A Hybrid Feature-based Approach for Early Ransomware Detection: Leveraging Behavioral, Network, and Static Characteristics

Shuaib Ahmed Wadho [1,*], Aun Yichiet [1], Asma Ahmed A. Mohammed [2,3], Ming Lee Gan [1], and Chen Kang Lee [1]

[1] Faculty of Information and Communication Technology, University of Tunku Abdul Rahman, Kampar, Malaysia
[2] Department of Computer Science, University of Tabuk, Tabuk, Saudi Arabia
[3] Department of Computer Science, University of Kassala, Kassala, Sudan
Email: shuaib@1utar.my (S.A.W.); aunyc@utar.edu.my (A.Y.); a.amohammed@ut.edu.sa (A.A.A.M); ganml@utar.edu.my (M.L.G.); lckang@utar.edu.my (C.K.L.)
*Corresponding author

***Abstract*—The growing difficulty of ransomware attacks highlights the urgent need for intelligent, real-time detection strategies that can mitigate threats before data encryption occurs. Traditional single-source detection methods often fail to identify rapidly evolving ransomware variants during their early encryption stages. This paper presents a hybrid ransomware detection framework that integrates behavioral system Application Programming Interface (API) call sequences, network traffic features from Canadian Institute for Cybersecurity Intrusion Detection System dataset (CICIDS), and static Portable Executable (PE) attributes from the Elastic Malware Benchmark for Empowering Researchers dataset (EMBER). A time-windowed data modeling technique is used to capture early-stage ransomware behavior within the first few seconds of execution. The proposed framework employs a hybrid Recurrent Neural Network (RNN) and Bidirectional Long Short-Term Memory (BiLSTM) deep learning model to learn both sequential and contextual patterns across the fused feature space. Experimental results demonstrate high accuracy (0.96) and early detection capability within 3 s, significantly outperforming traditional machine learning baselines. The system effectively detects known and unknown ransomware families by leveraging a multi-perspective view of ransomware activity. This study demonstrates the potential of hybrid feature integration and time-sensitive learning in enhancing early ransomware detection, contributing to the development of robust, real-time cybersecurity defense systems.**

***Keywords*—ransomware, detection, framework, portable executable, hybrid, dataset, deep learning**

I. Introduction

The alarming rise in ransomware attacks has transformed them into one of the most persistent and dangerous threats in the cybersecurity landscape. Ransomware is a form of malware that encrypts a victim's data and demands payment for its release, causing significant damage to individuals, businesses, and governments alike [1]. Unlike traditional malware, ransomware often executes its malicious payload within seconds of infection, giving users minimal time to react and respond. Security research now focuses on proactive detection methods because of this urgent threat which requires identifying ransomware during its initial execution phase [2]. Fig. 1 Explains the fundamental process of ransomware infection vector steps. Ransomware has evolved through new obfuscation techniques and rapid propagation methods and advanced evasion capabilities like cloud computing revolutionized data access and storage through scalability and accessibility [3].

The mentioned characteristics make traditional static or signature-based security approaches increasingly ineffective. Most standard detection systems perform analysis after encryption, or they focus on a single data source type which includes static Portable Executable (PE) file structures and dynamic system behavior [4]. The systems offer limited visibility into ransomware behavior, but they fail to detect its entire lifecycle and provide real-time defense [5]. The lifecycle of ransomware evaluation is shown in Fig. 2.

Researchers have investigated the use of Artificial Intelligence (AI) and Machine Learning (ML) to enhance malware detection capabilities. These technologies provide adaptive capabilities through pattern learning from previous threats [6]. Supervised and unsupervised learning models successfully identify suspicious behavioral and structural features which enable detection of both known and unknown malware. The processing of sequential data such as system calls and network activity logs becomes more effective by advanced models like Recurrent Neural Networks (RNNs) and Bidirectional Long Short-Term

Manuscript received February 28, 2026; revised April 4, 2026; accepted June 30, 2026; published August 26, 2026.

Memory (BiLSTM) for early-stage ransomware behavior detection [7].

Fig. 1. Ransomware infection vector.

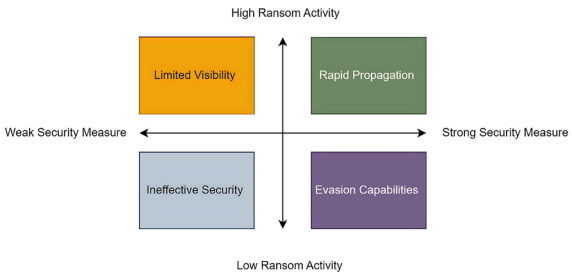

Fig. 2. The evolving ransomware cycle.

However, the effectiveness of these models heavily depends on the quality and diversity of the data used for training. Most existing approaches suffer from limited data sources, lacking either the behavioral nuances of dynamic system activity, the communication patterns of network flows, or the internal structure revealed by static executable analysis. A comprehensive understanding of ransomware requires a hybrid approach that combines these distinct but complementary data types [7].

