
RDP	Remote Desktop Protocol
RF	Random Forest
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic
SAE	Stacked Autoencoder
SIEM	Security Information and Event Management
SMB	Server Message Block
SMOTE	Synthetic Minority Over-sampling Technique
SSH	Secure Shell
SSL	Secure Sockets Layer
VPN	Virtual Private Network

I. INTRODUCTION

Intrusion Detection Systems (IDS) are one of the security controls that plays an important role in detecting ransomware attacks in network traffic and preventing it, positively affecting the cybersecurity of organizations. As ransomware has become more advanced in scale and sophistication, traditional countermeasures such as signature-based detection are no longer useful in the face of polymorphic and zero-day ransomware that is evolving to become a challenge to detect [1]. More recent developments in machine learning (ML) have shown great potential to improve the accuracy, flexibility, and robustness of IDS, specifically when used in analysis of network traffic behavior as it happens in real-time and behavioral profiling [2]. Using datasets to capture both normal and abnormal traffic patterns, trained supervised ML models can be used to detect malicious activity early, making it possible to implement proactive defense measures against a ransomware attack [3].

Although this has developed, there is a research gap. A significant part of the available literature is devoted to general malware detection or anomaly-based IDS with no particular emphasis on ransomware as a specific and increasingly common cyber threat. Moreover, a great number of studies use older or generic data that is not comprehensive when it comes to the description of the features of contemporary ransomware traffic, restricting the practical usability of their findings. This poses an urgency to conduct a research which assesses supervised ML methods on ransomware-specific datasets that reflect current attack patterns [3].

In this research gap, this study focuses on UGRansome dataset coherent dataset of realistic ransomware traffic traces. Such data allows making a better assessment of supervised ML algorithms in ransomware detection than with traditional malware datasets. Through the examination of the ransomware-specific traffic indicators, the study aims at improving the relevance and applicability of the ML-based IDS solutions in the contemporary cybersecurity environment.

While a number of previous works have used machine learning classifiers on the UGRansome dataset, most of them are focused on the evaluation of individual algorithms or propose complex deep learning architectures without addressing systematically the practical requirements of real-time IDS deployment. The current work develops the state-of-the-art through the following specific contributions:

1) Comprehensive and reproducible benchmarking: Unlike previous research which reports findings on a single pipeline configuration, a rigorous side-by-side comparison of four classifiers in two experimental settings (original imbalanced

and SMOTE balanced), with the same preprocessing, feature sets and evaluation protocols, is given in this work. All hyperparameters, random seeds, and computational timings are documented fully in order to allow for independent replication.

2) Quantified impact of SMOTE on minority-class detection: This research is one of the first studies on the UGRansome dataset to give a detailed, quantitative comparison of the performance of models before and after balancing them with SMOTE, showing an 81% decrease of false negatives for the Decision Tree model with a marginal sacrifice on precision, which is 1.36%, a trade-off that was explicitly justified for cybersecurity applications where missed attacks have asymmetric risk.

3) Statistical feature validation: In addition to basic feature importance rankings, this study reports correlation coefficients, mutual information scores, F-statistics, and p-values for all 13 predictive features, which provide some statistical evidence of feature relevance. Furthermore, a systematic comparison of all features with those of Genetic Algorithm (GA) and Correlation-Based Feature Selection (CFS) subsets shows that the compact feature set is already optimal, which is not suitable for dimensionality reduction.

4) Real-time deployment feasibility analysis: This work makes a unique report of the inference latency (4 ms per batch) and throughput (7.5 million samples/second) for the Decision Tree model along with the training times for all classifiers providing concrete proof that the proposed framework meets the computational needs of integration into SIEM systems or in-line NIDS appliances.

II. Background

Intrusion Detection Systems (IDS) are essential for detecting and eliminating transforming ransomware threats in an efficient manner. The ability of IDS to adapt and learn from new data is essential for maintaining their effectiveness in ransomware attacks [4]. The capacity of IDS to learn and update as new data is crucial to their effectiveness against more and more sophisticated ransomware attacks [5] [6].

The resistance and flexibility focus will ensure that the organizations remain at advantageous position to accommodate the complexity of ransomware attacks and maintain their cybersecurity frameworks in the event of future risks and provision of organizational properties. This kind of proactive solution not only gives rise to improved defense systems of organizations, but also to the establishment of a culture of preparedness and reaction to future ransomware attacks [7], [8].

The constantly evolving ransomware requires the proactive approach to the development of Intrusion Detection Systems that must go in tandem with the emerging threats and safeguard the organizational assets. It is a proactive attempt to not just increase the protection mechanisms of organizations, but it also establishes the culture of resiliency and preparedness against future attacks by ransomware [5].

A. Evolution of IDS

The ongoing evolution of IDS reflects the urgent need for organizations to implement adaptive strategies that can effectively counteract the dynamic nature of ransomware threats. In conclusion, the integration of adaptive strategies within IDS is essential for effectively combating the ever-evolving ransomware landscape and ensuring organizational resilience [9].

A combination of advanced machine learning methods with real-time monitoring will have a crucial role in improving the effectiveness of IDS against ransomware attacks and will allow organizations to respond promptly to any crisis of a new ransomware and improve their overall cybersecurity resilience [10].

B. Types of Intrusion Detection Systems

As shown in Fig. 1, there are two primary categories of IDS, Network-based Intrusion Detection Systems (NIDS) and Host-based Intrusion Detection systems (HIDS), which have a different monitoring scope, detection methods, and contribute to ransomware protection in different ways [9].

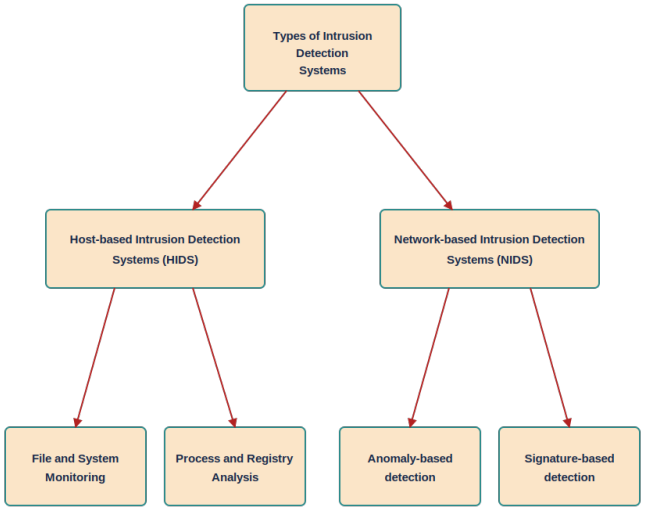

Fig. 1. Types of Intrusion Detection Systems.

1) Network-based Intrusion Detection Systems (NIDS): Network-based Intrusion Detection Systems (NIDS) are installed at strategic locations within a network, e.g., in a gateway or router, to monitor and examine network traffic flowing through the router in real-time. Through analysis of packet data, NIDS is able to identify suspicious patterns or anomalies in different types of attacks such as ransomware, before such attacks hit endpoints. NIDS generally use an implementation of a mixture of signature-based and anomaly-based detection strategies:

a) Signature-based detection: It can be used to identify malicious activity based on signature-known malicious signatures, which is the comparison of the network traffic with previous known malicious patterns. Nevertheless such an approach might not identify new or quickly developing ransomware variants that do not correspond to existing signatures [11], [12].

b) Anomaly-based detection: It creates a normal network behavior baseline, and detects any significant deviations

like uncharacteristic encryption traffic, unusual file transfers, or sudden bursts of data flows-typical ransomware activity indicators [13].

More recent developments have introduced machine learning and deep learning models into NIDS, allowing detection of advanced ransomware campaigns, and zero-day threats by detecting subtle and previously unknown traffic deviations. Architectures that use multi-stage with supervised and unsupervised models have shown improved accuracy and less false positive when it comes to identifying emerging ransomware attacks [14].

Snort and Suricata are practical examples of NIDS implementations in enterprise (as well as in small-to-medium enterprise) environments, with Suricata demonstrating that it is possible to deploy an IDS even in resource-constrained environments [12].

Network-based IDS are especially useful in detecting ransomware at an early stage of network activity, including initial infection stages, command and control traffic, or other data exfiltration efforts, thus facilitating quick containment measures [15].

2) Host-Based Intrusion Detection Systems (HIDS): Host-Based Intrusion Detection Systems (HIDS) are systems that are directly attached to the endpoints or servers and they track the activities within the internal system to identify any classified threats. HIDS normally examine operating system logs, file integrity, process behaviors and registry changes:

a) File and system monitoring: HIDS is capable of detecting ransomware-related operations, including mass file changes, unauthorized file access, or suspicious encryption activity that could be an indication of an attack [13].

b) Process and registry analysis: With HIDS checking newborn processes or unordinary activity in critical Windows registry keys, it is possible to detect instances of ransomware to create persistence or alter system settings [16].

The ability to learn the behavioral baselines of legitimate user and system processes to classify ransomware is further extended due to machine learning-enhanced HIDS. Any deviation, e.g., unauthorized encryption operations or ransom note creation initiates alerts or automated remediation measures. This host-centric approach allows HIDS to identify ransomware even when it tries to avoid network-based detection, particularly when ransomware transfers itself over encrypted networks or when it opts to use fileless attack vectors [16].

HIDS are also crucial in the implementation and distribution stages of ransomware, since it is capable of detecting malicious code that NIDS might fail to identify because of network obfuscation or encryption.

C. Ransomware: An Overview

1) Definition and Characteristics of Ransomware: Ransomware is a unique type of malware that tries to deprive victims from access to their information or systems, usually by encrypting the files and asking a ransom–often in the form of cryptocurrencies like Bitcoin–as a way to unlock access [17]. They are divided into two major categories: locker ransomware which denies access to the user interface of the system and crypto-ransomware which encrypts user files with strong algorithms making them data restoration virtually impossible without the decryption key provided by the attacker [17], [18].

Crypto-ransomware is often considered as the most common and more dangerous threat because it is based on more advanced cryptographic algorithms and the financial motivation it provides to cybercriminals [18].

The ransomware of today is often highly evasive (polymorphism, metamorphism, and packing) and can be used to come up with many different versions and thus makes it difficult to detect [17]. Furthermore, the emergence of the Ransomware-as-a-Service (RaaS) business model has allowed regular individuals to engage in ransomware operations and this has led to the significant growth rate in the number of incidents and financial losses globally [16], [19].

D. Impact of Ransomware on Organizations

The effects of ransomware attacks on organizations are multidimensional, and they include direct economic and operational impacts, reputational damage, and possibly serious social or safety risks [17], [16].

As an example, the WannaCry (2017) and Colonial Pipeline (2021) attacks resulted in massive business disruptions, the crippling of critical services, and caused a significant disruption in society because of the stoppage of several key processes, including healthcare services and fuel supply [17].

The cost of paying a ransom is not the only thing expense organizations can experience due to ransomware, as there is also the cost associated with downtime, data recovery, fines imposed by the regulations, and loss of customer trust [17], [16]. These attacks pose a safety risk and in certain industries they are even life-threatening, particularly critical infrastructure and healthcare [17].

