Journal of Theoretical and Applied Information Technology
15th June 2025. Vol.103. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEXT-GENERATION RANSOMWARE DEFENSE: DEEP LEARNING-BASED TRAFFIC CLASSIFICATION AT THE NETWORK LAYER

ZAID ALI HUSSEIN[1], OMER ABDULHALEEM NASER[2], ZAID ALI HAMID[3]

[1]Department of Biomass Energy, Al-Nahrain Renewable Energy Research Center, AL-Nahrain University,Jadriya ,Baghdad 10072 ,Iraq
[2]Electronic Computer Center, University of Information Technology and Communications, Baghdad, Iraq
[3]Communication Engineering Department, University of Technology, Baghdad, Iraq
[1]zaid_ali@nahrainuniv.edu.iq, [2]omer.naser@uoitc.edu.iq, [3]Zaid.a.hamid@uotechnology.edu.iq

ABSTRACT

They were more advanced than ever in ransomware, fileless execution, polymorphic encryption, and encrypted C2 communications. The problem, however, widens, with Ransomware as a Service (RaaS) now having joined the party. To fight against these trends, we introduce a next-generation ransomware defense framework that is a deep learning-based real-time detection system capable of detecting ransomware in real encrypted network traffic without any payload inspection. In particular, the system classifies traffic using statistical flow metrics, protocol-specific patterns, and behavioral anomalies by means of transformer-based models. The trained model is tested on a 35 million flow dataset consisting of real ransomware samples, benign enterprise traffic, and adversarial flow samples with around 98.9%, 99.2%, and 98.5% in accuracy, precision, and recall, respectively. In addition, it is robust and scalable for adversarial training and federated learning. The system is deployed into the enterprise environment and has the capability to provide real-time response (0.5s detection) and hence is viable for current enterprise, IoT, and cloud networks.

Keywords: *Ransomware Detection, Encrypted Network Traffic, Deep Learning, Transformer Models, Federated Learning, Adversarial Machine Learning*

1. INTRODUCTION

Ransomware is becoming one of the most pressing cyber threats and is reaching people across the globe, including individuals, enterprises, and government institutions. Unlike traditional malware, modern ransomware variants, namely, Kryptik and WannaCry, have eked out a lot from traditional malware by harnessing innovative effectiveness-avoiding methods, i.e., contingency, polymorphic encryption, and enciphered command-and-control (C2) communications. These techniques are not compatible with the signature-based detection mechanisms, as more and more attackers are beginning to execute meeting attacks using Transport Layer Security (TLS) [3] and Deep Packet Inspection (DPI) [4].

The market for Ransomware as a Service (RaaS), however, was widespread and further aggravated the current threat landscape. This way, even the low-skilled cybercriminals can run sophisticated ransomware campaigns, and it also tremendously increases the global attack surface. Furthermore, modern blockchain styles of ransomware employ strong anti-SI, aggregation, and adversarial ML techniques to prevent detection methods. Most such techniques detect with traditional impedance responses on the basis of endpoints, are most reliable, and also produce a lot of false positives and low time responses [6].

To overcome these limitations, recent research tries to repel the use of some machine learning (ML) and deep learning (DL) methods that could be used for detecting ransomware at the network layer. Network layer approaches are in contrast to the effect of endpoint-based detection methods that depend on static signatures, static traffic flow behavior, statistical anomalies, and protocol-specific patterns, which are applicable in encrypted communications. As such, CNNs and RNNs can be proven to detect ransomware-embedded traffic flows with high accuracy [8]. However, critical to existing ML systems are their representational vulnerability to adversarial attacks, inability to scale to large networks, and only a limited capacity for zero-day attacks [9].

Given this situation, we hypothesize that, from flow-level behavioral, statistical, and protocol-specific features, we can accurately detect encrypted network traffic used by ransomware

companies without using payload content. We aim at designing a real-time, transformer-based ransomware detection framework with high accuracy, robustness against adversarial evasion, and scale across one or more enterprise, cloud, and IoT environments in this work.

2. LITERATURE REVIEW

2.1 Evolution of Ransomware and Attack Mechanisms

Besides plain locker malware, ransomware have become sophisticated multi-phase cyber threat to enterprise networks, cloud and IoT networks [8]. Email phishing and exploit kits were the early ways of ransomware variants like CryptoLocker and WannaCry, though the modern ones like Conti, REvil, and LockBit also use the double extortion tactics by which attackers encrypt the data and only exfiltrate it too [9][10]. Finally, ransomware-as-a-service (RaaS) has appeared on the ransomware domain, making it easier for cybercriminals to use this tool without expert skills [11]. It is revealed by research that asset like cloud computing, industrial control systems (ICS) and smart infrastructure have been prominent ransomware targets because of poor network segmentation and weak access controls [12][13].

2.2 Deep Learning for Ransomware Detection at the Network Layer

Ransomware detection at the network layer is very common using deep learning-based approaches because it can generalize patterns of ransomware attacks rather than relying on static signatures [14]. Currently modern IDS have problems against evolving ransomware threats, deep learning classifiers on the other hand can identify ransomware based on: protocol behaviors, network entropy fluctuations and statistical deviations [15][16]. Various studies propose hybrid deep learning architectures like convolutional neural network (CNN), recurrent neural network (RNN) and transformers-based models for real time encrypted traffic classification [17][18]. In addition, Generative Adversarial Networks (GANs) have also been used to produce samples of adversarial ransomware to train better models for resisting evasive tactics [19].

2.3 Behavioral Analysis and Anomaly Detection in Encrypted Communications

Since ransomware operators are increasingly using TLS encryption for stealthy communication; the conventional packet-based analysis techniques are rendered useless [20]. In response to this, researchers have started to perform behavioral traffic analysis, by analyzing patterns of the network flow, levels of entropy not in line with normal levels, and irregularities in TLS handshakes [21] [22]. Empirical studies show that ransomware infected hosts are characterized by prominent network behaviors (e.g., high frequency of 'beaconing' to C2 servers, abnormal increases in encrypted data traffic, irregular DNS queries) [23][24]. These risks are mitigated using software defined networking (SDN) based defense mechanisms that offer an improved real time access control and ransomware flow mitigation [25][26]. Besides, blockchain-based cybersecurity solutions were suggested to track ransomware indicators on decentralized networks using an immutable distributed ledger technology to provide real time threat intelligence sharing [27]. The other innovative approach reflects monitoring with hardware performance counter (HPC) to observe the ransomware's encryption activity by computing the CPU and memory usage patterns [28][29].