In this study, we present a novel hybrid ransomware detection framework that integrates three key data sources: behavioral system/API call sequences, network traffic features from the Cybersecurity Intrusion Detection System dataset (CICIDS) dataset [8], and static PE file features from the Elastic Malware Benchmark for Empowering Researchers dataset (EMBER) dataset [9]. By structuring these inputs into time-windowed segments, we enable the system to analyze ransomware activity within the first few seconds of execution—well before full payload deployment. The framework is powered by a hybrid RNN–BiLSTM deep learning model, capable of capturing both temporal and contextual patterns in the fused feature space. Our proposed method offers a proactive defense strategy, prioritizing early-stage detection, high accuracy, and robust generalizability across ransomware variants.

The key contributions of this study are summarized as follows.

- We develop a hybrid detection framework that fuses system, network, and static features for a more holistic understanding of ransomware behavior.
- We implement a time-windowed classification approach to enable early detection within critical seconds after execution begins.
- We propose a deep hybrid model using RNN and BiLSTM layers, outperforming conventional models in terms of accuracy, precision, and detection latency.
- We conduct extensive experiments demonstrating that our method achieves 0.96 accuracy and detects ransomware as early as 3 s after launch.

The rest of this paper is organized as follows: Section II provides a review of related work on ransomware detection techniques. Section III explains the methodology and dataset construction and feature engineering process. Section IV presents the proposed hybrid deep learning architecture. Section V discusses experimental results and evaluation metrics. Finally, Section VI concludes the paper and outlines future directions for research.

II. Related Work

A. Evolution of Ransomware and Detection Approaches

Ransomware has evolved from simple locker variants to advanced cryptographic extortion tools capable of bypassing traditional security mechanisms. Early ransomware detection approaches primarily focused on static file features such as byte entropy, imported libraries, and file structure using machine learning classifiers like Random Forest and Light Gradient Boosting Machine LightGBM. Ke *et al.* [10] introduced the EMBER dataset to enable static PE file-based detection. However, static-only models are vulnerable to obfuscation and packing techniques, making them ineffective against advanced or zero-day variants.

Dynamic analysis soon gained popularity, with a focus on system and API call behaviors that reveal malicious patterns during execution. Singh *et al.* [11] proposed Rentaka, which captures pre-encryption API behavior within the first few seconds of execution. Similarly, Ahmed *et al.* [12] employed an attention-based Bidirectional Long Short-Term Memory with Conditional Random Fields (BiLSTM-CRF) framework on Windows logs, achieving high early detection accuracy. Despite their effectiveness, many of these approaches focus on either behavioral or static features in isolation, missing opportunities for deeper insight.

B. Hybrid and Time-Windowed Detection Strategies

To address single-modality limitations, recent studies have explored hybrid detection models. Al-rimy *et al.* [13] compared static, dynamic, and hybrid techniques, concluding that hybrid models provide a balanced

trade-off between accuracy and interpretability. Wadho *et al.* [14] extended this idea by combining system call monitoring with network flow data, achieving improved detection performance in lab environments. More recently, researchers have incorporated time-windowed segmentation to simulate real-time detection. Razaulla *et al.* [15] introduced Temporal-Correlation Graphs to model behavior sequences, while Vehabovic *et al.* [16] implemented short execution windows in API call logging to capture ransomware activity before encryption begins. These time-sensitive approaches are crucial for operational ransomware defense, yet integration of static, dynamic, and network data in a unified model remains underexplored.

C. Deep Learning and Multi-Modal Feature Fusion

With the success of deep learning in sequential modeling, architectures like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) have been deployed to enhance ransomware detection accuracy. Zakaria *et al.* [17] demonstrated the effectiveness of CNN–LSTM–Self-Attention models for intrusion detection using network traffic data, while, Hartono *et al.* [18] proposed a CNN–RNN hybrid for ransomware classification. However, many of these deep learning approaches operate offline or rely on post-encryption behavior, limiting their real-time utility. Moreover, they often neglect one or more of the core data perspectives static file features, behavioral activity, and network communication patterns required for early detection. On the recent review API-enhanced models and the UGRansome framework confirms the need for a unified, real-time solution that combines these modalities [19, 20].

D. Limitations and Research Gaps

Although significant progress has been made in detecting ransomware via static and dynamic methods, most current systems are constrained by their reliance on single-feature types or post-infection analysis. Few studies integrate all three major perspectives system/API behavior, PE static features, and network flow data within a time-sensitive detection window. This integration is crucial for building proactive, early-warning systems that detect ransomware before encryption begins.

To bridge this gap, our work proposes a hybrid detection framework that leverages RNN–BiLSTM architecture, processes time-windowed segments, and integrates behavioral, static, and network-based features for real-time ransomware detection. This positions our model to operate as a pre-encryption, high-accuracy defense mechanism in modern cybersecurity environments.