The opportunity nature of ransomware and the increasing threat environment, e.g., the COVID-19 pandemic, is an additional example of its opportunistic character and the emergence of a new threat environment [16].

E. Ransomware Attack Vectors

Ransomware is based on various attack vectors in order to access target networks first and spread through environments [17], [18].

1) Phishing: Social engineering tricks are used to lure the user into opening harmful attachments or clicking on harmful links of well-crafted emails that appear to be a trusted source. They are also one of the most frequently observed vectors in reality [17].

2) Leveraging remote access and vulnerabilities: Attackers exploit weak credentials or software vulnerabilities to attack unpatched internet-facing services (Remote Desktop Protocol (RDP) or VPN gateways) to access organizational IT or operational technology (OT). This threat is enhanced by automated vulnerability scanning that is driven by AI [17], [16].

3) Hijacked third parties and chains: Ransomware pirates are abusing trusted third parties, including managed service providers, to spread malware laterally to several organizations [17].

4) Physical infection: USB drives and malicious insiders also poses less frequent, but equally serious threats in a highly-secured or air-gapped environment [17].

5) Network propagation: Modern ransomware has worm-like or sideways movement functionality, rapidly traversing domains as it takes advantage of techniques like credential theft, privilege escalation, and open SMB (Server Message Block) shares [17].

F. Network Traffic Analysis

1) Importance of network traffic in cybersecurity: Network traffic analysis is known as detective control in cybersecurity, because it gives a full inspection of traffic passing through a network allowing identification and investigation of cyber threats, such as ransomware attacks. Through packet analysis of network packets and their interactions, security analysts can detect malicious activities, command-and-control communications, data exfiltration efforts and other threats that anti-virus programs would not detect using traditional signature-based detection techniques. It is especially important considering the changing level of sophistication of malware and ransomware attacks that cannot be detected by traditional protection mechanisms [20].

Network traffic analysis facilitates the early detection of indicators of compromise (IOCs) in the form of infected file hashes, IP addresses, domain names, and abnormal traffic patterns, among others, crucial to an immediate incident response and mitigation [21]. Additionally, network traffic analysis can give geographic contextualization of cyber threats as they could map the IP origins and destinations, thereby showing the worldwide dissemination and dynamic nature of malware campaigns [20]. This dimension can provide a crucial viewpoint to the perception of the threat landscape outside of technical signatures, which will further contribute to the strategic response to ransomware and other network-based attacks. A number of methods are used to profile network traffic at different levels and granularity:

a) Deep Packet Inspection (DPI): DPI is a process whereby both the packet headers and the payload are examined exhaustively to identify anomalies, signatures, or suspicious content in the network flows, which helps to identify malware activities, unauthorized data transfer, and efforts to exploit vulnerabilities in the network [22]. DPI is more useful than normal packet inspection but it also needs a lot of processing and complex analysis programs like Wireshark [20].

b) Packet Capture (PCAP) analysis: This is used to obtain raw network information that can be analyzed offline or in real-time to reveal specifics about communication patterns between nodes, protocols, contents of messages, and other possible signs of a ransomware infection [23].

c) Social Network Analysis (SNA): Applied alongside PCAP data, SNA can display communication relations and key participants in a network and enhance the threat attribution process and aid in detecting coordinated ransomware-related campaigns [23].

- Entropy-Based Traffic Analysis: This is metric of packet sizes, time between packets and source to destination, to identify the ransomware activity by the entropy changes or unusual deviations by measuring increments or aberrant changes in the entropy [24].
- Machine Learning Classification: Decision Trees, Random Forests and SVM are examples of supervised learning algorithms that have been trained using features identified by inspecting network traffic, such as statistics, and entropy. A differentiate between benign and ransomware-related traffic with high precision, such approaches help in the removal of the shortcoming of the traditional identification whose focus is on the behavioral patterns [24].

These methods assist in the elimination of the weaknesses of the traditional detection that is centered on the patterns of behavior.

2) Challenges in Network Traffic Analysis: Network traffic analysis may be hard to control, bandwidth, overload, and necessity of tracking traffic in data networks. The issues in network traffic analysis include the issues in network configurations, bandwidth changes, bandwidth bottlenecks and the need to monitor traffic in data networks

We will highlight the most common challenges in the network traffic analysis:

- Encryption and Anonymity: The encrypted traffic deprives any capability of DPI and signature-based mechanisms to look at payloads. Similarly, the anonymization (e.g., TOR) and IP spoofing conceal the actual origins of traffic and complicate the attribution and detection even further.
- False Positives and Thresholding: Anomaly and entropy-based systems must compromise sensitivity and specificity. A smaller detection threshold will be detected more often, and potentially too high detection thresholds will be missed by minor ransomware characters. This has to be defeated by dynamic thresholding and adaptive learning mechanisms.
- Resource Constraints: DPI and real-time traffic analysis are extremely resource-intensive in CPU, memory and network infrastructure. To keep up without endangering as bottlenecks, efficient algorithms, hardware accelerator and modular architecture are required.
- Threats and Evasion Methods: The malware and the ransomware are constantly under development and will use polymorphic code, encrypted communication channels, and stealth functions. The detection systems will have to match this through frequent updates, the mixture of the standard detection methods and machine learning to be fruitful [22], [24].

G. Machine Learning and Intrusion Detection system

Supervised ML finds wide usage in (IDS) to detect the ransomware attack at different stages, such as pre-encryption

phase, execution phase, and communication phase, as shown in Fig. 2.

For ML, models are run on the sequences of API calls, system calls, and file system interactions to extract ransomware behavior characteristics, which can be observed in time before it is too late and irreversible harm is caused. The classification of ML classifiers on dynamic runtime detection accuracy such as the random forest and SVM when used on realistic datasets.

In addition to that, ensemble learning and hybrid methods that combine deep learning feature extraction with conventional classifiers have proven to be more effective in detection performance and increased resistance to changing different ransomware attacks [25].

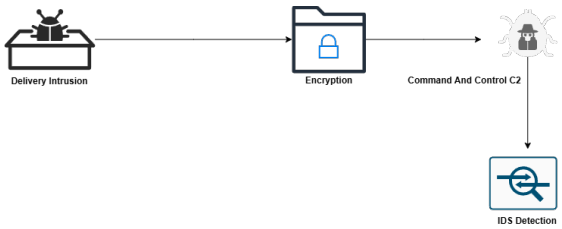

Fig. 2. The ransomware attack life cycle with IDS detection points.

III. Related Work

The related work presents all papers and publications published within 2020 to 2025, as shown in Fig. 3, that are analyzed in this work, with the hightlight on the ransomware detection methods based on the structured datasets of network-traffic data.

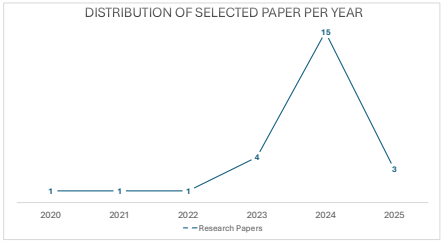

Fig. 3. Distribution of selected paper per year.

Rahman et al. [26] provides an answer to that issue by using a Transformer-based model to identify ransomware, in various types of High-Dimensional Network Data, by fine-tuning the BERT architecture on both structured and non-text features that can be converted to natural language format to do a deep contextual analysis of those features. The authors use the UGRansome dataset (14 attributes, $> 149,000$ network traffic samples) to encode both the categorical and numerical features into text, and then utilize the multi-head attention mechanism of BERT to find the interdependencies between those features. The model had state-of-the-art accuracy, 99.21%, far surpassing that of both state-of-the-art (STI) examples, (LSTM: 98.5%), and ensemble examples (99.0%) with a very low increase in training examples, demonstrating the capability of extracting complex feature relations from Network Telemetry that are similar to prime examples of language models in text-based formats. Their ablation studies indicated that Protocol, Port, Flag and Flow-based features were the best features that provided prediction, while some indicators can lead to overfitting. There were limitations of being computationally expensive for large models of the same type of this transformer, and you risk overfitting, if there is a data leakage. Their main contribution was to demonstrate that transformer models (with carefully structured data) provide superior, generalizable results for ransomware detection in cybersecurity applications with partial data without native text glimpse and extension to other structured security domains of applications.

Azugo et al. [27] developed a Random Forest based model, which uses UGRansome to detect and classify ransomware with a level of accuracy of 96%, in order to differentiate between normal and abnormal network behaviors. The model is created through feature engineering and data encoding, and is able to accurately recognize ransomware that are difficult to identify (specifically EDA and Globe), due to their large financial transaction amounts. The methodology used by Azugo et al., includes the systematic removal of uninformative features, normalization, and the evaluation of feature relevance. In addition, practical results demonstrate the necessity of having high levels of diversity among networks, as well as maintaining datasets in order to ensure continuous performance. Limitations include reliance upon a single dataset, and potential limitations associated with collecting all types of emergent attacks from multiple organizations. Azugo et al. established a baseline for creating quality datasets in current day datasets, and demonstrated that when utilizing a Random Forest model on a quality dataset, it will be capable of identifying both known and unknown ransomware activity.

Nhlapo et al. [28] study network intrusion detection of zero-day and ransomware attacks using a comparative analysis of machine learning models based on the UGRansome dataset. Their objective is to enhance the detection performance than the legacy NIDS based on the evaluation of the improvement of RFC, XGBoost, SVM, Naive Bayes and ensemble learning. Results are categorical, as most will see RFC XGBoost ensemble methods as performing perfectly (100%, perfect metrics), while SVM and NB are performing significantly under the curves (perfect performance as shown). The methodology includes data normalization, feature selection and comprehensive multi-algortithmic data testing. The study brings an underlined realization of the limitations of traditional methods towards legacy data, and an advocate for updated datasets, diversity in datasets. Limitations are recognized regarding more generalisability and practical impact of class balancing and feature selection. The main part of the contribution is a strict empirical validation in which using ensemble approaches and tree-based algorithms under the condition of providing them with well-structured modern datasets help us to perform far superior to conventional classifiers regarding the ability to detect both the known and novel attacks in a real-time.

Chaudhary et al. [25] proposed comparative study of decision tree, SVM and Multi-layer perceptron (MLP) classifiers on the UGRansome datasets in terms of accuracy, precision, recall and F-measure. They use Scikit-learn and TensorFlow models upon extensive preprocessing (e.g. deduplication, normalization, encoding). The decision tree model is an undisputed vector with a high accuracy of 98.83 and 99.41%, depending on other significant measures, but SVM and MLP are in the shadows (accuracy of approximately 62%). The study demonstrates the adequacy of decision trees in the modeling of non-linear relationships, which are essential in the real-world for the discrimination of ransomware. Such limitations are the extent of individual dataset, over fitting or leakage of data. The major contributions are the empirical validation that the decision tree algorithms outperform the more complex Neural or SVM, where we deal with the high dimensional non-linear network security data.