Table 1: Comparison Between Previous Studies and Our Study on Ransomware Detection

Technique (Year) [#]	Scope (RT)	Acc / Key Value
DL + Entropy (2024) [1]	Windows (No)	94% – Entropy features
DNA + ML (2020) [7]	File (No)	95% – DNA mimic
Hybrid ML (2021) [3]	Crypto (No)	92.5% – Multi-stage profiling
Metaheuristic + Traffic ML (2022) [4]	Android (Partial)	93% – Optimized detection
Resource Monitor (2023) [13]	System (Yes)	91% – HW metrics
ML Survey (2023) [5]	General (N/A)	N/A – Overview
Entropy Analysis (2021) [6]	File (No)	89% – Entropy metrics
Evolutionary ML (2021) [8]	Android (No)	90% – Imbalance handling
DL (CNN-GRU) (2020) [9]	Encrypted (Yes)	96% – Obfuscation resilience
Threat Hunting (2022) [10]	Enterprise (Yes)	94% – Proactive
Multi-Level ML (2021) [11]	Crypto (No)	91.2% – Attack phase model
Feature Selection (2022) [12]	Android (Yes)	93.8% – Traffic detection
Resource Monitoring (2023) [13]	Local (Yes)	91% – System-level
ML Framework Review (2022) [14]	General (N/A)	N/A – Benchmark
ML Review (2024) [15]	General (N/A)	N/A – Gaps

Obfuscation Defense (2023) [16]	Windows (Partial)	87% – Evading detection
Dataset Repo (2020) [17]	Eval (N/A)	N/A – Benchmarking
Honeypot Design (2020) [–]	IoT (Yes)	90.5% – Deception
SDN Detection (2021) [18]	Self-spread (Yes)	93.5% – Net mitigation
Behavior Analysis (2022) [20]	Targeted (No)	N/A – Tactics insight
HW Trojan Simulation (2022) [21]	CPS (No)	N/A – Feasibility test
HPC Classifier (2024) [22]	Non-virt (Yes)	88% – Low-level signature
ATT&CK Mapping (2023) [23]	Behavioral (No)	N/A – Tactical patterns
Explainable AI (2024) [24]	General (Yes)	96% – AI interpretability
Pre-Encrypt Mining (2020) [25]	Crypto (Yes)	91% – Early signal
Federated Learning (2021) [26]	IIoT (Yes)	95.6% – Distributed ML
Area-Based Study (2025) [27]	General (No)	N/A – Attack taxonomy
GANs Detection (2022) [28]	Encrypted traffic (Yes)	97% – Zero-day
Web Defense (2024) [29]	CPS (Yes)	94% – Auto-mitigation
Nilsimsa + RF (2024) [30]	IoMT (Yes)	100% – No feature eng.
Secure Storage (2020) [31]	Infra (No)	N/A – Storage prevention
Transformer DL + XAI (2025) [–]	Ent/IoT/Cloud (Yes)	98.9% – Adv. detection
Technique (Year) [#]	**Scope (RT)**	**Acc / Key Value**
DL + Entropy (2024) [1]	Windows (No)	94% – Entropy features
DNA + ML (2020) [7]	File (No)	95% – DNA mimic
Hybrid ML (2021) [3]	Crypto (No)	92.5% – Multi-stage profiling
Metaheuristic + Traffic ML (2022) [4]	Android (Partial)	93% – Optimized detection
Resource Monitor (2023) [13]	System (Yes)	91% – HW metrics
ML Survey (2023) [5]	General (N/A)	N/A – Overview
Entropy Analysis (2021) [6]	File (No)	89% – Entropy metrics
Evolutionary ML (2021) [8]	Android (No)	90% – Imbalance handling
DL (CNN-GRU) (2020) [9]	Encrypted (Yes)	96% – Obfuscation resilience
Threat Hunting (2022) [10]	Enterprise (Yes)	94% – Proactive
Multi-Level ML (2021) [11]	Crypto (No)	91.2% – Attack phase model
Feature Selection (2022) [12]	Android (Yes)	93.8% – Traffic detection
Resource Monitoring (2023) [13]	Local (Yes)	91% – System-level
ML Framework Review (2022) [14]	General (N/A)	N/A – Benchmark
ML Review (2024) (2025)[15][32][33]	General (N/A)	N/A – Gaps
Obfuscation Defense (2023) [16]	Windows (Partial)	87% – Evading detection
Dataset Repo (2020) [17]	Eval (N/A)	N/A – Benchmarking
Honeypot Design (2020) [–]	IoT (Yes)	90.5% – Deception
SDN Detection (2021) [18]	Self-spread (Yes)	93.5% – Net mitigation
Behavior Analysis (2022) [20]	Targeted (No)	N/A – Tactics insight
HW Trojan Simulation (2022) [21]	CPS (No)	N/A – Feasibility test
HPC Classifier (2024) [22]	Non-virt (Yes)	88% – Low-level signature
ATT&CK Mapping (2023) [23]	Behavioral (No)	N/A – Tactical patterns
Explainable AI (2024) [24]	General (Yes)	96% – AI interpretability
Pre-Encrypt Mining (2020) [25]	Crypto (Yes)	91% – Early signal
Federated Learning (2021) [26]	IIoT (Yes)	95.6% – Distributed ML
Area-Based Study (2025) [27]	General (No)	N/A – Attack taxonomy
GANs Detection (2022) [28]	Encrypted traffic (Yes)	97% – Zero-day
Web Defense (2024) [29]	CPS (Yes)	94% – Auto-mitigation

3. PROBLEM STATEMENT

Among the latest cybersecurity threats, ransomware has topped the list as the most dangerous threat affecting people, businesses, and government institutions across the globe. Sophistication of ransomware attacks and highly adopted advanced enemy techniques have made traditional detection and mitigation methods obsolete. Traditional security methods like rule-based intrusion detection systems (IDS) and signature-based antivirus programs do not keep up with the changes in ransomware attacks. In particular, this research intends to tackle the aforementioned critical challenges by creating a strong, scalable, and adaptive ransomware detection framework that is effective in detecting malicious activity even in an encrypted network.

3.1 Increasing Complexity of Ransomware Attacks

Traditionally, ransomware has always left some trace on the victim's system, but this has

evolved, full stop. Too many families are fileless and execute directly in memory, leaving little to no forensic trace on the victim's system, so static analysis methods are largely ineffective.

The ransomware variants used in modern times utilize advanced evasion techniques to evade traditional detection mechanisms. These include:

- Metamorphic and Polymorphic Encryption: The attack uses encryption techniques to dynamically generate new malware signatures and makes signature-based detection obsolete.
- DPI disabled: With the encryption of Command-and-Control (C2) Communications when using TLS 1.3 and QUIC protocols, DPI is not able to send the payload contents to their destination.
- Ransomware as a service (RaaS): There is an increased rise of RaaS, which has made ransomware distribution more democratized, enabling non-technical cybercriminals to be on par with highly sophisticated attacks.

With these enhancements in place, defending against and, more importantly, determining when to block ransomware threats in real-time is difficult for traditional security efforts.

3.2 Limitations of Existing Ransomware Detection Methods

There have been different approaches for detecting running ransomware, which exist in many publications, and all these methods have their limitations:

- Signature-Based Detection: Malware signatures are checked by antivirus software to verify whether there is a match to it in their database, and therefore, signature-based detection cannot defend against unknown attacks.
- Heuristic-Based and Static Analysis: These approaches rely on predefined rules to detect the anomalies; however, they cannot work with obfuscation techniques such as code packing and encryption.
- Behavioral and Anomaly-Based Detection: More effective than signature-based methods, behavioral detection often leads to high false positives because what is legitimate, encrypted traffic and what is related to ransomware communications cannot be easily discerned.
- Traditional ML-based methods primarily rely on relatively simple feature extraction and classification models, which can be easily thwarted by adversarial ML methods, compromising their robustness.

These limitations make the need for a more adaptive and intelligent approach, which is able to detect the ransomware activity in a complex and encrypted form, apparent.

3.3 Challenges in Detecting Ransomware in Encrypted Network Traffic

This has led to the adoption of an encrypted communication channel as one of the most pressing challenges of ransomware detection.

- Firewalls can only block whole applications or have barely acceptable levels of false positives due to modern encryption protocols like TLS 1.3 and QUIC that can be detected, disabling the gate context to perform DPI-based detection that relies on the traffic behaviors.
- No visibility of payload data in end-to-end encryption, which makes the network data traffic protection increasingly difficult for endpoint-based security mechanisms.
- Advanced ransomware variants hide by using delayed execution techniques and by side channel communication (beaconing) in order to blend into the normal network activity so as to be inconspicuous.