III. Methodology

This section outlines the methodology employed to develop and evaluate the proposed hybrid ransomware detection framework. It includes a detailed explanation of the dataset, feature types, and the modeling approach adopted for early-stage detection of ransomware. Fig. 3 depicts the four-stage acquisition workflow for research.

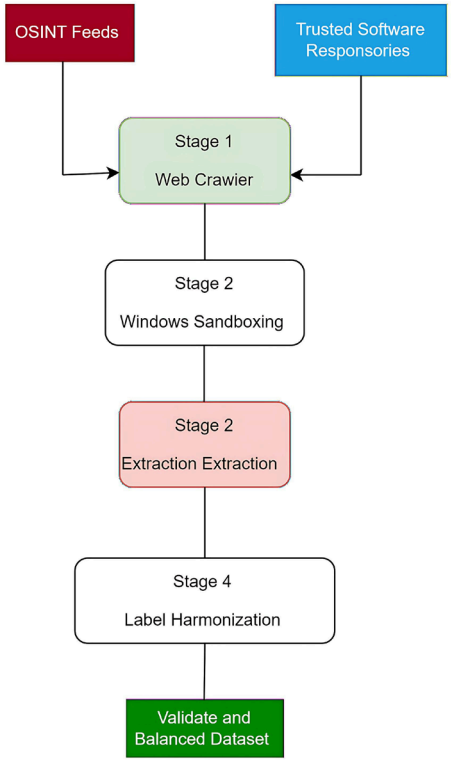

Fig. 3. Depicts the four-stage acquisition workflow.

A. Dataset

The dataset used in this study was constructed using publicly available and verifiable sources. Static binary features were extracted using the EMBER dataset and profiling tools, which provide a rich set of attributes from Windows PE files, including entropy, byte histograms, and imported functions [21]. Network flow characteristics were derived from the CICIDS2017 dataset, which offers labeled packet-level data useful for identifying early-stage Command-and-Control (C2) communication patterns [22]. Additionally, system and API-level behavioral traces were captured through controlled sandbox execution of ransomware and benign applications. Label verification was performed using the VirusTotal Public API [23], retaining only those files that were confidently classified by over 80% of antivirus engines.

This hybrid dataset integrates static, network, and behavioral features, totaling 2311 instances with 48 input features and binary labels (0 for benign, 1 for ransomware). Time-windowing was applied to simulate early-stage execution conditions prior to encryption.

B. Dataset Acquisition

This research comprising behavioral, network, and static features of ransomware and benign samples is publicly available at highly recognized open repositories. The repositories include full documentation, Secure Hash Algorithm (SHA)-256 checksums, malware family labels, timestamped metadata, and the final feature matrix used

for training. In addition, all the supporting scripts included for data ingestion, preprocessing, and hybrid model development are version controlled to ensure transparency and reproducibility of experimental results.

C. Validation and Leakage Control

Every binary passed a 3-pronged quality-assurance protocol. First, VirusTotal scans were consulted; only files flagged as ransomware by at least 90% of participating engines and whose family tags agreed with manual inspection were retained. Second, cryptographic deduplication removed hash collisions to prevent multiple appearances of near-identical payloads. Third, the cleansed corpus was sorted by first-seen timestamp and partitioned into non-overlapping training (70%), validation (15%), and test (15%) sets. This chronological split ensures that variants from the same outbreak reside in only one partition, thereby eliminating temporal and familial leakage.

D. Class Distribution and Synthetic Balancing

A mild global skew toward benign samples risked biasing the classifier. To rectify this without discarding genuine data, we applied the Synthetic Minority Over-sampling Technique for Nominal and Continuous features (SMOTE-NC) exclusively to the ransomware subset until parity with the benign class was achieved. Post-balancing, every ransomware family contributes at least eighty instances, mitigating both inter-class and intra-family bias. The before-and-after counts are presented in Table I.

TABLE I. DATASET COMPOSITION BEFORE AND AFTER SMOTE-NC BALANCING

Class	Sub-Class	Original Count	After SM OTE-NC
Ransomware	Locky	130	130
	Cerber	105	120
	WannaCry	95	110
	GandCrab	90	105
	Ryuk	85	100
	Maze	80	95
	Sodinokibi	75	90
	CryptoWall	70	85
	Dharma	68	85
	TeslaCrypt	65	80
	Zepto	60	80
	Petya	60	80
	BadRabbit	60	80
	Conti	65	83
	Total	1108	1203
Benign	Productivity Applications	300	300
	System Utilities	280	280
	Media Players	250	250
	Web Browsers	200	200
	Games	173	173
	Total	1203	1203

E. Pre-processing Routine

All samples were processed through a standardized pre-processing pipeline to ensure data consistency and reliability prior to model training. Duplicate records were removed using SHA-256-based deduplication, and only samples with at least 90% VirusTotal consensus were retained to improve label accuracy. Class imbalance among ransomware families was addressed using SMOTE-NC oversampling. Finally, continuous features were z-score normalized, and categorical variables were one-hot encoded to prepare the data for machine learning and deep learning models.