Nkongolo, van Deventer, and Kasongo [29] present the UGRansome1819 dataset - a modern and large-scale dataset has been created for anomaly detection and zero-day threats analysis by bundling and labeling attacks from the UGR16 and ransomware-specific data. The goal is to aid in the evaluation of machine learning using realistic attack/benign class balance. The methodology outlines dataset creation and class legalities together with early benchmarks with ensemble getting used (Random Forest, SVM, Naive Bayes), thoroughly demonstrating that ensemble techniques and Random Forest provide classification charges better than existing IDS benchmarks, and can reduce false positives. Key findings include better detection of advanced persistent threats, cyclo-stationary patterns, and better generalization. In this study, the limitation of dataset coverage and the imbalance of classes (NP-Hard) This dataset will serve as an up-to-date and publicly available benchmark against which future research into ML IDS techniques can be compared, enabling accurate detection of a threat and comparison of models.

Alraizza and Algarni [30] provide a comprehensive overview of Machine Learning-based Ransomware Detection, including a summary of Ransomware Narai detection studies and trends from 2017 to 2022 that have categorized these studies based on static/dynamic features, detection techniques and evaluation metrics. The authors discuss many of the challenges associated with selecting appropriate features for use within models; the need to balance datasets and the need to develop models that are resistant to high false positive rates and also resilient to adversarial attacks. In this study that several machine learning techniques were tested and reported to be effective in detecting ransomware, including Random Forest, Decision Tree, SVM and Neural approaches; however, they indicate each approach has its own trade-offs (for example: one method had accuracy in data detection of over 99%; but it was limited in terms of dataset and generalization). The authors note that due to the rapidly evolving nature of ransomware attack vectors, there is a need for developing real-time adaptive processes to detect such attacks. Additionally, the authors report that their study provides a list of significant obstacles for developing successful ransomware detection models.

Fan et al. [31] introduces TEVEInet, a deep learning-based anomaly detection model which is based on two architectures, Transformer and Variational Autoencoder (VAE) model architecture, to create user behavior and a robust feature extraction framework based on training on the UGRansome data set. The method is to encode the network telemetry and behavior-based features (after the numeric/categorical conversion and normalization), to embed it for learning in the sequence learning and latent space learning. TEVEInet outperforms DNN, CNN and their hybrid baselines by achieving F1-scores, as high as 99.26%, and sophisticated clustering separation on UGRansome. Ablation studies show the need for both the transformer attention and the latent modeling to hold discriminative power. Limitations are a consequence of the number of parameters and optimization complexity, which brings up scaling questions to resource-constrained environments. The contribution consists of the empirical validation that deep neural architectures with both sequential and latent representations, named hybrids, can establish new benchmark for anomaly detection related to ransomware. The contribution is the empirical validation of deep neural architectures capable to set the new standards in case of anomaly detection, applicable not only to ransomware.

Zahra et al. [32] examines ensemble learning capabilities of detecting anomaly intrusion and zero-day attacks in cloud environments against UGRansome dataset assets using approaches that include Random Forest, SVM, Naive Bayes, ensemble stacking and genetic algorithm optimization. Preprocessing like label encoding and normalization AWS SageMaker supports Scalable model deployment. Genetic algorithm feature selection leads to improve computational efficiency and decrease overfitting problems (Random Forest and ensemble methods up to 99.6% accuracy). The work validates the benefit of hybrid feature selection and ensemble techniques, and at the same time, reveals the opportunity to conduct more research into SVM/Naive Bayes robustness and scalability for large cloud workloads. This study is one of such early examples of advanced, cloud-centric anomaly detection using modern ransomware datasets.

Alhashmi et al. [33] are focused in ransomware detection techniques based on UGRansome data in terms of 6 ML classifiers (Logistic Regression, Decision Tree, Naive Bayes, Random Forest, AdaBoost, and XGBoost classifiers). The models are all tested based on accuracy, precision, recall, F1-score and on computational efficiency. Random Forest (99.37%, Decision Tree (99.42%) and XGBoost 99.48%, are the uncontested ones who received almost perfect scores and can be used in real-time/space. Cautious datasets normalization, class balancing and exploitation of computational speed is a significant deployability measure and is covered under methodology. The results support the use of tree-based mesh and ensemble as the leading candidate strategies in the most effective ransomware detection techniques that can be scaled. The limitations are directed at the degree to which the datasets are representative and also the necessity of testing on a larger quantity of real-world network data. The most important contribution to this study is the practical validation of the get ready, high-confidence ransomware detection environments of specific ML models.

Yan et al. [34] develop two-layer machine learning-based approach to ransomware detection and classification using the UGRansomeware dataset. The research focus on solving the problems both accurately and detecting wide ransomware attack and its accurate family classification which is important

to identify exact mitigations for evolving attack threats. The dataset ranges over 149,043 network traces in 14 features of both numerical and categorical data. The methodology is based on stacked ensemble model in the first layer consisting of six classifiers: Gaussian Naive Bayes Classifier, K-Nearest Neighbors, Decision Tree, Logistic Regression, Multi-Layer Perceptron and Stochastic Gradient Descent Classifier to achieve overall detection accuracy of 98.22% with nearly 98% precision and recall values for binary classification (Normal vs ransomware attack). The second layer applies LightGBM to GPCs for classification of detected instances of ransomware into 14 different categories with an accuracy range from 74.9 to 99.1%, with varying lower performance for less common categories, which shows challenges related to class imbalance and within-family similarity issue. Combining Pearson correlation and decision tree importance, 12 important features are selected, based on the optimization of model input. Evaluation metrics-accuracy, precision, recall and F1-score are shown emphasizing good effects of the first layer with moderate effects of the second layer that needs to be improved.

Wa et al. [35] introduces a deep learning architecture for enhancing ransomware detection using network traffic by integrating LSTM classification and stacked autoencoder (SAE) feature selection. In particular, the purpose is to provide improved classification of ransomware and anomalies in the UGRansome dataset. The goal of this work is to identify a method that will effectively classify ransomware in large datasets of cyclostationary data that are of high dimensionality. The researchers begin their methodological process with preprocessing and normalizing the data, followed by use of the SAE to perform self-determined feature extraction that uses a previously trained LSTM classifier. The performance of the SAE-LSTM architecture was evaluated via accuracy, precision, recall and F1-score as compared to XGBoost. The results show that the SAE-LSTM architecture has an accuracy of 98.5% and demonstrates that some ransomware variants that occur less frequently (i.e., NoobCrypt, DMALocker) may have a larger financial impact than other variants. Some limitations of the SAE-LSTM architecture include the current unbalanced nature of the data set, the need for additional feature engineering and the possibility that the model may not generalize well across a variety of ransomware samples.

Nkongolo et al. [36] utilizes ensemble-based approaches optimized by Recursive Feature Elimination (RFE) to identify Zero-Day Threats, and ultimately aid in the prevention of those threats. Nkongolo also has a goal of enhancing UGRansome's utility for Real-Time Anomaly Detection, and to provide input to both policy and rulemaking for IDS. The methodology utilized by Nkongolo combines Gradient Boosting (GB) and Random Forest (RF) base models, with Naive Bayes (NB) as the Meta-Blender, and RFE is used to determine the most critical features from the data set, so that the IDS may effectively block malicious network traffic. Accuracy, Precision, Recall, and F1-Score are the key metrics used in this study; the ensemble framework achieved a balanced accuracy of 97%, which is better than legacy systems, and specifically was able to achieve a high degree of accuracy in identifying Zero-Day Vulnerabilities. In addition to achieving high degrees of accuracy, Nkongolo was also able to demonstrate the framework's ability to block greater than 100kbps of suspect traffic; Secure Shell (SSH) Attacks were particularly noteworthy in terms of mitigation. However, there are several notable limitations to the study conducted by Nkongolo including: the lack of a real-time IDS implementation; the non-comparative evaluation of all feature selection methods; and, the limited scope of the study to the tested dataset.

Sharath et al. [37] presents an ensemble method that incorporates AdaBoost, Random Forest, and Naive Bayes to improve the detection of ransomware in various security fields. The primary purpose of this study is to address the limitations of current detection methods due to the continuously changing nature of ransomware and limitations in available data sets utilizing the UGRansome data set as the basis for experimentation. Kumar et al. applied each individual classifier separately and then ensembled the results to utilize the strength of each individual classifier to evaluate the performance of the ensemble model using accuracy, precision, recall, and F1-score. The ensemble model achieved a detection accuracy of 96% and was superior to the results obtained from using the individual classifiers alone; the ensemble model demonstrated the capability to decrease the number of false positives generated during the detection process and increase the overall efficiency of the IDS. The authors concluded that ensemble methods have a higher degree of resistance to sophisticated types of ransomware however additional research is needed to fully realize the potential of ensemble methods in terms of real time detection capabilities and scalability to detect future novel forms of ransomware.

Torky et al. [38] investigated the use of ensemble-based machine learning techniques for anomaly detection in enterprise system transactional data, with the ultimate goal of increasing detection accuracy and robustness and thereby decreasing the likelihood of financial loss and disrupting services. Torky examined multiple ensemble-based methodologies (bagging, boosting, stacking) along with stochastic models and Support Vector Machine (SVM) utilizing a dataset created from operational payment systems; accuracy and robustness were the two primary metrics examined. The SVM model exhibited the greatest level of detection performance at an accuracy rate of 80.2%; ensemble methodologies enhanced the reliability of classification and decreased variance. A major finding of the study is that it emphasizes the significance of labeled data, the inability of models to recognize previously unknown anomalies, and the need for substantial computing power. The primary drawback of the artifact of the study is the reliance upon supervised methodologies and labeled datasets, which limits the ability to adapt to new or evolving attack vectors.

Mohamed, et al. [39] constructed a hybrid neural network model that could be applied to detect new and unidentified zero-day exploits through a combination of four sub-networks: Adaptive WavePCA-Autoencoder (AWPA), Meta-Attention Transformer Autoencoder; Genetic Mongoose-Chameleon Optimization, and Adaptive Hybrid Exploit Detection Network. The overall aim of this research study is to enhance the performance of detecting advanced network attacks that are of zero-day in the dynamic networks. The implementation of the proposed hybrid network is to develop and train the hybrid network on a few diverse datasets, e.g., UGRansome and extract a vast range of feature values of the data sets with the aid of a large number of advanced optimization algorithms; followed

by testing the effectiveness of the proposed hybrid network by testing the accuracy, precision, recall and computational cost of the hybrid network. Study findings revealed that the hybrid model recorded more detection precision and recall rate as compared to the traditional autoencer and tree-based models and also incurred lower computational costs. The hybrid model was also identified to have certain limitations such as the complexity of the model training, the fact that much large powering computing resources are required and continuous large-scale inputs into the hybrid model are necessary in order to keep the model operating efficiently.