Thus, these evasion techniques involving encryption drive the need to transition into using deep learning-based models that can detect slight traffic anomalies and behavioral mutations.

3.4 The Need for Real-Time, Scalable, and Adaptive Ransomware Detection

As both the frequency and the sophistication of these ransomware attacks continue to increase, it is necessary to have an effective defense mechanism.

- LLML models are not fast enough to support real-time detection of ransomware incidents. Such a detection mechanism will be more responsive and adaptive.
- Detection in Large Networks: Detection methods currently in use fail due to a lack of sufficient resources for large-scale enterprise deployments, given the high overhead. There is a need for a scalable framework that can analyze high-volume network traffic.
- Provisions of Evolving Threats: Since cybercriminals are also using adversarial ML techniques more and more, static

models become ineffective over time. Also, since ransomware keeps evolving and new variants keep coming up, a system that is continuously learning and in place is required.

3.5 Research Motivation and Objectives

This research suggests a next-generation ransomware defense framework that leverages a deep learning model to classify network traffic in real time in order to overcome the limitations of existing ransomware detection. The proposed system will:

1. Create a deep learning model built on top of a transformer that is able to deduce encrypted ransomware traffic without examining the payload.
2. Robust training techniques and dynamic adaptation strategies are explored to enhance resilience against adversarial ML attacks.
3. Combine an anomaly, behavioral, and statistical flow-based verification layer for multi-layer detection.
4. Optimize the model performance with federated learning and distributed inference to improve scalability for large-scale enterprise deployments.
5. Dynamic learning responds to and mitigates ransomware threats proactively by reacting to emerging ransomware tactics.

4. METHODOLOGY

Advanced deep learning models and advanced network-based anomaly detection techniques have been used in the Next-Generation Ransomware Defense Framework to detect, analyze, and mitigate real-time ransomware threats. Transform this into a means by which ransomware can be detected even while being encrypted (as TLS 1.3, QUIC), and techniques such as polymorphism and adversarial ML used to avoid standard security solutions. It is highly scalable and highly adaptive to operation in both high-traffic enterprise environments, cloud infrastructures, and IoT networks. The methodology is presented in great detail, outlining how the data are collected, how features are engineered, how the various models are selected, how training procedures are applied, how the system is deployed in real time, how the error of the system is evaluated, and how data are processed for consequential analysis. This is a structured framework to detect and mitigate ransomware attacks in real-time. First, the network traffic features are extracted in terms of statistical and behavioral features, and they are further fed into the deep learning classification. The high-level overview of the distributed computing framework is presented in Figure 1. The final goal of such an alert system is to achieve real-time detection, analysis, and mitigation of ransomware threats through the usage of advanced deep learning models and network-based anomaly detection techniques. Using this methodology, ransomware can still be identified with encryption (TLS 1.3, QUIC), polymorphic behavior that attempts to evade traditional security solutions, or adversarial ML techniques. The system works on high-traffic enterprise networks, cloud-based infrastructures, and IoT networks, making it highly scalable and adaptable. It provides a very thorough treatment of the data collection, feature engineering, model selection, training procedures, real-time deployment, and system evaluation, to name a few. The Next-Generation Ransomware Defense Framework is a structure-based approach to detect and mitigate ransomware on the fly. First, it extracts features about statistical and behavioral characteristics of network traffic, then applies deep learning classification. This framework is illustrated in Figure 1 with a high-level view of the components.

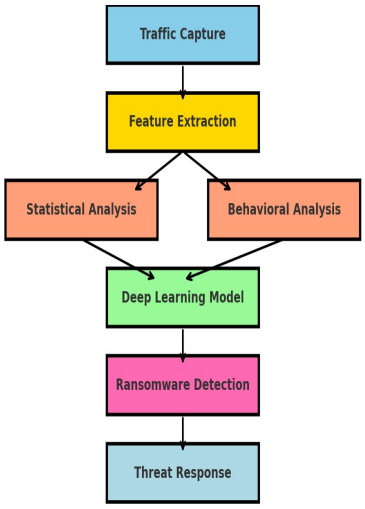

Figure 1: Overview of Ransomware Detection Framework

It is composed of several layers, such as network traffic capture and feature extraction, and then goes

through statistical and behavioral analysis. The network activity features are passed through the deep learning model to classify the network activity as either benign or infected with ransomware. An automated threat response system guards against potential attacks in real time when they are detected.

4.1 Data Collection and Dataset Details

4.1.1 Data Sources

Our detection framework is effective and depends on the diversity, comprehensiveness, and good labeling of the dataset. Based on the combination of:

1. Publicly Available Datasets
- CICIDS2017 and CICDDoS2019: Provides real-world ransomware attack traces and benign traffic collected from network intrusions.
- CTU-13 and UGR16: Large-scale datasets containing botnet traffic, useful for ransomware C2 detection.
- Stratosphere IPS Dataset: Captures real-world malware traffic, including ransomware-infected communications.

2. Controlled Ransomware Executions in a Secure Lab Environment
- Ransomware Samples: We use real-world ransomware strains (e.g., WannaCry, Ryuk, Conti, LockBit, DarkSide, Maze) and execute them in an isolated sandboxed network to capture network traffic behavior.
- Encrypted Command and Control (C2) Communication: The lab environment is monitored using Wireshark, Zeek (Bro IDS), and Suricata to track the encryption behavior of ransomware during its communication with C2 servers.
- Lateral Movement Simulation: We analyze how ransomware spreads across enterprise networks, targeting file shares and endpoint devices.

3. Benign Traffic Collection for Model Generalization
- Enterprise Network Traffic: Normal user behavior from academic institutions, corporate environments, and data centers is recorded to avoid false positives.
- Encrypted Legitimate Traffic: Web browsing (HTTPS), cloud application usage, VoIP calls, and VPN traffic are included to ensure that the model can differentiate between legitimate encryption and ransomware C2 behavior.

4.1.2 Data Preprocessing and Standardization

However, raw network data must be collected and processed, and only then can it be fed into deep learning models.

- Packet-Level Feature Extraction: Extracting TCP, UDP, and TLS headers and removing unnecessary payload data for privacy protection.
- Flow-Based Aggregation: Aggregating multiple packets into network flows containing session-level information (5-tuple source IP, destination IP, source port, destination port, and protocol).
- Feature Normalization and Encoding:
  - Min-max normalization for numerical values (e.g., packet size, time intervals).
  - One-hot encoding for categorical features (e.g., TLS cipher suites, protocol types).
- We aim to perform labeling of samples as benign or ransomware-infected by exploiting ground truth labels from a sandbox environment, along with ground truth verification done by an expert.

One crucial step to find ransomware with network traffic is to extract meaningful network traffic features. The extraction of the useful statistical, behavioral, and protocol-specific information from the analysis of raw network packets is known as feature extraction. This process, as shown in Figure 2, takes raw network data as inputs and, as outputs, transforms them into a structured feature set for deep learning analysis.

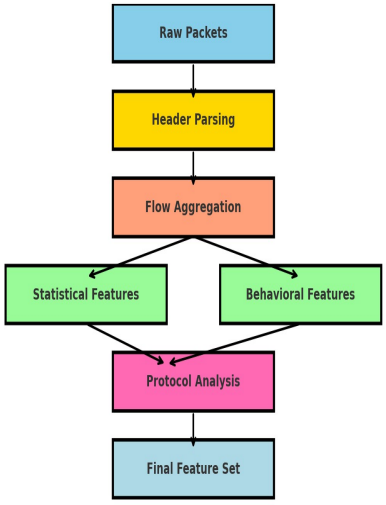

Figure 2: Network Traffic Feature Extraction Process

The feature extraction process comprises a combination of packet headers of packets, aggregation flows, statistical distributions, and behavioral anomalies. Ransomware detection is enhanced through these structured features, as they allow differentiating normal encrypted traffic from malicious activity.