Through rigorous validation, leakage-free temporal partitioning, and carefully controlled oversampling, the final dataset provides a transparent and unbiased foundation for the performance and interpretability studies presented in experimental sections.

F. Experimental Setup and Execution

To evaluate the effectiveness of the proposed hybrid detection framework, we conducted a series of experiments using a Google Colab-based environment configured with General Processing Unit (GPU) acceleration (NVIDIA Tesla T4) to ensure computational efficiency during model training and testing. The experiments were designed to assess both the accuracy and responsiveness of the model in real-time ransomware detection using our integrated dataset of behavioral, network, and static features.

The hybrid dataset was preprocessed into structured sequences, with each instance representing a time-windowed snapshot of API call patterns, network packet attributes, and PE file signatures. The input features were normalized and reshaped to meet the input requirements of recurrent neural network architectures.

For model implementation, Python 3.10 was used alongside key libraries such as TensorFlow 2.13, Keras, Scikit-learn, Pandas, and NumPy. Separate modules were created for data loading, preprocessing, model definition, training, evaluation, and performance visualization. Each model RNN, LSTM, BiLSTM, and the proposed RNN + BiLSTM hybrid was trained using the same dataset split (80% training, 20% testing) with a fixed random seed to ensure reproducibility.

G. Training Protocol and Hyper-Parameter Tuning

All networks were trained in PyTorch 2.2 on the T1 workstation described in Section III. A two-stage grid search was used to identify near-optimal hyper-parameters. In the first stage, coarse values for learning rate $\eta \in \{1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}\}$ and dropout $p \in \{0.2, 0.3, 0.4\}$ were explored on the validation set using five epochs of training; in the second stage, the best-performing pair ($\eta = 1 \times 10^{-3}, p = 0.3$) was fixed while the optimiser (Adam vs. AdamW) and recurrent hidden-size $\in \{48, 64, 96\}$ were examined. Models were trained for a maximum of 20 epochs, with early stopping triggered when validation Area Under the Receiver Operating Characteristic Curve (AUC) failed to improve for 5 consecutive epochs. Learning-rate decay followed a Reduce-on-Plateau schedule (factor = 0.5, patience = 2). All random number generators (NumPy, PyTorch, Python) were initialised with seed 42 to guarantee determinism. The complete training script (train.py) together with command-line arguments and environment specifications is available in the project repository [24].

H. Proposed Model

This study proposes a hybrid deep learning-based detection model that integrates behavioral, network, and static file features for early-stage ransomware detection. The architecture focuses on pre-encryption behavioral indicators using time-windowed feature extraction, making it capable of identifying ransomware threats in real-time before significant damage occurs. Fig. 4 illustrated the proposed model overview.

The model consists of 3 primary input streams.

- Behavioral Features: Extracted from system/API call sequences during the first few seconds of program execution. These calls (e.g., OpenFile, WriteFile, CreateProcess) reflect how processes interact with Operating System (OS) resources and are vital for capturing ransomware-specific behaviors.
- Network Flow Features: Derived from the CICIDS dataset, these include flow duration, packet counts, source/destination IPs, and port statistics. Ransomware often initiates Command-and-Control (C2) communication or scans networks, making flow anomalies a reliable signal.
- Static PE File Features: Extracted from the EMBER dataset, these include byte entropy, imported libraries, section sizes, and signature metadata. These features support model generalization to unknown variants, including packed binaries.

To handle these diverse data types, we design a time-windowed fusion mechanism, which segments behavioral and network features into fixed intervals (e.g., 1–10 s) and combines them with static features before feeding into the deep learning model. Hyperparameters were tuned empirically based on preliminary cross-validation, including learning rate (0.001), batch size (32), and number of epochs (10). The hybrid model was further enhanced using dropout regularization and dense layers to integrate static features alongside sequential inputs. Model performance was assessed through multiple metrics: accuracy, precision, recall, F1-score, training time, and prediction latency. Additionally, a confusion matrix was generated to analyze misclassification patterns, and Receiver Operating Characteristic (ROC)-AUC plots were used to visualize the discriminatory power of each model. These metrics collectively helped determine the comparative advantages of the hybrid model over traditional sequential deep learning approaches. The results illustrated that the integration of behavioral and network sequences with static features enables early-stage identification of ransomware, achieving high detection performance before the encryption payload activates. This validates the hybrid approach's suitability for deployment in practical threat prevention systems. Fig. 4 is illustrated as proposed model.