Mutmobo et al. [40] extended the utilization of the UGRansome dataset to employ machine learning automation to enhance security of the blockchain network. The authors identified that blockchain networks are susceptible to ransomware and zero-day attacks, and, therefore, they automated the assessment of numerous models to determine real-time detection. The key characteristics of the data- timestamps, network protocols, the financial transactions with Bitcoin, port information, and address flows were critical to the detection of attack patterns occurring on the blockchain activity. By using Lazy Predict, the authors were able to automatize benchmarking of the decision trees, random forest, light gradient boosting machines (LGBM), and nu-support vector classifier (NuSVC) models and compare the models against each other by their accuracy and F1-score and training time time. The findings showed that decision tree and extra tree classifier had 99% accuracy and with extremely low training periods (0.12 -0.30 seconds), it was specifically applicable in a blockchain based setting where quick response was essential. Also, the models successfully determined zero-day exploits (designated as anomaly "0") and ransomware signatures (1), and concluded whether there are any correlations between Bitcoin spikes and ransomware financial activity. As it was proved by the authors, the implementation of the network flow analysis and financial characteristics significantly optimized the possibility to detect ransomware in blockchain systems in real-time. Nonetheless, the authors recognize that the study had certain limitations like lack of ROC-based performance comparisons because of the limitations in the library, and lack of explicit optimization of hyperparameters.

Al-Binahmed et al. [41] investigated how deep learning methods can be used to detect Android ransomware attacks based on network traffic patterns. The authors used a publicly accessible dataset in Kaggle, having 392,045 records, in order to differentiate between benign traffic and ten ransomware variants. After data preprocessing and feature selection to deal with class imbalance and dimensionality questions, two experiments were performed with all features, as well as most relevant features (19). The algorithms that were evaluated were Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), an ensemble of these three algorithms, Feedforward Neural Network (FNN) and TabNet. The authors assessed the performance of the algorithms in terms of the accuracy, precision, recall, and F1-score. DT model had the best accuracy (97.24%), precision (98.50) and F1-score (98.45) to detect ransomware, but the SVM algorithm had 100% recall, which implies that DT is an efficient way to get fast performance in such case. The authors mention that their research had a number of limitations, which comprise a simulated environment, the fact that the models could not be interpreted, and that it was required to create lightweight models to be used on resource-constrained mobile devices. The authors indicate that they would consider including sensor data and the study of unsupervised algorithms as the possible direction of future studies to make their ransomware detection system more robust. Overall, the authors have shown that the potential of the use of automated machine learning/deep learning to recognize Android ransomware attacks is very high, but they also note that there is still a need to validate and adapt it to successful attacks in the field.

Srivastava et al. [42] present a two phase ransomware detect framework which incorporated supervised machine learning combined with a statistical Hidden Markov Model with Gaussian Emission (GMM-HMM) to ensure the detecting of malicious network behavior with more accuracy and less false positives. The dataset used by them is the ISOT ransomware set that included traffic logs of 669 ransomware examples belonging to a broad array of families and 103 malicious Windows programs. The authors have extracted and dimensionally reduced the features using Principal Component Analysis (PCA) and then trained and tested a number of machine learning (ML) models, choosing the Decision Tree model due to its high accuracy (99.9%). A prediction made by the Decision Tree model was then inputted in the GMM-HMM, which confirmed the findings of the Decision Tree model and also gave an extra measure of confidence to the classification. They showed the strong detection of ransomware in three experimental conditions benign only, malicious only, and mixed traffic; the GMM-HMM-HMM offered a greater classification rate in all conditions than the ML output alone. The main achievements of the study are the capacity of the framework to detect the known and unknown ransomware samples, and the elimination of the weaknesses associated with the use of individual ML algorithms. The most notable limitation of the research was that they used binary classification (malicious vs. benign) but did not name the specific ransomware families. Future research directions suggested by the authors are the real-time implementation and the creation of classification at the family level.

Manzano et al. [43] perform empirical comparison of the performance of a set of various supervised learning algorithms that are applicable in detecting ransomware on Android devices through network traffic analysis. They made use of the CICAndMal2017 dataset, which was composed of network capture data of ten families of Android ransomware and some benign applications. Before performing the analysis, the authors used the Kendall correlation procedure to pick the most informative features of traffic in order to retain them (in the final analysis, they picked nine critical features). They used the standard metrics to train and assess the Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbor (KNN) classifiers (accuracy, precision, recall, and F1-score). The RF model achieved the most accuracy in the classification of ransomware (96%), and outperformed both DT and KNN; and DT also performed very well in recall (94%). One of the main results was that the chosen time- and packet-based features presented an incredibly precise approach to the discrimination of ransomware and non-malicious traffic. Nevertheless, the authors also state a major limitation of the research in the aspect of preserving the practiceability of the methods in the future against the variants of ransomware because of the deep speed of development of

the attack patterns. The authors also acknowledge that deep learning-based methods should be developed in order to offer better detection of new and unknown samples.

Woo et al. [44] introduce a hybrid ransomware detection system that involves the use of Convolutional Neural Networks (CNN) in extracting features and Random Forests (RF) in classifying network traffic. The authors generated a dataset consisting of benign and ransomware generated network traffic and added synthetic examples to achieve variability in the real world. The strict normalization and segmentation of the data enabled the CNN to find the intricate patterns of raw flows that could be utilized in the classification process by the RF model with robust and low-bias results. The hybrid model was found to have an accuracy of 94.6 and a false positive rate of 2.3 and performed better than the standalone CNN and RF models alongside the traditional ML baselines, such as the SVM and Logistic Regression. The model presented capability of adaptation to novel ransomware, good generalizability, and scalability. Nevertheless, the authors do not ignore the computational complexity of deep feature extraction, the application of synthetic rather than purely real-world data, and even the inability to find the differences between benign anomalies and the real ransomware, especially.

Batlov et al. [45] proposed a hybrid ransomware detection scheme which combines Isolation Forest (IF) anomaly detection and Long Short-Term Memory (LSTM) network sequential modeling of network traffic for ransomware detection. The study was designed with the view of overcoming the limitations of the traditional signature-based systems, which tend to fail against polymorphic or zero-day ransomware. Using a combination of publicly available benign network traffic and ransomware flows simulated in lab, the authors then derived packet-level and flow-level features like size, duration, and intervals from PCAP files and did normalization and augmentation to ensure generalization. The Isolation Forest model first learned a normal behavior of the network and detected the anomalous traffic and sent them to the LSTM model to learn the temporal dependencies that are common in ransomware communication patterns. We tested the performance using the accuracy, precision, recall, F1-score, and the ROC-AUC, in which we obtained 97.5% accuracy, 96.3% precision, 95.8% recall, 96.0% F1-score, and an AUC of 0.982. Results showed that anomaly detection and sequential analysis with the combined method increased detection accuracy and decreased false positive discovery significantly while keeping the system operating in real time with an average latency of 9 ms. The research also pointed out that the variation in packet size and the flow duration were good indicators of ransomware activity. However, the computational cost of processing the dual-model and some false negatives for ransomware closely mimicking benign traffic were the limitations.

Merhban et al. [46] present the implemented ransomware detection system which takes a combination of machine learning algorithms and network traffic analysis to detect ransomware behavior across various environments. The goal of the study was to construct an effective and precise detection mechanism that can be used to distinguish ransomware and benign software based on the flow-level network characteristics. The dataset consisted of 396 samples from 54 ransomware families, and 420 benign samples, which were collected from the Ransomware Tracker and VirusTotal Intelligence platforms. Features like IP Address, port, protocol type, duration and packet/byte count were extracted using TShark and normalized to form training inputs. Several supervised algorithms were tested: Random Forest, Multilayer Perceptron (MLP), J48 Decision Tree, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Bayes Network to find out which model provided the highest accuracy in identification of ransomware. Performance was evaluated by True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, F-measure and Accuracy, where MLP achieved 97.30% TPR rate which is better as compared to other models. The results showed that machine learning algorithms can be used to accurately detect ransomware based on network traffic and Random Forest and J48 performed competitively as well. Comparative analysis with previous works proved that network traffic-based detection outperforms static analysis in terms of detection rate and reliability. However, the limitations are high processing time for complex models and poor performance in large-scale real-time environments.

Brinkely et al. [47] developed an ML-based IDS capable of identifying ransomware attacks from unknown data, which illustrated the ability of ML systems to identify emerging threats as they develop, without having to be defined prior to deployment. The study utilized three different algorithms; RF, SVM and NN, to identify anomalies in the way a system behaves by tracking file system changes and encryption activities, as well as unusual activity in network traffic. A total of 2850 samples were used for the research, which were sourced from various public repositories of cyber security information, as well as local system logs and simulated network traffic, all of which were normalized, segmented and had features extracted prior to being analyzed. PCA and correlation analysis was also conducted to reduce the number of redundant variables and increase the overall generalizability of the models. All of the algorithms were trained and tested via k-fold cross-validation, after which each model was evaluated based upon accuracy, precision, recall and F1 score. The NN produced the most accurate results (92.4%); however, both the RF (91.2%) and SVM (88.7%) maintained very similar levels of performance, with all of the models achieving high F1 scores. These results supported the use of ML for identifying zero-day ransomware attacks, since the ML models were able to identify these attacks based upon learned patterns of abnormal behavior, as opposed to the reliance on established signatures.

Cen et al. [48] developed Zero-Ran Sniff (ZRS). This is a novel early ransomware detection framework which applies the methodology of zero-shot learning (ZSL) to recognize unknown ransomware attacks prior to the point of encryption. The researchers attempted to address some of the shortcomings associated with current signature-based approaches, as well as other dynamic feature-dependent methods which do not support new/unknown ransomware variants. The dataset was obtained from Sgandurra et al. (2016). The dataset contained 582 examples of ransomware along with 942 examples of benign samples and spanned 11 different ransomware families; including: CryptoLocker, Locker, and TeslaCrypt. The static Portable Executable (PE) header attributes were extracted using the Python library pefile in order to allow for the efficient processing of data. There are two stages of the proposed ZRS method: AE-CAL (Autoencoder-based Core

Attribute Learning) is the first phase, where AE-CAL extracts deep semantic representations from previously known classes; SA-CNN-IS (Self-Attentive Convolutional Neural Network Inference Stage) is the second phase where it classifies the representation based on previously seen classes. The model was trained by utilizing Adam Optimizer and activation functions of type ReLU. The performance of the model was evaluated using metrics of Accuracy, Precision, Recall, and F1-Score. Results of experiments showed a high level of accuracy (96.02%), a high degree of precision (91.49%), a very high level of recall (98.47%) and a high level of F1-Score (96.31%) compared to state-of-the-art methods of Random Forest, KNN and AdaBoost, as well as to deep learning baselines such as VGG16 and ResNet. Also the researchers improved recall of their zero-day model by 2.9%, and reduced feature extraction time to 0.3 seconds.