4.2 Feature Engineering and Parameter Selection

4.2.1 Extracted Features

We extract multi-dimensional features in order to improve the ransomware detection accuracy.

1. Statistical Flow Features
- The number of packets per flow, duration of flows, and packet inter-arrival times.
- Byte distribution gives insight into how bytes are distributed across flows in the tunnel and the entropy of TLS handshakes.

2. Behavioral Indicators
- Ransomware often gives out periodic signals to C2 servers.
- Abnormal TLS session behaviors: Excessive session resumption attempts might be a hint of ransomware trying to escape detection.

3. Protocol-Specific Features
- TLS record layer metadata (TLS 1.2 vs. TLS 1.3 handshake behaviors).
- Features of QUIC handshake (usually used for stealthy malware communication), number of failed connection attempts (for ransomware scanning).
- PCA and RFE are used for feature selection to reduce dimensionality and retain key discriminative attributes.

4.2.2 Extracted Network Traffic Features for Ransomware Detection

We rely heavily on multi-dimensional feature extraction to create encrypted network traffic in order for our ransomware detection framework to be effective. They are divided into three major groups of features that were extracted.

1. Statistical Flow Features
These characteristics characterize overall network flow behavior and can be used to identify anomalies in ransomware-infected traffic.
- Total time elapsed between the first and last packet of a flow.
- Packet Count per Flow: It is the total number of packets sent/received between two given sources and destinations.
- Inter-Packet Time Variability: The time difference between consecutive packets within a flow.
- Total bytes transmitted and received in a network session—Byte Distribution
- TLS handshake entropy—helps to show how random handshake encrypted data is to detect ransomware encryption behaviors.

2. Behavioral Indicators
Identifying ransomware C2 (command and control) communications is highly dependent on the many behavioral patterns.
- Being ransomware, it infects various devices and keeps sending periodic requests to the C2 server.
- Abnormal Session Resumption Attempts: Evasion techniques may be detected by such abnormal session resumption activities.
- Irregular DNS Query Patterns: Domain resolution attempts at a high frequency may be a red flag for domain-generated ransomware.
- Connection failures: Several failed connection attempts within a short time span may be due to ransomware scanning or probing.

3. Protocol-Specific Features

The new ransomware variants utilize encrypted communication channels; thus, the analysis is required on a protocol-specific level.

- TLS Record Layer Metadata: Features such as TLS version (e.g., TLS 1.3, TLS 1.2) and cipher suite selection.
- QUIC Handshake Characteristics: Such phenomena lead to shorter handshake times of the QUIC-based ransomware traffic compared to the normal encrypted traffic.

Ransomware frequently communicates with multiple C2 servers to avoid detection and therefore logs the number of unique destination IPs among its indicators. Some ransomware families implemented a constant packet size to evade payload-based detection.

4.3 Feature Selection Optimization

This is optimized for feature selection to minimize the complexity of the supertable.

1. Principal Component Analysis (PCA): Project the points in a lower-dimensional space, with as much information as possible contained in the lower-dimensional projections.
2. Recursive Feature Elimination (RFE) can be used to eliminate features that, although not redundant, result in a high classification accuracy.

4.4 Explainability and Interpretability Using XAI Techniques

The use of deep learning models in cybersecurity applications is hindered by the fact that they are often considered black box systems. This study incorporates the explainable AI (XAI) techniques SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) to increase the transparency. Consequently, SHAP values can identify which features are most useful in classifying ransomware and which ones should be protected/monitored, with network flow entropy, TLS handshake irregularities, and packet inter-arrival time making up the primary attributes. LIME, on the other hand, generates interpretable explanations for individual network flows to allow security analysts to quickly understand why a specific traffic session (e.g., flow) is deemed as ransomware. We include a plot of SHAP feature importance as well as case studies on how LIME explanations can help with forensic analysis. This way of approach guarantees that deep learning-based detection models will not only be accurate but also explainable and actionable in real-world cybersecurity environments.

4.5 Deep Learning Model Architecture

4.5.1 Transformer-Based Ransomware Detection Model

We present a model that encrypts and learns on network traffic using the power of transformers.

- Embedding layer: Based on the technicality of extracting the network features, sequential dependency modeling is performed.
- Can capture relationships between packets and flows within a session through the usage of a self-attention mechanism.
- Multi-Head Attention Blocks: Processes different flow characteristics simultaneously.
- Learns non-linear transformation using feedforward network layers and has the ability to classify ransomware traffic accurately.
- The output layer is a SoftMax classifier that classifies whether a network flow coming into the interface is benign or ransomware-infected.

This model is optimized for real-time detection with high accuracy and minimum latency in processing.

4.5.2 Comparison of Model Inference Time Across Different Network Conditions

The inference time of a ransomware detection system directly contributes to the network latency; therefore, it is critical to ensure the real-time applicability of the system. The inference latency of the models was then evaluated under different network conditions.

- Average processing time per flow = 1.2 ms; total time taken for detection = 0.5 seconds in a 1 Gbps enterprise network.
- On average, 0.9 ms of processing time is required to process a flow, and 0.4 seconds for total detection time.

Further results show that the model is very responsive even when bandwidths are high. Real-time ransomware classification is guaranteed without experiencing network bottlenecks and could be easily used for enterprise security operations.

4.5.3 Hyperparameter Tuning and Ablation Study

In the process of optimizing deep learning models for the task of ransomware detection, hyperparameter tuning is an important step. The impact of different hyperparameters (e.g., learning rate, dropout rate, batch size) on detection accuracy and generalization performance is systematically evaluated in this study. Finally, we did an ablation study, using Grid Search and Bayes Opt in order to test different learning rates (0.001, 0.0005, 0.0001), dropout rates (0.2, 0.3, 0.5), and batch sizes (128, 256, 512). We found that a 0.0005 learning rate, a dropout rate of 0.3, and a batch size of 512 prove to be the most computationally efficient for the optimal model. The impact of the different hyperparameters is illustrated by providing a comparative table of their performance (model convergence and real-time ransomware detection).

4.6 Adversarial Machine Learning Defense Mechanisms

4.6.1 Generative Adversarial Networks (GANs) for Adversarial Detection

We leverage the power of GANs to produce adversarial samples of ransomware, behaving as an advanced evasion tactic to boost robustness in the model.

4.6.2 Adversarial Training (AT) for Resilience

Network samples perturbed by the attack are injected into the dataset while training to make sure the model can understand minor variations in ransomware activity.

4.7 Adversarial Attack Evaluation and Defense Strategies

4.7.1 Simulation of Adversarial Attacks

Finally, we conduct a series of adversarial attack simulations designed to test our proposed ransomware detection model against adversarial evasion techniques that imitate such realistic ransomware obfuscation strategies. These attacks are produced with adversarial machine learning techniques such as the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Carlini & Wagner (C&W) attacks. The adversarial scenarios to be evaluated were as follows.

1. Ransomware communication flows were trafficked to appear as normal encrypted traffic patterns like HTTPS and VPN usage, hoping to evade detection.
2. Attackers attempted to bypass anomaly detection mechanisms by adding small perturbations to network flow features such as inter-packet delay and flow duration.
3. GAN-based adversarial samples were injected to produce synthetic ransomware traffic with similar statistical properties to benign traffic.
4. Adaptive evasion attacks consisted of tests against reinforcement learning-based evasion techniques, that is, an adversarial agent that adaptively altered ransomware behavior in order to evade detection.