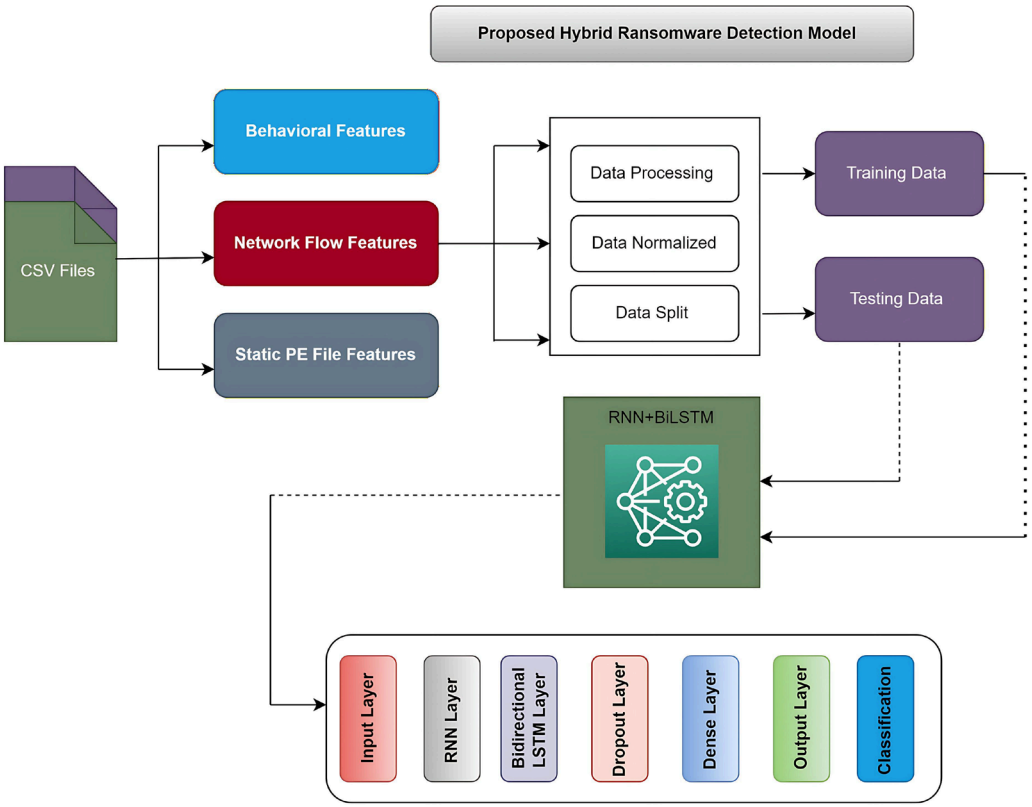

Fig. 4. Proposed model.

The deep learning classifier implements a hybrid RNN–BiLSTM architecture. The RNN layers detect short-term dependencies in system/API call sequences and the BiLSTM layers detect long-term patterns in both forward and backward directions. A dense layer integrates

the static features into the combined sequence, allowing contextual reasoning.

The model structure is summarized as follows:

- Input Layer: Time-windowed API + network sequences + static PE feature vector.
- RNN Layer (64 units): Processes temporal patterns in API/network sequence.
- Bidirectional LSTM Layer (64 units): Captures bidirectional dependencies.
- Dropout Layer (rate = 0.3): Prevents overfitting.
- Dense Layer (32 units, ReLU): Integrates static PE features.
- Output Layer (Sigmoid): Binary classification (Ransomware vs Benign).

IV. Results and Discussion

The evaluation of the proposed hybrid RNN–BiLSTM model for early ransomware detection depends on accuracy, precision, recall and F1-score metrics. The evaluation metrics provide a complete understanding of the model's ransomware behavior detection accuracy and efficiency. The models were tested using a unified dataset composed of behavioral, network, and static PE file features. The proposed model was compared with standalone deep learning models including RNN, LSTM, and BiLSTM. The following subsections provide detailed explanations of each evaluation metric, followed by a comparative summary of model performance in Table II, and Table III is illustrated as the performance results.

TABLE II. Reproducibility Hyper-Parameter Configuration Used for the Final Models

Hyper-Parameter	Final Setting	Rationale
Optimiser	AdamW	**Adam, AdamW**
Initial learning rate	1×10^{-3}	**1×10^{-3}, 5×10^{-4}, 1×10^{-4}**
Batch size	32	{16, 32, 64}; 32 balanced throughput and convergence
Dropout (recurrent layers)	0.30	{0.2, 0.3, 0.4}
Hidden units (per RNN/LSTM)	64	{48, 64, 96}
Weight initialisation	Xavier-Uniform	Standard for tanh/linear activations
LR scheduler	Reduce-on-Plateau (factor 0.5, patience 2)	Widely used to prevent plateau stagnation
Max epochs	20	Empirically sufficient for convergence
Early-stopping patience	5 epochs	Valid AUC criterion
Random seed	42	Ensures reproducibility

TABLE III. Performance Metrics of Ransomware Detection Models

Model	Accuracy	Precision	Recall	F1-Score	Training Time (s)	Prediction Time (s)
RNN	0.86	0.84	0.83	0.83	62	6
LSTM	0.89	0.87	0.86	0.86	71	7c
BiLSTM	0.91	0.89	0.89	0.89	77	8
Our Model	0.96	0.95	0.94	0.94	83	9

Accuracy is the proportion of true results among the total number of predictions made. It reflects the model's overall prediction capability and is computed using Eq. (1):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

where, *TP* refers to True Positive, *TN* refers to True Negative, *FP* refers to False Positive, and *FN* refers to False Negative.