Fevid et al. [49] proposed a new zero-day ransomware detection approach based on analysis of assembly language bytecode and random forest classification of unknown ransomware family types based solely on their bytecode characteristics. The researchers sought to improve the early-stage detection rate of ransomware by analyzing the opcode level behaviors common to all ransomware families. The researchers assembled an extensive dataset from VirusShare comprising over 1,700 samples including the major ransomware families (LockBit, REvil, BlackMatter and DarkSide) along with benign executables for comparative control testing. The researchers used IDA Pro and Radare2 to decompile each executable and generate opcode frequencies, instruction sequences, entropy values and control flow graphs, which were subsequently converted into vectorized feature formats via n-gram tokenization and principal component analysis (PCA). The Random Forest classifier was comprised of 500 decision trees and its performance was assessed against several metrics: Precision, Recall, F1 score and Accuracy. The developed model demonstrated an accuracy of 93.5%, precision of 0.95, recall of 0.94 and F1 score of 0.945 and demonstrated strong generalizability to previously unseen ransomware samples based on a high ROC-AUC curve. The results also indicated that opcode frequency and instruction sequence pattern were significantly more indicative of ransomware behavior than deep learning techniques such as CNNs while providing better performance at a lower computational cost. However, the authors acknowledged several limitations of the developed model, including reliance upon static features, inability to handle polymorphic ransomware and potential false negative results due to high levels of code obfuscation. They suggested utilizing a combination of static and dynamic analysis techniques, representing control flow as graphs and employing hybrid machine learning architectures to develop a future generation of ransomware detection frameworks with greater adaptability, scalability and real-time responsiveness.

A. Summary of the Discussion of Related Work

In summary, Ransomware detection systems can only be built on datasets to be reliable. In order to train and evaluate the ransomware detection models with the help of ML, several publicly available datasets have been extensively used, including CICandMAL 2017, ISOT, VirusShare randoms dataset, UGRansome, UGRansome1819, and UGRansome-2024.

Table I presents an overall comparison of some of the most common ransomware detection datasets. Each dataset is analyzed based on its features, data distribution, and recognized shortcomings. Most of the existing ransomware datasets have the same issue: synthetic data, suffer from a serious class-imbalance problem or lack newer ransomware.

TABLE I. SUMMARY OF THE RANSOMWARE DETECTION DATASETS AND THEIR KEY FEATURES AND LIMITATIONS.

Author(s)	Dataset(s)	Features	Limitation(s)
[39]	UGRansome1819	Logistics and Network flow	Needs more real-world validation
[26]	UGRansome1819	14 network attributes	Potential overfitting
[27]	UGRansome2024	12 selected features	Imbalanced classes
[32]	UGRansome1819	14 attributes including financial features	Imbalance and lack of diverse ransomware
[28]	UGRansome	Multiple network flow	Balance issues
[25]	UGRansome	14 features including protocol and flags	Class imbalance
[34]	UGRansome	14 features network and transaction	Imbalanced ransomware family
[30]	UGRansome1819	Diverse, protocol, payload, time, etc.	Legacy datasets used as benchmarks are outdated
[31]	UGRansome	Multi-type network and behavior features	High dimensionality
[33]	UGRansome	Network and financial features	limitations in diversity and imbalance
[41]	Android Ransomware Dataset	19 selected network traffic features	lacks real-world variety
[42]	ISOT Ransomware Dataset	Network flow and time-based statistics reduced by PCA	Binary labeling only benign/malicious
[43]	CICAndMal2017	9 key timing and packet-size traffic features	Limited representation of new ransomware
[44]	Custom hybrid network dataset	Extracted CNN features s	Ptential false positives on encrypted flows
[45]	Custom dataset	Flow-level features: packet size, duration,etc..	High computational cost for dual models
[46]	VirusTotal Intelligence	IP, ports, protocol type, etck	scalability for large-scale networks
[47]	Public malware repositories	File system changes, and log features	High CPU/memory cost
[48]	Viursshare dataset	PE header static features and Self-Attentive CNN layers	Windows PE structure
[49]	VirusShare (12 ransomware families)	Opcode frequency, and entropy-based attributes	Dynamic behavior capture
[40]	UGRansome1819	Network flow	Outdated for most recent ransomware trends

IV. METHODOLOGY

A. Research Framework

In this study, we propose a supervised machine learning framework for ransomware detection in network traffic. The overall workflow, shown in Fig. 4, follows these stages which are: data collection, preprocessing, feature engineering, classification, evaluation, and validation.

Fig. 4. Overall methodology framework

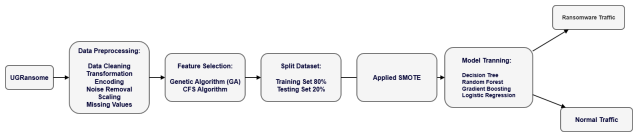

Fig. 5. Overall methodology framework illustrating the six main stages of the proposed ransomware detection pipeline.

B. Experiments

We ran all of our experiments on a machine that had the exact same operating system Ubuntu 22.04, as well as programming language Python 3.10, and also included the hardware Intel Core i9 13th Gen and 64GB DDR5 RAM.

C. Dataset Description and Preprocessing

As shown in Table II, the UGRansome dataset comprises 149,043 network traffic entries of which fourteen (14) numerical and categorical variables have been defined.

TABLE II. SUMMARY OF THE UGRANSOME DATASET

Attribute	Description
Dataset Name	UGRansome (UGRansome1819)
Source	University of Pretoria; combines UGR16 traffic data with ransomware-specific network traces [29]
Total Instances	149,043 network traffic records
Number of Features	14 attributes (13 predictive + 1 target label)
Feature Types	7 categorical (Protocol, Flag, Family, SourceAddress, ExpAddress, IPAddress, ThreatType) and 6 numerical (Time, BTC, USD, Netflow, Port, Clusters)
Class Distribution	Normal: 108,941 (73.09%) | Ransomware: 40,102 (26.91%)
Imbalance Ratio	2.72:1 (Normal : Ransomware)
Missing Values	None detected
Preprocessing	Label encoding for categorical variables; StandardScaler normalization for continuous features; duplicate identifier removal
Class Balancing	SMOTE applied on training set only; 55,072 synthetic ransomware samples generated (0.56 s)
Train/Test Split	80/20 stratified split (119,234 training / 29,809 testing); random seed = 42

This dataset encompasses an array of different network behavior that is representative of both typical activity, as well as ransomware-based activity.

The "Prediction" label has been translated to a binary target value: Normal Network Activity Class 0 and Ransomware Class 1.

About 73% of all instances are classified as normal and approximately 27% as ransomware; these percentages represent a moderate degree of imbalance, which contributed to the decision to apply SMOTE.

Data preprocessing included cleaning the data, converting the categorical variables (Protocol, Flag, Family, SourceAddress, ExpAddress, IPAddress, ThreatType) to encoded format, and normalizing the continuous features using StandardScaler.

No missing values were discovered during the data preprocessing, and duplicate identifier fields were removed, leaving thirteen (13) predictive feature(s).

As shown in the Fig. 5, the implementation steps demonstrate a more detailed view of the data processing and classification.

D. Feature Engineering and Class Balancing

UGRansome database contained 14 attributes covering different characteristics of network traffic, including temporal behavior, type of protocol, financial transactions and threat intelligence. The given attribute that was dropped was the one called Prediction, since it is the target label, and it would cause data leakage to be retained as an input. Following this, the dataset had 13 predictive features.

Among these 13 features, seven of them were categorical (Protocol, Flag, Family, Sender Address, Receiver Address, IP Address and Threat Type). These were coded with labels in order to make them compatible with machine-learning algorithms without expanding the feature space. The other six characteristics (Time, BTC, USD, Netflow\Bytes, Port, and Clusters) were numeric.

The best overall performance was obtained with the full 13-features set at an accuracy of 99.4%, Precision 98.00%, recall 99.9% and a AUC-ROC 99.95%, of the Decision Tree classifier. These findings prove that the features chosen are very informative, non-redundant, and can be used as the most appropriate features to identify ransomware.

Table III presents the statistical validation results for all 13 predictive features.

TABLE III. STATISTICAL VALIDATION OF THE 13 PREDICTIVE FEATURES IN THE UGRANSOME DATASET.

Feature	Correlation	Mutual Information	F-Statistic	p-Value
USD	0.1473	0.4007	9318.68	< 0.001
BTC	0.2426	0.3376	441.38	< 0.001
Port	0.1152	0.0861	27652.33	< 0.001
Flag_encoded	0.0786	0.3016	2005.54	< 0.001
Clusters	0.3005	0.0513	2613.02	< 0.001
Netflow_Bytes	0.1313	0.2497	1280.96	< 0.001
ExpAddress_encoded	0.0474	0.1008	1479.71	< 0.001
SourceAddress_encoded	0.0543	0.0474	936.89	< 0.001
Threats_encoded	0.0077	0.0177	886.57	< 0.001
Time	0.3956	0.0153	17.19	< 0.001
IPaddress_encoded	0.0107	0.0160	827.12	< 0.001
Protocol_encoded	0.0923	0.0140	8.01	0.005
Family_encoded	0.0769	0.0209	336.14	< 0.001

To prevent any kind of data leakage, the data was first divided into training and test sets (20% data as test and 80% data as train) and then SMOTE was applied on the training set only after the split. Likewise, the StandardScaler was only fitted to the training fold, then applied to the training and test folds.

Before oversampling, there are 119,234 records in the training partition (87,153 normal and 32,081 ransomware samples). The training set was completely balanced with 87,153 samples of each class (174,306 total) and synthetic ransomware samples

were generated in 0.56 s with the application of SMOTE. A test set of 29,809 samples was held out and only used for final evaluation.

E. Model Training and Validation

The dataset was divided into two parts by employing an 80/20 stratified split: 119,234 samples for training and 29,809 for validation. All feature scaling occurred post-split (0.03 s).

A fixed random seed (42) was used for all stochastic operations to ensure reproducibility. Four supervised classifiers were trained using `scikit-learn` (v1.3): Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB). The hyperparameters used for each model are listed in Table IV; default scikit-learn values were retained where not specified.

TABLE IV. HYPERPARAMETER CONFIGURATION

Model	Hyperparameters
LR	solver = `lbfgs`, C = 1.0, max_iter = 1000
DT	criterion = `gini`, max_depth = None
RF	n_estimators = 100, max_depth = None
GB	n_estimators = 100, learning_rate = 0.1, max_depth = 3
SMOTE	k_neighbors = 5, sampling_strategy = `auto`

Training times on the balanced data were: DT = 0.37 s, RF = 0.63 s, GB = 13.75 s, LR = 2.07 s. All trained models and preprocessing pipelines were saved for downstream evaluation.

V. EVALUATION METRICS

To evaluate the performance of the developed ransomware detection models, we will use five common evaluation criteria to measure the accuracy of the models' ability to recognize ransomware as well as to differentiate it from normal internet traffic. These five are:

1) Accuracy: It reflects the quality of its predictions, and the model classifies accurately by following the formula given below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2) Precision: This measures how many actual instances of ransomware were detected by the model by following the formula given below:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

3) Recall: It is a measure of how well the model identifies true cases of ransomware from all the possible ransomware instances that exist by following the formula given below:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

4) F1-Score: This metric gives you a single number to assess both the number of False Positives (FP) and False Negatives (FN) of the model by following the formula given below:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

5) AUC-ROC: This value measures how good the model is at separating normal and ransomware distributions into normal and ransomware traffic distributions:

$$\text{AUC-ROC} = \int_0^1 TPR(FPR)\, d(\text{FPR}) \tag{5}$$

Here, TP, TN, FP, and FN refer to the counts of *True Positives*, *True Negatives*, *False Positives*, and *False Negatives*, respectively.