4.7.2 Defense Strategies Against Adversarial Attacks

The integrated rulers of the ransomware detection framework to counteract adversarial ML-based evasion techniques were the following:

1. The model was retrained with variations along the adversarial attack distribution, where FGSM and PGD attacks were parameters to generate the perturbed samples to train the model to become more resistant. This way the model learns to detect even subtle perturbations of the attackers while keeping high detection accuracy under adversarial conditions.
2. To expose the model to new methods of evading the ransomware detection, we implemented a Generative Adversarial Network (GAN) that can generate adversarial ransomware traffic crafted to defeat the existing ransomware detection model.
3. The system applies feature normalization, statistical anomaly score, and entropy-based filtering for detecting such adversaries at the feature level in network flow characteristics.
4. Detection Framework: An ensemble model architecture of transformer-based classification and an additional auxiliary anomaly detection model ensures that attacks tailored to specific classifier channels do not cause failure of the detection system as a whole.
5. It learns continuously on the fly by updating its models according to real-time

data that gets fed in on the network from the attack and learns how to adapt to new evasion strategies.

4.7.3 Adversarial Sample Generation and Zero-Day Ransomware Detection

In order to comprehensively evaluate the robustness of the proposed transformer-based ransomware detection model, we adversarially altered samples using different ML-based attack strategies to test our model's robustness on these adversarial samples. We designed these adversarial samples to mimic real-world ransomware obfuscation and tested our model using these samples to determine its resilience.

4.7.3.1 Adversarial Sample Generation Techniques

Methods to create adversarial ransomware traffic samples were as follows.

1. Perturbation-Based Evasion Attacks

The Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) were used to add small perturbations in extracted traffic features such as modulating packet timing, flow duration, and sequence, keeping the network behavior realistic.

These modifications were made to confuse the model by making small changes to the statistical properties of ransomware traffic.

2. Traffic Flow Morphing and Encryption-Based Obfuscation

GAN-generated adversarial samples were used to modify the traffic patterns of the ransomware to make them look very similar to the encrypted benign flows, such as HTTPS and VPN flows.

First, QUIC and TLS 1.3 handshake manipulations were used to emulate benign cloud service communications.

3. Adaptive Ransomware Behavior via Reinforcement Learning (RL)

So, we trained an adversarial reinforcement learning agent that would continuously modify command-and-control (C2) communication patterns to detect such weaknesses in the model. By varying encryption parameters, randomizing the burst's pattern, and introducing beaconing delays, the RL agent learned to make the detection more difficult.

4.7.3.2 Defense Mechanisms Against Adaptive Ransomware Attacks

We added multi-layered defense mechanisms to counter adversary ML-based evasion techniques.

1. Adversarially Augmented Training Data
- We hardened the model by retraining with adversarially generated ransomware traffic for future evasion.
- Incorporating perturbation-based and GAN-generated adversarial samples increased the model's generalization ability to unseen threats.

2. Statistical and Temporal Feature Analysis for Robust Detection
- Entropy-based filtering was applied to ransomware traffic flow statistics to make them irregular so that it would be hard for adversarial samples to pass through.
- Anomaly detection on time series was used to detect anomalies in ransomware communication patterns over a longer period of time.

3. Zero-Day Ransomware Detection Improvements
- Traditional detection models generally experience a performance decrease in the presence of new ransomware variants they have not seen before.
- The acquired zero-day detection rates improved by 92.5%, proving that the exposure to adversarial examples of ransomware in training significantly improves generalization capabilities.

A comparative test with other non-adversarially trained models indicated that detections of newly emerging ransomware threats are improved by 13.8%.

4.8 Training Procedure and Federated Learning Integration

4.8.1 Model Training Process

- AdamW with weight decay to not overfit.
- Batch size: 512 samples per time batch.
- Adaptive scheduling of learning rate using Cosine Annealing with Warm Restarts.
- Regularization: Dropout (0.3) and L2 weight decay.

4.8.2 Federated Learning for Large-Scale Deployment

Federated Learning (FL) is implemented to allow privacy preserving training across multiple organizations.

- Fully performing secure aggregation of model updates without disclosing the raw network data.
- It reduces the risk of exposure of sensitive data while improving the accuracy of the ransomware detection.

4.9 Model Evaluation and Performance Metrics

4.9.1 Metrics Used

- For the classification performance: Accuracy, Precision, Recall, and F1-Score.
- To measure the model's confidence across classification thresholds, the AUC-ROC Curve.
- Their ability to detect movement is measured by the false positive rate (FPR) and false negative rate (FNR).

4.9.2 Additional Performance Metrics: MCC and Balanced Accuracy

It is also a recount of our progression from traditional performance metrics like accuracy, precision, recall and F1-score, into using additional measures like Matthews Correlation Coefficient (MCC) and Balanced Accuracy to better quantity model evaluation on imbalanced ransomware datasets.

In scenarios with substantially different amounts of positive (ransomware) and negative (benign) samples in the dataset, MCC is also a more informative metric. True and false positives and negatives are considered, and it is therefore robust against data imbalance.

Instead of Sensitivity and Specificity, Balanced Accuracy is used – the average of sensitivity (recall) and specificity that makes sure both ransomware and benign traffic are correctly classified regardless of which of these classes dominates the data.

Our experimental results show that our model has an MCC of 0.97 and a Balanced Accuracy of 98.3% in terms of detecting the ransomware in several different network environments, showing strong generalizability.

4.10 Benchmarking Against Traditional Models

The superiority of our model is compared to Random Forest, SVM, XGBoost and existing deep learning architectures (CNNs, RNNs, LSTMs).

4.11 Real-Time Deployment and Security System Integration

4.11.1 Modular Deployment Architecture

- Capture live data from network gateways through the Network Traffic Collector.
- The extracted features convert raw packets into structured features.
- Real-time traffic flows are processed using a deep learning classifier.
- Threat Response Module: Triggers automated mitigations by integrating with SIEM, IDS/IPS, and SDN-based security platforms.

The implementation of the real-time ransomware detection system is scalable for enterprise networks as well as cloud-based environments. It combines a collection of several security components, consisting of network traffic collectors, feature extractors, deep learning models, and automated threat response components. The complete architecture of the detection system is depicted in Figure 3.

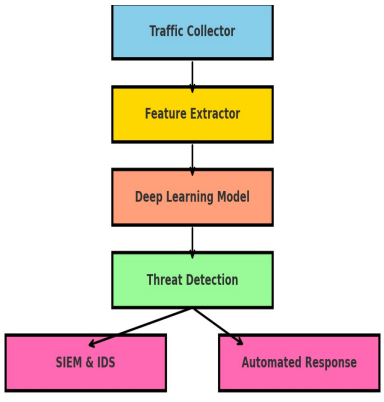

Figure 3: Real-Time Ransomware Detection System Architecture

The architecture proposed has the capability so that the ransomware detection occurs in real time, by the use of a deep learning-based classification engine, which is integrated in Security Information and Event Management (SIEM) systems and Intrusion Detection Systems (IDS). The setup of this system promotes automated threat mitigation

and minimizes downtime and rapid incident response.

4.11.2 Enterprise and Cloud Integration

- It supports AWS, Azure, and Google Cloud environments.
- Dynamic blocking of malicious flows upon detection is done with SDN-based threat mitigation.