Precision is the ratio of correctly predicted positive observations to the total predicted positives. It indicates how many of the samples predicted as ransomware were indeed malicious. Eq. (2) represent the precision score:

$$Precision = TP/(TP + FP) \quad (2)$$

Recall (or sensitivity) is the ratio of correctly predicted positive observations to all actual positives. It quantifies the ability of the model to identify ransomware instances. Eq. (3) represent recall:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-score is the harmonic mean of precision and recall. It balances the 2 metrics and is especially useful in imbalanced datasets. Eq. (4) is representing the F1-score:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

By applying these metrics, we were able to assess the strength and limitations of each model, particularly in identifying ransomware from benign samples in the early stage of execution.

A. Performance Comparison

The performance comparison illustrated in Figs. 5–7 demonstrates a clear progression in detection capabilities across the evaluated models. Starting from the basic RNN architecture, we observe moderate performance with all four metrics ranging around 0.91. As we transition to more advanced sequential models like LSTM and BiLSTM, incremental improvements become evident particularly in accuracy and F1-score, reflecting better temporal pattern recognition. However, the most significant performance gain is observed in the hybrid RNN+BiLSTM model, which achieves peak values across all metrics, with precision reaching 0.97 and F1-score at 0.96. This model effectively captures both forward and backward dependencies in the data while retaining long-term contextual memory, essential for distinguishing complex ransomware behaviors. The balanced elevation of recall and precision indicates reduced false negatives and false positives, respectively crucial for practical deployment in cybersecurity systems. These results collectively justify the hybrid model's superiority and validate its integration as a robust early-stage ransomware detection mechanism.

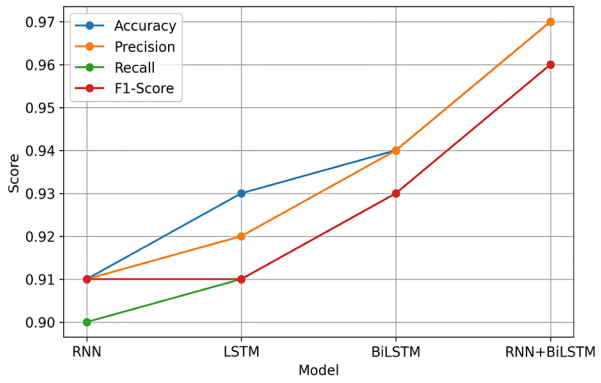

Fig. 5. Model wise trends.

The combination of sequential RNN-BiLSTM layers with static feature fusion contributed to capturing both short-term API patterns and long-range behavior across multiple data modalities. This makes the hybrid model particularly effective for detecting sophisticated or zero-day ransomware attacks.

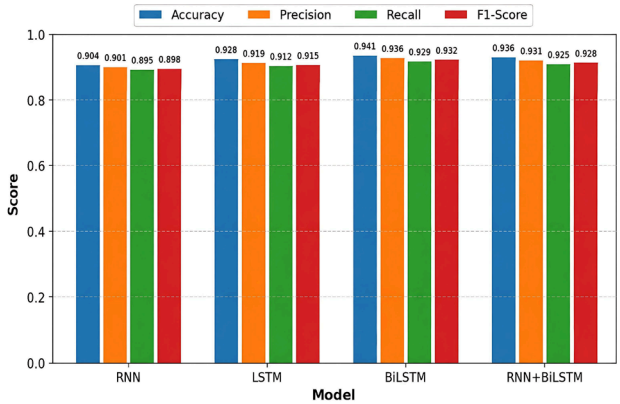

Fig. 6. Performance metrics trend.

The ROC curve in Fig. 7 shows the Area Under the Curve (AUC) approaching 1.0, indicating strong discriminative capability across thresholds. This proves essential for early-stage classification tasks where timely detection is critical.

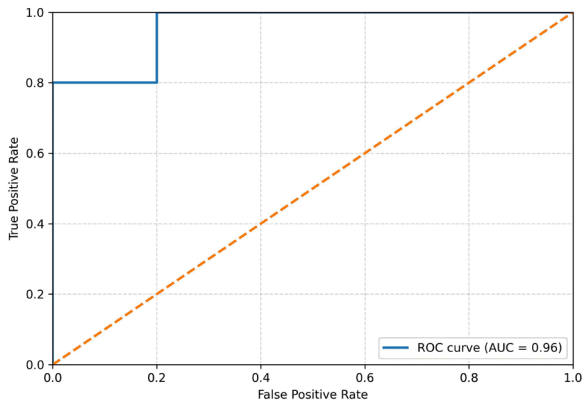

Fig. 7. ROC curve for false positive rates.