VI. RESULTS AND ANALYSIS

The implementation used `scikit-learn` and `imbalanced-learn` to train models; to perform cross-validation; and to create synthetic samples through SMOTE sampling.

A. Overview of Experimental Evaluation

UGRansome dataset was used to test four trained machine learning algorithms, namely, Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB) to identify ransomware in network traffic.

Two experimental were conducted: first one, imbalanced data and the second, balanced data (with the help of SMOTE). This two-way comparison allowed to fairly compare the effect of class imbalance on detection accuracy, and the use of synthetic oversampling to enhance the detection of minority (ransomware) samples without affecting overall accuracy.

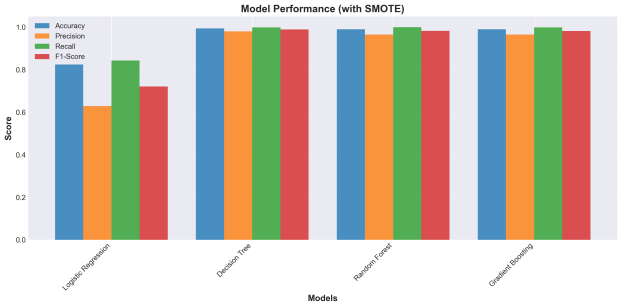

Fig. 6. Performance comparison of all models on the SMOTE-balanced dataset.

As shown in Fig. 6, the Decision Tree had the highest detection accuracy (99.4% and recall (99.9%). On the other hand, Logistic Regression did not perform as well 82.4%, which shows that it is not effective in capturing the complex, non-linear feature interactions that are likely to be in captured encrypted or polymorphic network traffic.

B. Model Performance Comparison

Table V summarizes that the Decision Tree displayed the best overall performance having 99.4% accuracy, 98.0% precision, 99.9% recall, F1-score of 98.9%, and an AUC-ROC of 99.95%. The RF and GB models also had a high level of performance (accuracy of more than 99%), indicating the potential strength of ensemble models.

In contrast, the Logistic Regression obtained an accuracy of 82.4% with an AUC of 89.9%, thus being incompatible with ransomware detection.

TABLE V. PERFORMANCE COMPARISON OF SUPERVISED MODELS (SMOTE-BALANCED DATASET).

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
DT	99.40	98.00	99.90	98.90	99.95
RF	99.00	96.50	100.00	98.20	99.98
Gradient Boosting	99.00	96.60	99.90	98.20	99.97
Logistic Regression	82.40	62.90	84.30	72.10	89.90

C. Confusion Matrix Analysis

Fig. 7 and Fig. 8 demonstrate the model behavior balancing with and without SMOTE. This model before balancing, the model have good results precision, but can't classify many of the ransomware examples, 43 false negatives. When applying SMOTE, this model (DT) had only eight false negatives resulting in an 81% improvement in ransomware detection and very little loss in precision. This enhancement shows the effectiveness of teaching the minority class, in order to teach the model itself, more efficiently rarely attack patterns.

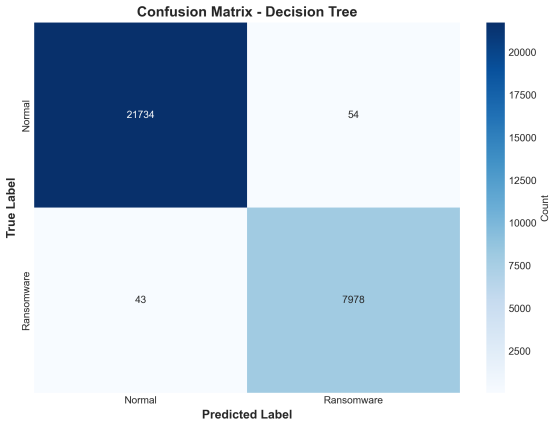

Fig. 7. Confusion matrix of the Decision Tree on the original (unbalanced) dataset.

D. ROC Curve Analysis

The ROC curve illustrated in Fig. 9 and Fig. 10 are almost perfectly classified with AUC of 99.95% or higher with either of the data points. The SMOTE-balanced showed some consistent improvement in the detection performance at lower thresholds with the consequence that fewer false negatives are obtained at no cost to specificity. Conversely, the Logistic Regression model gave a much flatter curve (AUC = 89.9%) that gave a relatively low discriminatory capability between

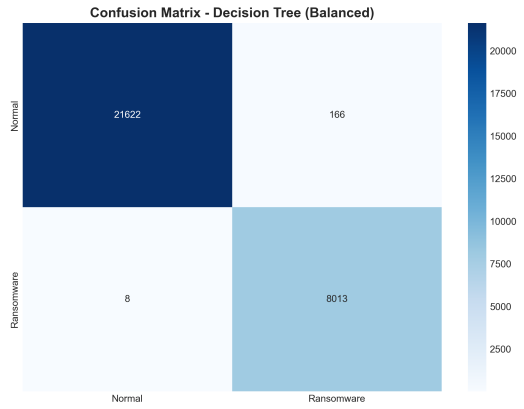

Fig. 8. Confusion matrix of the Decision Tree after SMOTE balancing.

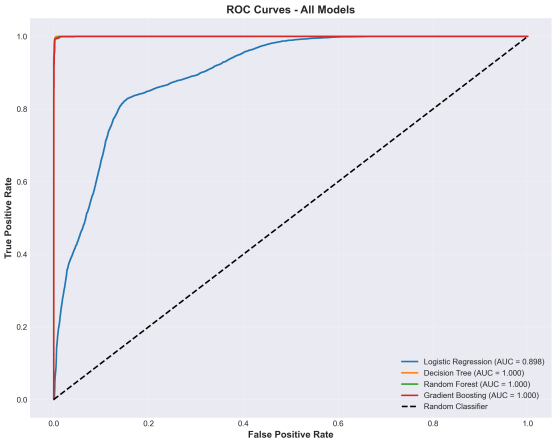

Fig. 9. ROC curves for all models on the unbalanced dataset.

normal and ransomware traffic. To further explore the contribution of various features, two feature selection techniques were compared with the baseline (all 13 attributes): Correlation Based Feature Selection (CFS) and Genetic Algorithm (GA).

E. Comparison of Feature Selection Methods on the Decision Tree Model

As shown in Table VI, it illustrates that the complete feature set was the most accurate and least time-consuming. In fact, CFS substantially decreased dimensionality, it caused a decrease in accuracy by 4.23% and falsely negative results. GA was also able to compete at eight features (99.07% all accuracy) with approximately one second of computation overhead. On the whole, as the dataset already includes a rather small amount of features (13), it made it effective and efficient to take all of them as the ransomware features.

TABLE VI. COMPARISON OF FEATURE SELECTION METHODS ON THE DECISION TREE MODEL.

Method	Features	Acc.	Prec.	Rec.	F1	Total(s)
All Features	13	**99.67**	**99.33**	**99.46**	**99.40**	**0.13**
GA	8	99.07	97.20	99.41	98.29	1.03
CFS	3	95.44	90.43	92.87	91.63	0.37

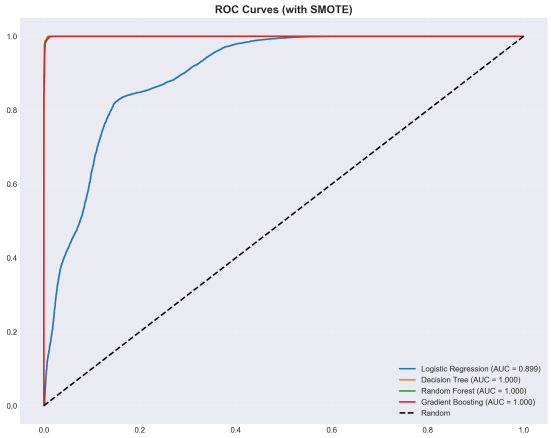

Fig. 10. ROC curves for all models after SMOTE balancing.

F. Comparative Analysis: Original vs. SMOTE Pipeline

Fig. 11 shows comparative analysis. The use of SMOTE decreased precision (-1.36%), accuracy (-0.26%) and recall increased by +0.44This trade-off is positive in the context of cybersecurity, since dedicating more time to identifying attacks is much more important than preventing some additional false signals. This proves that training is strengthened and more trusted in the case of SMOTE-enhanced ransomware detection.

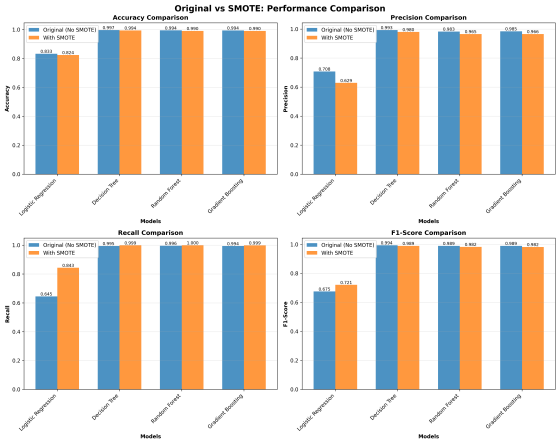

Fig. 11. Comparison of model metrics between original and SMOTE-balanced pipelines.

G. Computational Efficiency

As shown in Fig. 12 and Table VII, Decision Tree train 0.37s to 4ms with a throughput rate of approximately 7.5 million samples per second. Logistic Regression achieved greater throughput (13.7M/s) its accuracy is less than ideal. Random Forest and Gradient Boosting moderate latency (63ms,34ms) and can be fit in real-time security appliances.

H. Summary of Results and Analysis

In short, the Decision Tree model, which was trained on the SMOTE-balanced dataset, has the highest overall balance between precision, recall and processing speed. It presented

TABLE VII. Training and Inference Efficiency of Evaluated Models.

Model	Training Time (s)	Test Time (ms)	Throughput (samples/s)
Decision Tree	0.37	4	7.5M
Random Forest	0.63	63	471K
Gradient Boosting	13.75	34	874K
Logistic Regression	2.07	2	13.7M

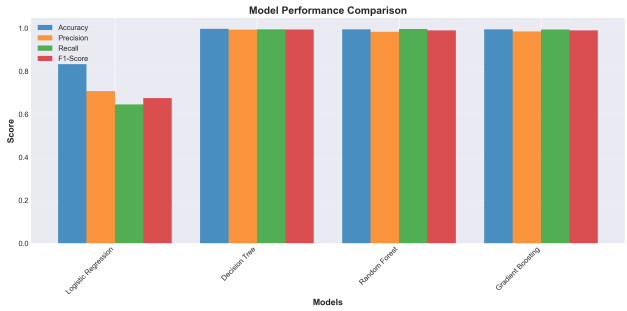

Fig. 12. Comparative computational performance (training and testing times).

almost an ideal AUC of 99.95%, and it is also highly interpretable and efficient in real-time detection. These findings confirm that transparency and rule-based models can significantly enhance the performance and high-quality outcomes of IDS in relation to ransomware attack-resistance.