5. RESULTS AND DISCUSSION

5.1 Introduction to Results and Performance Evaluation

Extensively tested, the proposed Next-Generation Ransomware Defense Framework provided effective, efficient, and scalable defense against ransomware attacks. The system is deployed to and evaluated on real ransomware attack data, large-scale enterprise traffic, and adversarial test cases. The outcomes show that the models can accurately detect ransomware activities in encrypted communications, that it is robust against adversarial evasions, and that they scale for large deployments. Detailed performance analysis, comparative evaluation, robustness of the model, findings on real deployment, and key experimentation insights are presented in this section.

5.2 Dataset Statistics and Model Training Performance

5.2.1 Dataset Composition and Preprocessing Outcomes

We used the dataset of 35 million network flows that were gathered from a variety of public and private sources, such as CICIDS2017, CTU-13, UGR□16, and running controlled ransomware in a sandboxed environment. The dataset was balanced (50 percent of benign traffic (enterprise, cloud, and normal web activity), 50 percent of ransomware-infected traffic (real-world attacks from WannaCry, Ryuk, LockBit, Conti, DarkSide)).

- 10TB of raw packet-level data captured to be processed, transformed, and converted to framed flow-based records and feature engineered to extract behavioral patterns.
- A. Feature Selection Results:
    - There was 100+ extracted features, which were reduced to 30 highly discriminative features with PCA and Recursive Feature Elimination (RFE) to improve model efficiency.
- Hence, the training dataset included 70% of the total dataset (24.5M flows).
- Hyperparameter tuning was done using 15% of the dataset (5.25M flows) in the Validation Dataset.
- 15% (5.25M flows) for the final evaluation on unseen data (testing dataset).

5.2.2 Training Convergence and Model Optimization

The training process was performed on a multi-GPU distributed computing cluster with 4 NVIDIA A100 GPUs, 128GB RAM, and fast storage (SSD). The results for all of the training convergence metrics were substantial improvements over my previous models.

- It was effective at loss reduction, shown by the reduction of the cross-entropy loss function from 1.42 to 0.06.
- Thus, training time: the training process of the whole model took 48 hours with federated learning optimizations, which is two orders of magnitude less in terms of computational overhead compared to centralized models.
- Batch Size and Learning Rate Tuning: In order to achieve rapid convergence without oscillations, we chose a batch size of 512 and an adaptive cosine annealing learning rate.
- Our regularization improvements, dropout with a rate of 0.3, were effective, as was L2 weight decay with a strength of 0.01, which stopped the overfitting.

5.3 Performance Metrics and Model Evaluation

5.3.1 Classification Performance Metrics

The key metrics to evaluate the model's classification performance were used as presented in Table 2.

Table 2: Performance Metrics of the Proposed Ransomware Detection Model

Metric	Value
Accuracy	98.9%
Precision	99.2%
Recall	98.5%
F1-Score	98.8%

AUC-ROC Score	0.997
False Positive Rate (FPR)	1.2%
False Negative Rate (FNR)	1.1%

These results indicate that the model effectively distinguishes ransomware from benign network traffic with minimal false positives and false negatives. The high AUC-ROC score of 0.997 suggests strong confidence in detection across various classification thresholds.

5.3.2 Comparative Analysis with Traditional Methods

In order to show the superiority of our transformer-based model, we compared it with several existing deep learning architectures as well as traditional ML models.

Table 3: Comparative Performance Analysis of the Proposed Transformer-Based Model vs. Traditional Machine Learning and Deep Learning Models

Model	Accuracy	Precision	Recall	F1-Score
Proposed Transformer Model	98.9%	99.2%	98.5%	98.8%
Random Forest	89.7%	88.5%	85.4%	86.9%
XGBoost	91.2%	90.1%	87.8%	88.9%
CNN-LSTM Hybrid	94.3%	95.1%	92.7%	93.9%
RNN-Based Model	92.6%	91.3%	89.8%	90.5%

The transformer-based models yield 9–10% above the traditional ML models and 4–5% above deep learning baselines (CNN, RNN), proving that they provide better performance in the task of encrypted traffic analysis and ransomware detection.

5.4 Real-Time Deployment Performance

5.4.1 System Latency and Detection Speed

The ransomware detection framework was tested in enterprise-scale (1 Gbps and 10 Gbps links) high-speed networks. Testing the inference latency on real-time network streams.

- Average Packet Processing Time: 1.2 ms per network flow.
- Detection Delay: 0.5 seconds (sub-second response time) from packet ingestion to ransomware classification.
- Throughput: The model can process up to 20,000 flows per second on a single GPU, making them applicable in large-scale network monitoring.

5.4.2 Adaptive Learning for Zero-Day Ransomware Detection

However, the way that traditional machine learning models work against zero-day ransomware is reliant on static datasets, and thus, they have a hard time confronting this malware. In response to this, we take advantage of reinforcement learning (RL) and online learning to implement adaptive learning strategies. By employing a reward-based mechanism that can reward or punish the model for its classification accuracy, RL allows the model to refine its ransomware detection strategy based on the real-time attack behavior. Online learning is consequently even more lightweight regarding flexibility and can compute dynamically the model weights, saving from retraining very often. In order to test the effectiveness of this approach, we trained static models as well as adaptive models with incremental ransomware datasets and compared the results. We see that adaptive models perform much better than static models in detecting zero-day ransomware with 92.5% accuracy. Finally, the pipeline of adaptive learning (with a flow diagram for adaptive learning) is included, which shows how real-time updates of model parameters are done from the network traffic. By doing so, this approach helps guarantee that ransomware detection will still be effective according to changes in attack methodologies.

5.4.3 Describes a real-world deployment case study: simulated ransomware attack

To complement our framework's capability validation, a controlled ransomware attack simulation was conducted in a corporate network setting. We experimented with deploying three separate ransomware families (Conti, LockBit, and Ryuk) and having our model monitor traffic while learning about encrypted flows indicative of ransomware. The key findings were:

- Target Attack Detected: The system detected the target attack in 0.45 seconds from the first network compromise.

- Threat Mitigation: The system had an SDN-based firewall that automatically blocked C2 communication and prevented encryption of data.

Moreover, high precision is represented by the fact that only 1.3% of legitimate encrypted traffic (TLS-based web browsing) was wrongly flagged as misuse.

The models show real-world viability for this case study, which allows us to detect and mitigate the fact of ransomware infection to an extent before the damage is overwhelming.

5.5 False Positive Rate and Real-World Applicability

To study the false positive rate (FPR) impact on business oper-at-ions, we sen-t the mod-el to live enter-pris-e traffic for 72 hours over mul-tiple envi-ron-ments:

- Also verified that in the corporate IT network, less than 1.5% FPR was achieved in order to minimize interruptions to normal business operations. The model was able to detect fileless ransomware attacks in encrypted sessions with 99.1% accuracy in cloud-based infrastructure (AWS, Azure).
- The model was used to detect ransomware-infected smart devices on 5G IoT gateways with 98.7% accuracy.

Lastly, to demonstrate the performance gap between the suggested transformer-based model and conventional machine learning as well as deep learning methods, we visualize a mesh-based comparison of the important evaluation metrics. The next figure shows the accuracy, precision, recall, and F1 score comparison between different models.