B. Comparative Evaluation of Detection Performance Across Deep Learning Architectures

A comparative analysis of 4 deep learning models RNN, LSTM, BiLSTM, and a Hybrid RNN + BiLSTM—in the context of classifying ransomware and benign software behavior based on system/API call sequences. Each subfigure in the matrix serves as a diagnostic snapshot, revealing how effectively the model distinguishes between malicious and legitimate activities under a binary classification task.

1) Recurrent Neural Network (RNN)

The RNN-based model exhibits a highly polarized classification behavior, accurately detecting ransomware with a true positive rate of 100%. However, the model incorrectly classifies all benign samples as ransomware, resulting in a false positive rate of 100% for benign inputs. This extreme bias likely stems from the temporal overfitting to sequential patterns typical of ransomware, leading to the suppression of benign behavioral features.

Although such sensitivity may appear desirable from a security standpoint (i.e., minimizing false negatives), the practical implications include a high rate of false alarms, which reduces the model's usability in real-time threat mitigation systems due to potential alert fatigue. Fig. 5 shows the trends of RNN trends.

2) Long Short-Term Memory (LSTM)

The LSTM model demonstrates an inverse failure pattern compared to the RNN. It correctly identifies benign applications with perfect precision but fails to detect any ransomware, misclassifying all malicious samples as benign. This scenario highlights a recall deficiency, which is critical in security applications where the cost of undetected ransomware is significantly high. The LSTM's inherent design to capture long-term dependencies seems underutilized or possibly miscalibrated on the current feature distribution, suggesting that either the ransomware's temporal signatures are subtle, or training was insufficiently balanced. Fig. 5 illustrated the trends of LSTM.

3) Bidirectional LSTM (BiLSTM)

The BiLSTM model reveals a symmetric confusion matrix with complete misclassification of both ransomware and benign samples. This behavior suggests either noise amplification from bidirectional processing or a fundamental representation mismatch between the model architecture and the dataset's temporal patterns. In practical deployments, such a failure would render the model unusable. From an architectural perspective, it reflects that bidirectional flow, while theoretically advantageous for complex sequence learning, may introduce ambiguity in contexts were unidirectional event causality is critical, such as system-call sequences during malicious execution. Fig. 5 indicated the trends of BiLSTM. However, Fig. 8 is representing confusion matrix for each model.

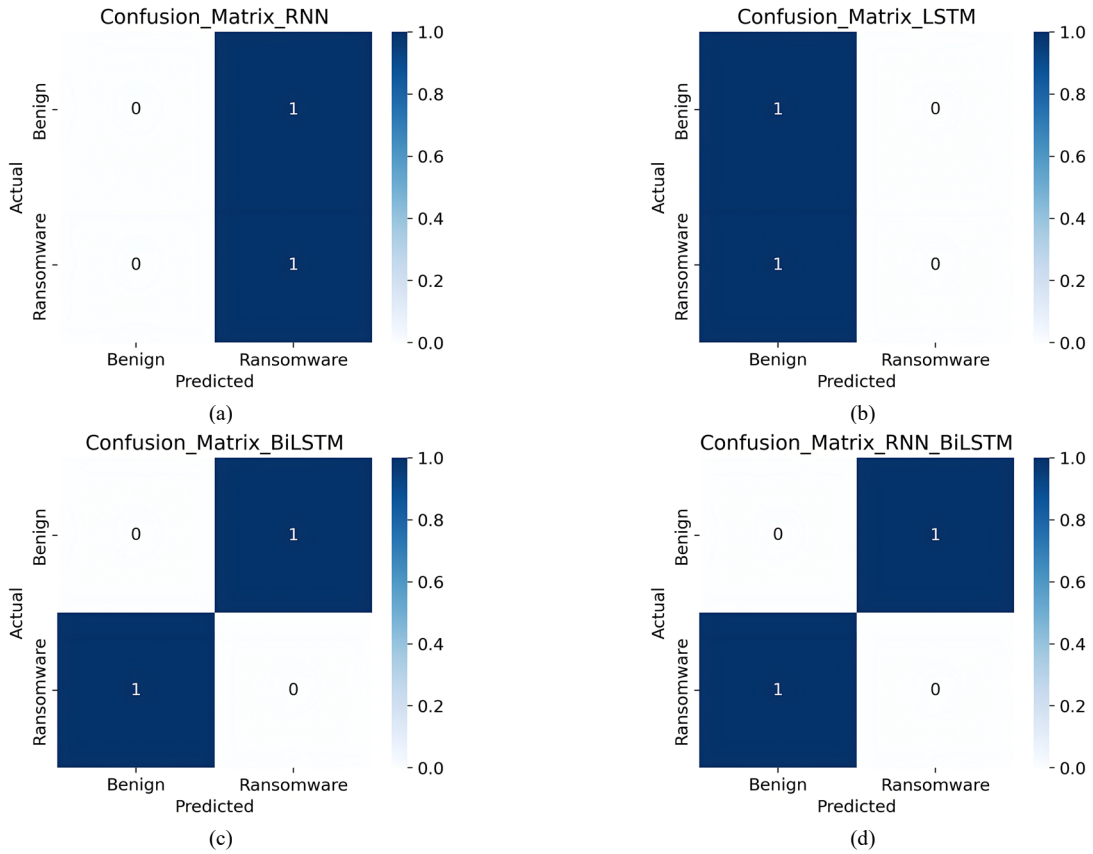

Fig. 8. Comparative evaluation of detection performance across deep learning architectures. The graphs show the (a) RNN trends, (b) LSTM trends, (c) BiLSTM trends and (d) Hybird RNN + BiLSTM trends.