In addition, All Features also supported this conclusion as the highest accuracy (99.65%) was obtained using all 13 attributes with the least amount of runtime. This validates the move to maintain the entire feature set within the ultimate detection model.

VII. Discussion

The results of the experiments clearly illustrate that supervised learning models, particularly the Decision Tree model applied to the SMOTE balanced traffic data are very effective in the detection of malware associated with ransomware applications. As we discussed in the pervious sections, the model Decision Tree shows impressive performance metrics of 99.40% accuracy, 98.0% precision and 99.9% recall confirming the model's strong ability to identify complex behaviors associated with ransomware. With an equally impressive AUC R - ROC metric of 99.95%, the Decision Tree model can easily distinguish between benign normal traffic and malicious traffic. Although there are ensemble machine learning algorithms like Random Forests and Gradient Boosting which have similar accuracy, the Decision Tree algorithm has the unique advantage of speed of computation, thus making it suitable for cyber defense applications.

The analysis of feature importance reflected certain key features such as the type of protocol used, the presence of certain flags, and source/destination addresses as well as transaction-related features such as the USD value and clustering information of the traffic data. These were thus the important features in malware detection associated with ransomware behavior. This result is in accordance to much earlier experimental work that has linked unusual protocol-type activity or behavior shown by network traffic and the

suspicious transactions associated with ransomware and the theft of data. The application of SMOTE for effectively balanced datasets shows promise for the considerable decrease of false negatives of around 81% and the detection of even rare types of ransomware without loss of accuracy. This is an important tradeoff in intrusion detection applications where the loss of an attack results in much greater consequences than the triggering of a few additional alerts. In the comparison of feature selection methods performed, it was found that all 13 features produced the best results by achieving 99.67% accuracy with an insignificant increase in runtime. Therefore, the decision was made to keep all features within the model, as an appropriate, well defined and complete feature collection aids in sustaining accuracy without additional expense.

From a pragmatic view point, the Decision Tree model was found to produce the best combination of accuracy, transparence and run-time. It was trained very quickly and thus completed training in 0.37 seconds and produced predictions of approximately 4 ms for each prediction, with the ability to process more than 7.5 million samples a second. Such efficiency would easily allow for integration of the model within practical situations such as Security Information and Event Management (SIEM) OR NIDS where decisions need to be made rapidly and transparency is the requirement. It can be seen that our study has found that lightweight, interpretable models lend themselves to improvement the data they are trying to process and, through intelligent use of feature engineering can rival the larger and more complex algorithms and be much more serviceable to real time processing.

There are numerous possible avenues that our work could take in the future. Enlarging our data to include more families of ransomware and real network traffic data would help the model be more discriminating and able to detect new attacks. Future work could assess the logic of using interpretable models in conjunction with other ensemble learning methods or deep learning methods, to form resilience to zero day attacks or adversarial attacks. Other avenues of study could include features that use explainability tools such as SHAP or LIME which would help Security Analysts determine the reasoning behind the model decisions taken.

Finally, deployment of such an integrated detection mechanism could be ideal in applications where the distributed nature of processing or streaming nature of the data, as would be the case using Apache Flink or Google Cloud Dataflow. This enables testing of the algorithms in real network conditions of high speed and large scale that would be the requirement in real industrial circumstances. Integrating within the detection algorithms other data sources, for example, telemetry data from endpoints or block chain based threat indicators, could assist greatly in providing detection within a Zero Trust Security Framework. It is hoped that our work will continue along these routes and move this work along from a successful experimental prototype to a practical, adaptive and transparent model of intrusion detection that is capable of combating the ever evolving and changing face of the ransomware threat we face.

We deliberately scoped this study to classical supervised classifiers in order to isolate the effect of SMOTE balancing and feature selection under identical, fully reproducible conditions. Recent deep learning approaches reported on the same UGRansome dataset including the BERT-based model of Rahman et al. [26] (99.21%), the TEVEInet transformer VAE hybrid of Fan et al. [31] ($F_1 = 99.26\%$), and the SAE–LSTM architecture of Nkongolo et al [35] (98.5%) report accuracy that is comparable to, or marginally lower than, our Decision Tree result (99.40%), but at substantially higher computational cost and reduced interpretability. Our framework reaches equivalent accuracy with an inference latency of 4 ms per batch and a throughput of 7.5 M samples/s, suggesting that for the UGRansome dataset the marginal accuracy gain offered by neural architectures does not justify their deployment cost in real-time IDS pipelines. A direct empirical comparison with CNN–LSTM and transformer baselines under our exact preprocessing pipeline is identified as future work in the Limitations section.

A. Limitations

While the claimed success is high, there are limitations to this study that should be recognized, as they guide the interpretation of the current results, and future work in this field.

1) Single dataset dependency: We evaluate only using the UGRansome dataset. Although UGRansome is a prominent and publicly available malware traffic dataset that features realistic event trace of notorious ransomware its traffic characteristics may not be typical for real enterprise environments [29]. Variations in network topology, traffic volume, protocol distribution and user behavior for different organizations may impact model generalization. Validation on other datasets such as CICAndMal2017 or on live traffic captures would enhance the external validity of the proposed framework.

2) SMOTE-Induced overfitting risk: The use of SMOTE to tackle the class imbalance issue creates synthetic samples by interpolating between the existing minority class samples. This can lead to samples that are extremely similar to the training data, which can lead to an overestimation of performance due to overfitting to synthetic patterns rather than learning generalizable ransomware signatures. Although the use of SMOTE was restricted to the training partition (it was never used in the test set), and the high AUC-ROC scores across all the tree-based models indicate high discrimination power, further work is required in order to consider alternative balancing techniques (such as ADASYN, borderline-SMOTE, or cost-sensitive learning) in order to reduce this risk.

3) Binary classification scope: The existing framework conducts binary classification (normal vs. ransomware) and fails to differentiate ransomware families and detect multi-stage attack progression. In practice, the ability to group specific ransomware families (e.g. WannaCry, Locky, CryptoLocker) would make it possible to take more targeted incident response actions. Future extensions should explore multi-class classification with labeling at the family level, and multi-stage detection of the ransomware lifecycle, from initial infection to encryption to command-and-control communication.

4) Encrypted traffic challenges: Modern ransomware often uses the SSL (HTTPS, TLS, etc.) for hiding the command and control communications as well as for concealing data exfiltration activities. The UGRansome dataset is not specific to encrypted traffic scenarios and the features used in this

study (protocol flags, transaction values, network flow characteristics) may have had reduced discriminative power when payloads are encrypted. Addressing this limitation, encrypted traffic analysis techniques like TLS fingerprinting, JA3/JA3S hashing or metadata based analysis working on connection level attributes instead of payload content need to be incorporated.

5) Absence of deep learning and hybrid approaches: This study is only limited to classical supervised learning algorithms. While the results show that tree-based models show excellent results on the UGRansome dataset, recent developments in deep learning (e.g. CNN-RF hybrids, Transformer-based models, and LSTM-autoencoder architectures have shown promising results for the detection of more sophisticated and evasive ransomware variants. Including such techniques in future comparative research may help understand if further accuracy improvements are possible, especially with polymorphic or fileless ransomware edge cases [26], [44].

6) Limited real-time deployment validation: While the computational efficiency metrics suggest the Decision Tree model can reach throughputs in excess of 7.5 million samples per second, these results were derived in an offline environment. In a real-world network, other factors come into play such as the time taken to capture packets, extract features, the memory footprint of the application and other simultaneous processing tasks. A comprehensive assessment of system throughput, latency, and CPU/memory usage under realistic traffic loads remains necessary to validate practical deployability.

7) Default hyperparameters: All classifiers were trained using default scikit-learn hyperparameters with no optimization using grid search or random search, or Bayesian methods. While the good baseline results indicate that the feature representations are very discriminative, further improvements may be achieved through systematic hyperparameter tuning, especially for Gradient Boosting (where learning rate, tree depth and number of estimators interact non-linearly) and Logistic Regression (where the strength of regularization affects the decision boundary).

8) Dataset bias and result interpretation: The very high accuracy scores (exceeding 99% for the tree-based models) should be interpreted with care. Such performance may be partly a reflection of aspects of the UGRansome dataset itself, such as there being clear separation between class distributions or that there is availability of strongly predictive features (perhaps ones that may not be available in all real-world network environments).

VIII. Conclusion and Future Work

This study proposed a supervised ML technique to detect ransomware in network traffic using the UGRansome dataset. The complete workflow from data preprocessing and normalization to model training, balancing and evaluation was carefully undertaken to ensure its reproducibility and practicability for real-world cyber-security applications.

The proposed framework combines good preprocessing and balancing methodologies that help in ensuring the reliability and consistency of model performance. In this study we applied the four supervised algorithms: DT, (RF), (GB) and (LR) on the original imbalanced dataset and on its SMOTE balanced version. There were clearly observable differences between the balanced and unbalanced versions with respect to recall (improvement of approximately 81%) and also the benefits which followed from a large reduction in false negatives at the point of detecting ransomware attacks.

Of the models tested, the decision tree was the best overall performer with a good compromise on each of accuracy,and speed. Achieving an accuracy of 99.4% with a recall of 99.90% AUC-ROC area under the receiver operating characteristics curve of 99.95%, indicated a very positive capability of separation of normal and ransomware traffic. The model performed its training in less than 1 second, made for predictions largely in negligible time and hence is suitable for real time use in security systems. These results seem to indicate that simple and explainable models can perform to good effect and quality as the more complicated alternatives, whilst remaining of low weight suitable for speedy execution.

A comparison between Genetic Algorithm (GA) and Correlation-Based Feature Selection (CFS) showed that using all 13 features yielded the best accuracy (99.67%), precision (99.33%), recall (99.46%), and F1-score (99.40%) with the lowest computational overhead compared to the other techniques.

This result would indicate that in situations, where datasets are smaller, cleaner and compact in their construction that by employing provision of the whole range of features a superior result can be attained without supplementary feature reduction being necessary.

It can be seen that this study offers a framework for a system and approach to ransomware detection in real network environments which is highly efficient and user friendly, accurate and interpretable. With high performance, low latencies and good transparency this has strong claims to be introduced into IDS (Intrusion Detection System) and SIEMs (Security Intelligence Enabling Modules).

Future work will be directed into extending the dataset to include more ransomware families, hybrid and explainable AI models together with deployment of the technology in either streaming or distributed environments to increase scalability, flexibility and confidence of automated ransomware defense.

Funding

This work was funded by the King Faisal University, Saudi Arabia [Project No. GRANT KFU262639].

Acknowledgment

This work was supported through the Annual Funding track by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project No. GRANT KFU262639].

Conflicts of Interest

All authors declare no conflict of interest.

Author's Contributions

All authors equally contributed.