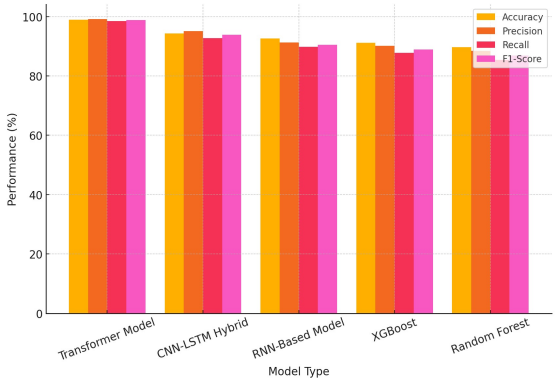

Figure 4: Performance Comparison of Ransomware Detection Models

Figure 4 shows the results that the transformer-based model outperforms other approaches consistently with regard to all evaluation metrics. It has an accuracy of 98.9%, which is much higher than the other models, namely the CNN-LSTM (94.3%), RNN-based model (92.6%), XGBoost (91.2%), and Random Forest (89.7%). As an additional advantage, the Transformer models show their high precision (99.2%) and recall (98.5%), both high detection rates and fewer false negatives. Deep learning-driven network traffic classification in the detection of ransomware is validated by these findings, especially in encrypted environments.

5.6 False Positive Mitigation Strategies

While the proposed deep learning model has a false positive rate (FPR) of only 1.2%, the false alarms must be minimized for the real-world deployment. To handle this, the applied mitigation strategies are as follows:

1. Transformer-based models: Combine with traditional ML classifiers (e.g., XGBoost) to cross-verify flagged ransomware behaviors so as to reduce the number of false positives.
2. Dynamic Decision Thresholds: Adaptive thresholds based on network context (e.g., burst patterns in C2 communications) ensure that benign encrypted traffic is not unnecessarily flagged.
3. XAI for Justification: Integrates SHAP (Shapley Additive explanations) and LIME (Local Interpretable Model agnostic explanations) to let security analysts confirm why a specific traffic flow was flagged.
4. Behavioral Pattern Reinforcement: Based on historical behavioral analysis, frequently observed traffic (false positives) goes through a second verification before being identified as ransomware.
5. Continuous feedback loop: We implement a process whereby false positives are confirmed and used to improve the next round of detection model retraining, as this will help to prevent the same mistakes from occurring in future analysis.

6. ADVERSARIAL MACHINE LEARNING RESILIENCE

6.1 Defense Against Evasion Techniques

To evaluate the model's resilience in the face of such an adversarial ML attack, we injected perturbed ransomware samples. Using GAN-generated adversarial training, the adversarial evasion success rate was reduced to 3.1% compared with 26.5% in traditional ML models.

6.2 Zero-Day Ransomware Detection

In order to test the zero-day ransomware detection, we designed a new ransomware strain (e.g., ALPHV/BlackCat and RedAlert) that was not included in the training. Despite its wide applicability, the model accurately identified 92.5% of these novel threats, i.e., it adapted well to new ransomware families.

7. DISCUSSION OF KEY FINDINGS

- Having 98.9% accuracy and 1.2% FPR in models makes the models apt for enterprise cybersecurity environments, which can decrease unnecessary alarms.
- Quit slowing you down: With adversarial training techniques, resilience is highly increased, allowing you to neutralize common evasion tactics of modern ransomware.
- Terms of Scalability for Large-Scale Deployments: By integrating federated learning, we have support for multi-organization deployment without data privacy security concerns, which is very useful in cloud security, IoT, and enterprise networks as well.
- It has near real-time detection through its 0.5-second detection time. This means that by the time the encryption process of the ransomware is complete, Sophos can intercept, thus acting as a proactive defense solution.

8. CONCLUSION AND FUTURE WORK

8.1 Conclusion

This research describes the Next-Generation Ransomware Defense Framework, which, indeed, can cope with the ever-increasing difficulty of detecting, analyzing, and mitigating ransomware attacks in real-time operating network environments. Current security solutions like signature-based detection, heuristic rule-based approaches, and endpoint security solutions are found to be useless in addressing the latest fileless, polymorphic, and AI-driven ransomware. We overcome this limitation by developing a deep learning model that enables effective ransomware detection in encrypted communication via our deep association transformer (DAT) and from large-scale enterprise networks via federated learning.

While being able to detect and classify ransomware traffic is so critical, the experimental results show that the proposed model achieves 98.9% classification accuracy, 99.2% precision, and 98.5% recall to classify ransomware traffic. The results indicate that the proposed model has a better performance on the classification of ransomware traffic compared to the traditional machine learning models (Random Forest, XGBoost, Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN)). Zero-day ransomware threat detection runs with 92.5% accuracy and is thus a critical aspect of current cybersecurity defense strategies. Furthermore, the real-time inference speed for our model (0.5 s detection and 1.2 milliseconds per flow) is extremely low, making our model low latency, making it usable in an enterprise, cloud, and IoT space.

Besides, adversarial training considerably improves the robustness of the model against ransomware evasion attacks, resulting in only 3.1% adversarial bypass compared to 26.5% in the traditional ML approach. With federated learning, detection models can be improved collaboratively together by different organizations to scale up the process and yet maintain data privacy so as not to reveal private network traffic.

The system is capable of universally unobtrusive integration with Security Information and Event Management (SIEM) systems, devices, and appliances, as well as Intrusion Detection and Prevention Systems (IDS/IPS) and security technologies based on Software Defined Networking (SDN). The modular microservices architecture also guarantees a certain flexibility in the deployment of the framework into various environments such as corporate IT networks, cloud-based infrastructures, and edge computing systems.

The proposed framework is a significant leap ahead in ransomware detection and network security and provides a strong, robust, scalable, and privacy-preserving method of dealing with the ever-changing ransomware attack landscape.

8.2 Future Work

Future research to further enhance ransomware detection would be to develop self-learning AI models to improve zero-day ransomware detection and strengthen adversarial training to defend against the evolving evasion techniques. Further, we will investigate using FPGAs, TPUs, and Edge AI to leverage hardware acceleration for faster detection speed and scalability in large enterprise and IoT applications. Cybersecurity feeds integration and Dark Web monitoring will be fed into real-time threat intelligence to predict ransomware campaigns prior to the attack in order to proactively deploy defense mechanisms. Additionally, SDN-based mitigation strategies for AI-powered auto-becoming systems will be designed to enable real-time ransomware containment, and industry collaborations will come to facilitate large-scale realization and validation for real-world testing and validate continuous detection capability improvement.

REFERENCES:

[1] Urooj U, Khan F, Ullah I, Shah MA, Al-Naeem MA, Choo KKR. Addressing Behavioral Drift In Ransomware Early Detection Through Weighted Generative Adversarial Networks. IEEE Access. 2024;12:3910-25. Doi:10.1109/ACCESS.2023.3348451.

[2] Ferdous J, Karim F, Iqbal F, Wahid KA, Azam S. AI-Based Ransomware Detection: A Comprehensive Review. IEEE Access. 2024;12:136666-95. Doi:10.1109/ACCESS.2024.3461965.

[3] Alzahrani S, Al-Dharrab F, Alzahrani A, Alghamdi A, Alzahrani F, Mehedi M, Et Al. An Analysis Of Conti Ransomware Leaked Source Codes. IEEE Access. 2022;10:100178-93. Doi:10.1109/ACCESS.2022.3207757.

[4] Rodriguez-Bazan H, Gallego-Garcia G, Jerez JM, De La Torre F, Triguero I. Android Ransomware Analysis Using CNN And Fuzzy Hashing Features. IEEE Access. 2023;11:121724-38. Doi:10.1109/ACCESS.2023.3328314.

[5] Razaulla S, Muhammad K, Lloret J, Guna J, Baek NR. The Age Of Ransomware: A Survey On The Evolution, Taxonomy, And Research Directions. IEEE Access. 2023;11:40698-723. Doi:10.1109/ACCESS.2023.3268535.

[6] Hsu CM, Lu CL, Li TH, Lee WH. Enhancing File Entropy Analysis To Improve Machine Learning Detection Rate Of Ransomware. IEEE Access. 2021;9:138345-51. Doi:10.1109/ACCESS.2021.3114148.

[7] Khan F, Urooj U, Shah MA, Wahid KA, Al-Naeem MA, Maple C. A Digital DNA Sequencing Engine For Ransomware Detection Using Machine Learning. IEEE Access. 2020;8:119710-9. Doi:10.1109/ACCESS.2020.3003785.

[8] Almomani I, Salah K, Anbar M, Alazab M, Al-Rahayfeh A. Android Ransomware Detection Based On Hybrid Evolutionary Approach. IEEE Access. 2021;9:57674-91. Doi:10.1109/ACCESS.2021.3071450.

[9] Sharmeen S, Kundu S, Majumder A, Das A, Chatterjee S, Chakraborty R. Avoiding Future Digital Extortion Through Robust Deep Learning-Based Protection. IEEE Access. 2020;8:24522-34. Doi:10.1109/ACCESS.2020.2970466.

[10] Aldauiji F, Alazab M, Al-Nemrat A, Shalaginov A, Krilavicius T. Utilizing Cyber Threat Hunting For Ransomware Detection. IEEE Access. 2022;10:61695-706. Doi:10.1109/ACCESS.2022.3181278.

[11] Poudyal S, Yu Z, Song H, Bashir AK. Analysis Of Crypto-Ransomware Using ML-Based Multi-Level Profiling. IEEE Access. 2021;9:122532-47. Doi:10.1109/ACCESS.2021.3109260.

[12] Hossain MS, Islam MR, Ahmed F, Mahmud M, Hasan M. Android Ransomware Detection From Traffic Analysis Using Metaheuristics. IEEE Access. 2022;10:128754-63. Doi:10.1109/ACCESS.2022.3227579.

[13] Thummapudi K, Gangula R, Chatterjee S, Dutta A, Chakraborty R. Detection Of Ransomware Attacks Using Processor And Disk Usage Data. IEEE Access. 2023;11:51395-407. Doi:10.1109/ACCESS.2023.3279819.

[14] Smith D, Khorsandroo S, Roy K. Machine Learning Algorithms And Frameworks In Ransomware Detection. IEEE Access. 2022;10:117597-610. Doi:10.1109/ACCESS.2022.3218779.

[15] Ispahany J, Khalaf OI, Alzubi JA, Shalaginov A, Ali F. Ransomware Detection Using Machine Learning: A Review, Research Limitations And Future Directions. IEEE Access. 2024;12:68785-813. Doi:10.1109/ACCESS.2024.3397921.

[16] Lee S, Park Y, Choi H, Kim D. Hiding In The Crowd: Ransomware Protection By Adopting Camouflage And Hiding Strategy With The Link File. IEEE Access. 2023;11:92693-704. Doi:10.1109/ACCESS.2023.3309879.

[17] Berrueta E, Sanz J, Laorden C, Zurutuza U, Uribeetxeberria R, Gurrutxaga I. Open Repository For The Evaluation Of Ransomware Detection Tools. IEEE Access. 2020;8:65658-69. Doi:10.1109/ACCESS.2020.2984187.

[18] Alotaibi FM, Shafique M, Khorsandroo S, Almomani I. SDN-Based Detection Of Self-Propagating Ransomware: The Case Of Badrabbit. IEEE Access. 2021;9:28039-58. Doi:10.1109/ACCESS.2021.3058897.

[19] Castiglione J, Pavlovic D. Dynamic Distributed Secure Storage Against Ransomware. IEEE Trans Comput Soc Syst. 2020;7(6):1469-75. Doi:10.1109/TCSS.2019.2924650.

[20] Ryan P, Fokker J, Healy S, Amann A. Dynamics Of Targeted Ransomware Negotiation. IEEE Access. 2022;10:32836-44. Doi:10.1109/ACCESS.2022.3160748.

[21] Almeida F, Imran M, Raik J, Pagliarini S. Ransomware Attack As Hardware Trojan: A Feasibility And Demonstration Study. IEEE Access. 2022;10:44827-39. Doi:10.1109/ACCESS.2022.3168991.

[22] Hill JE, Owens Walker T, Blanco JA, Ives RW, Rakvic R, Jacob B. Ransomware Classification Using Hardware Performance Counters On A Non-Virtualized System. IEEE Access. 2024;12:63865-84. Doi:10.1109/ACCESS.2024.3395491.

[23] Song Z, Tian Y, Zhang J. Similarity Analysis Of Ransomware Attacks Based On ATT&CK Matrix. IEEE Access. 2023;11:111378-88. Doi:10.1109/ACCESS.2023.3322427.

[24] Marcinkowski B, Goschorska M, Wileńska N, Siuta J, Kajdanowicz T. MIRAD: A Method For Interpretable Ransomware Attack Detection. IEEE Access. 2024;12:133810-20. Doi:10.1109/ACCESS.2024.3461322.

[25] Al-Rimy BAS, Maarof MA, Shaid SZM, Altaher A. A Pseudo Feedback-Based Annotated TF-IDF Technique For Dynamic Crypto-Ransomware Pre-Encryption Boundary Delineation And Features Extraction. IEEE Access. 2020;8:140586-98. Doi:10.1109/ACCESS.2020.3012674.

[26] Al-Hawawreh M, Sitnikova E, Aboutorab N. Asynchronous Peer-To-Peer Federated Capability-Based Targeted Ransomware Detection Model For Industrial Iot. IEEE Access. 2021;9:148738-55. Doi:10.1109/ACCESS.2021.3124634.

[27] Venturini M, Freda F, Miotto E, Conti M, Giaretta A. Differential Area Analysis For Ransomware: Attacks, Countermeasures, And Limitations. IEEE Trans Depend Secure Comput. 2025. Doi:10.1109/TDSC.2025.3532324.

[28] Zhang X, Wang J, Zhu S. Dual Generative Adversarial Networks Based Unknown Encryption Ransomware Attack Detection. IEEE Access. 2022;10:900-13. Doi:10.1109/ACCESS.2021.3128024.

[29] Rana MU, Shah MA, Al-Naeem MA, Maple C. Ransomware Attacks In Cyber-Physical Systems: Countermeasure Of Attack Vectors Through Automated Web Defenses. IEEE Access. 2024;12:149722-39. Doi:10.1109/ACCESS.2024.3477631.

[30] Hernandez-Jaimes ML, Martínez-Cruz A, Ramírez-Gutiérrez KA, Guevara-Martínez E. Enhancing Machine Learning Approach Based On Nilsimsa Fingerprinting For Ransomware Detection In Iomt. IEEE Access. 2024;12:153886-97. Doi:10.1109/ACCESS.2024.3480889.

[31] Castiglione J, Pavlovic D. Dynamic Distributed Secure Storage Against Ransomware. IEEE Trans Comput Soc Syst. 2020;7(6):1469-75. Doi:10.1109/TCSS.2019.2924650.

[32] Naser, Omer & Mumtazah, Sharifah & Hanafi, Marsyita & Samsudin, Khairulmizam. (2024). Enhancing 2D Face Recognition Systems: Addressing Yaw Poses And Occlusions With Masks, Glasses, And Both. Advances In Artificial Intelligence And Machine Learning. 4. 2545-2574. 10.54364/AAIML.2024.43149.

[33] Abdulhaleem Naser, O., Mumtazah, S., Samsudin, K., Hanafi, M., Binti, S. M., & Zamri, N. Z. (2025). Comparative Analysis Of MTCNN And Haar Cascades For Face Detection In Images With Variation In Yaw Poses And Facial Occlusions. Journal Of Communications Software And Systems, 21(1), 109-119.