REFERENCES

[1] Y.-L. Wan, J.-C. Chang, R.-J. Chen, and S.-J. Wang, "Feature-selection-based ransomware detection with machine learning of data analysis," in *2018 International Conference on Computer Communication and Multimedia (CCOMS)*, Apr. 2018.

[2] K. Schmaltz, S. C. Thompson, D. F. da C. M. Mendes, and J. Carvalho, "Robust defense mechanisms against adversarial ransomware attacks: Implementing a universal network-level detection filter," Sep. 2024.

[3] S. Ahmad, Z. Zulkifli, N. H. Nasarudin, M. Imran, and M. Ariff, "A recent systematic review of ransomware attack detection in machine learning techniques," in *2023 International Conference on Artificial Intelligence and Data Science (AIDAS)*, Sep. 2023.

[4] A. Abcam, L. Cranshaw, C. Worthington, and J. Vandenberghe, "Behavioral pattern analysis for real-time detection of ransomware attacks," Jan. 2025.

[5] P. Novak, P. Kaura, V. Oujezsky, and T. Horvath, "Ransomware file detection using hashes and machine learning," in *2023 15th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Jan. 2023.

[6] K. Higuchi and R. Kobayashi, "Rofbsa: Real-time backup system decoupled from ml-based ransomware detection," arXiv preprint, Apr. 2025.

[7] L. Chen, C.-Y. Yang, A. Paul, and R. Sahita, "Towards resilient machine learning for ransomware detection," *Proceedings of KDD Workshop on Security and Privacy*, pp. 1–10, 2019.

[8] C. Khoza, G. M. Komba, and S. P. Maswikaneng, "An enhanced ransomware defense strategy through behavior-based detection using machine learning algorithms," in *2024 IEEE 8th Conference on Energy Internet and Energy System Integration*, Jan. 2024.

[9] P. Veerasingam, S. A. Razak, A. F. A. Abidin, M. A. Mohamed, and S. D. M. Satar, "Intrusion detection and prevention in smes using suricata," *Malaysian Journal of Computing and Applied Mathematics*, vol. 6, no. 1, pp. 21–30, 2023.

[10] N. Rani and S. V. Dhavale, "Leveraging machine learning for ransomware detection," arXiv preprint, Jun. 2022.

[11] V. Kumar and O. P. Sangwan, "Signature based intrusion detection system using snort," *International Journal of Computer Applications in Information Technology*, vol. 1, no. 3, pp. 35–41, 2012.

[12] S. Chakrabarti, M. Chakraborty, and I. Mukhopadhyay, "Study of snort-based ids," in *Proceedings of International Conference & Workshop on Emerging Trends in Technology*, 2010, pp. 43–47.

[13] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[14] M. Al-Zewairi *et al.*, "Multi-stage enhanced zero trust intrusion detection system for unknown attack detection in internet of things and traditional networks," *ACM Transactions on Privacy and Security*, 2025.

[15] A. O. Almashhadani, M. Kaiiali, S. Sezer, and P. Okane, "A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware," *IEEE Access*, vol. 7, pp. 47 053–47 067, 2019.

[16] S. Sivalmi, Jayashree, and H. Noor, "The dual role of ai in cyber security: Enhancing ransomware and defending against attacks," *Journal of Computer and Communication Systems*, vol. 1, no. 3, pp. 20–30, 2025, discussion on ransomware evolution, impact including economic/social, and attack strategies, alongside AI-powered intrusion detection systems and attack countermeasures.

[17] M. Benmalek, "Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges," *Internet of Things*, 2024, comprehensive overview of ransomware threats, taxonomies, and implications on cyber-physical systems, highlighting attack vectors, organizational impacts, and mitigation challenges.

[18] W. Z. A. Zakaria, M. F. Abdollah, O. Mohd, S. M. W. Mohamed, M. Yassin, and A. Ariffin, "Rentaka: A novel machine learning framework for crypto-ransomware pre-encryption detection," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 378–387, 2022, details ransomware definitions, lifecycle, and ML-based early detection with emphasis on cryptographic ransomware behavior and attack vectors.

[19] M. Benmalek, "Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges," *Internet of Things and Cyber-Physical Systems*, Jan. 2024.

[20] Syed Shameem, "Estimating malware impact on network traffic analysis by using wireshark," vol. 20, no. 7, pp. 965–972. [Online]. Available: https://journal.esrgroups.org/jes/article/view/3477

[21] B. Dodiya and U. K. Singh, "Malicious traffic analysis using wireshark by collection of indicators of compromise," vol. 183, no. 53, pp. 1–6. [Online]. Available: http://www.ijcaonline.org/archives/volume183/number53/dodiya-2022-ijca-921876.pdf

[22] G. Pimenta Rodrigues, R. De Oliveira Albuquerque, F. Gomes De Deus, R. De Sousa Jr., G. De Oliveira Júnior, L. García Villalba, and T.-H. Kim, "Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection," vol. 7, no. 10, p. 1082. [Online]. Available: https://www.mdpi.com/2076-3417/7/10/1082

[23] I. Sembiring, S. , A. Iriani, J. V. B. Ginting, and J. A. Ginting, "A novel approach to network forensic analysis: Combining packet capture data and social network analysis," vol. 14, no. 3. [Online]. Available: http://thesai.org

[24] M. Williams, R. Morales, K. Johnson, G. Martinez, and J. Bennett, "Entropy-based network traffic analysis for efficient ransomware detection." [Online]. Available: https://doi.org/10.36227/techrxiv.172840776.66718131/v1

[25] I. Chaudhary and S. Adhikari, "Ransomware detection using machine learning: A comparative study," *Researcher CAB*, vol. 10, no. 1, pp. 96–114, 2024.

[26] M. Abdur Rahman *et al.*, "Machine learning models for ransomware detection using ugransom1819," *Computers*, 2025.

[27] P. Azugo, H. Venter, and M. W. Nkongolo, "Ransomware detection and classification using random forest: A case study with the ugransome2024 dataset," *arXiv preprint arXiv:2404.12855*, 2024.

[28] S. J. Nhlapo and M. N. W. Nkongolo, "Zero-day attack and ransomware detection," 2024. [Online]. Available: https://arxiv.org/abs/2408.05244

[29] M. Nkongolo, J. P. van Deventer, and S. M. Kasongo, "Ugransome1819: A novel dataset for anomaly detection and zero-day threats," *Information*, vol. 12, no. 10, p. 405, 2021.

[30] A. Alraizza and A. Algarni, "Ransomware detection using machine learning: A survey," *Big Data and Cognitive Computing*, vol. 7, no. 3, p. 143, 2023.

[31] H. Fan, W. Le, Z. Jia, H. Zhao, C. He, H. Jiang, Z. Hu, X. Lv, J. Yuan, and X. Huang, "Teveinet: A deep feature extraction network for anomaly detection," Preprint, SSRN, 2025, https://ssrn.com/abstract=5325605.

[32] S. R. Zahra, "Ransomware detection using ensemble machine learning models with ugransom1819," *Journal of Cybersecurity Research*, 2022.

[33] A. A. Alhashmi, A. A. Darem, A. B. Alshammari, L. A. Darem, H. K. Sheatah, and R. Effghi, "Ransomware early detection techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14 497–14 503, 2024.

[34] P. Yan, T. T. Khoei, R. S. Hyder, and R. S. Hyder, "A dual-stage ensemble approach to detect and classify ransomware attacks," in *2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, pp. 781–786. [Online]. Available: https://ieeexplore.ieee.org/document/10754695/

[35] M. N. W. Nkongolo, "Ensemble learning and genetic algorithm for the detection of novel network threat anomaly using the ugransome dataset," Ph.D. thesis, University of Pretoria, Faculty of Engineering, Built Environment and Information Technology, 2023, supervisor: Dr. Jacobus Philippus van Deventer; Co-supervisor: Dr. Sydney Mambwe Kasongo.

[36] M. N. W. N, "Zero-day vulnerability prevention with recursive feature elimination and ensemble learning," Cryptology ePrint Archive, Paper 2023/1843, 2023. [Online]. Available: https://eprint.iacr.org/2023/1843

[37] S. K. D.R.V.A, R. V, K. Rayudu, V. A, T. B, and S. K. R, "Enhancing ransomware detection in cybersecurity: A comprehensive ensemble approach," *Journal of Electrical Systems*, vol. 20-10s, pp. 5222–5232, 2024.

[38] B. Torky, "Ensemble methods for the anomaly detection in enterprise systems," Master's thesis, Rochester Institute of Technology, Department of Graduate Programs Research, RIT Dubai, 2023, capstone, Master of Science in Professional Studies Data Analytics.

[39] A. Mohamed, M. Mohamed, and M. Ata, "Zero-day exploits detection with adaptive wavepca-autoencoder (awpa): Adaptive hybrid exploit detection," 2025, submitted Manuscript.

[40] E. N. Mutombo and M. W. Nkongolo, "Blockchain security for ransomware detection using machine learning and the ugransome dataset," *arXiv preprint arXiv:2407.16862*, 2024. [Online]. Available: https://arxiv.org/abs/2407.16862

[41] A. A. Ahmed, A. Shaahid, F. Alnasser, S. Alfaddagh, S. Binagag, and D. Alqahtani, "Android ransomware detection using supervised machine learning techniques based on traffic analysis," *Sensors*, vol. 24, no. 189, 2024.

[42] A. Srivastava, N. Kumar, A. Handa, and S. K. Shukla, "Ransomware detection based on network behavior using machine learning and hidden markov model with gaussian emission," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2023.

[43] C. Manzano, C. Meneses, and P. Leger, "An empirical comparison of supervised algorithms for ransomware identification on network traffic," in *2020 IEEE ANDESCON*, 2020.

[44] L. Woo, S. Tan, T. Tan, and G. Lao, "A hybrid approach to ransomware detection using convolutional neural networks and random forests on network traffic patterns," in *2024 International Conference on Cyber Security (ICCS)*, 2024.

[45] I. Batalov, P. Smirnov, and A. Lebedev, "Ransomware detection via network traffic analysis using isolation forest and lstm neural networks," *Journal of Cybersecurity and Digital Forensics*, vol. 12, no. 3, pp. 145–160, 2024.

[46] S. Mehrban and E. Geransayeh, "Ransomware detection using machine learning algorithms and network traffic features," *Computer Communications*, vol. 213, pp. 190–204, 2024.

[47] J. Brinkley, S. Alvarez, and W. Zhang, "Machine learning-based intrusion detection for zero-day ransomware threats," *IEEE Access*, vol. 12, pp. 73 210–73 224, 2024.

[48] Y. Cen, R. Zhang, M. Liu, and Y. Hu, "Zero-ran sniff: Zero-shot learning for early detection of zero-day ransomware," *Computers & Security*, vol. 145, pp. 103–119, 2024.

[49] E. Fevid, C. Walsh, and L. Russo, "Zero-day ransomware detection via assembly language bytecode analysis and random forest classification," *TechRxiv Preprint*, September 2024, available at https://doi.org/10.36227/techrxiv.17256576.v1.