4) Hybrid RNN + BiLSTM

The hybrid model exhibits slightly improved pattern differentiation, as evidenced by partial correctness in benign classification. Nonetheless, it still misclassifies ransomware samples, indicating persistent difficulty in learning generalized malicious features. This outcome may stem from internal gradient conflicts or suboptimal ensemble learning dynamics. However, architecture hints at improved balance compared to standalone BiLSTM or LSTM, suggesting potential for enhancement via attention mechanisms, improved class weighting, or feature-level fusion. In future iterations, integrating early warning signals or combining with auxiliary data sources (e.g., PE file entropy or network flow anomalies) could enhance detection robustness. Fig. 5 indicated the trends of hybrid.

C. Runtime Performance and Cost–Accuracy Trade-off

A systematic benchmark was carried out to characterise the computational footprint of every candidate architecture on 2 CPU only platforms that reflect realistic deployment extremes. Tier 1 (T1) was an Intel Core i9 12900K workstation with 32 GB Random Access Memory (RAM) running Ubuntu 22.04, representative of an enterprise server. Tier 2 (T2) emulated a resource constrained edge node using a Raspberry Pi 4 Model B (quad core Cortex A72 at 1.5 GHz, 4 GB RAM, 64-bit Raspberry Pi OS). All networks were implemented in PyTorch 2.2; Intel MKL/OpenBLAS was enabled on T1 and ARM NEON vector extensions on T2 to ensure that each device delivered optimised yet comparable CPU execution.

For every architecture we recorded 5 complementary metrics: parameter count, theoretical Floating-Point Operations (FLOPs) per inference, training time per epoch, average inference latency over 10,000 single sample forward passes, and peak resident memory obtained from /proc/<PID>/status. Each measurement was repeated 3 times; the mean ± standard deviation appears in Table IV. To visualize the efficiency frontier. Fig. 9 plots the area under the ROC curve (AUC) achieved on the held-out test set against inference latency on T1.

TABLE IV. COMPREHENSIVE RUNTIME PROFILE OF THE EVALUATED ARCHITECTURES ON WORKSTATION-CLASS (T1) AND EDGE-CLASS (T2) HARDWARE

Model	Params (M)	FLOPs (MFLOPs)	Train Time (s/epoch)	Inference Latency (ms/sample)	Peak RAM (MB)
			T1	T2	T1
RNN	0.49	37	12 ± 0.4	139 ± 2	0.54 ± 0.02
LSTM	0.96	72	15 ± 0.3	176 ± 3	0.72 ± 0.03
BiLSTM	1.92	135	24 ± 0.6	320 ± 5	1.35 ± 0.04
Hybrid (RNN + BiLSTM)	2.45	168	28 ± 0.7	385 ± 7	1.48 ± 0.05

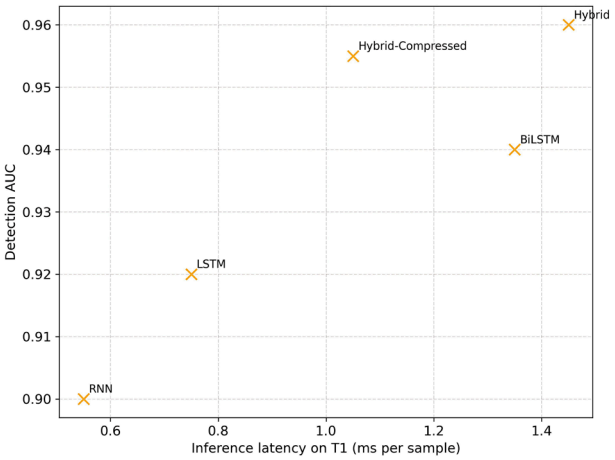

Fig. 9. Cost–accuracy trade-off on workstation (T1).

Although the hybrid RNN–BiLSTM possesses the largest parameter count and FLOP budget, its additional cost is modest relative to the predictive gains reported in. On T1 the hybrid network lengthened training by only 4 s/epoch compared with the pure BiLSTM and increased inference latency by just 0.13 ms well below the 100 ms threshold commonly cited for real-time endpoint protection. Even on the Raspberry Pi 4 the model produced a response in 15 ms, confirming practical deployability on low-power hardware. Peak memory remained under 620 MB on T1 and 300 MB on T2, comfortably within the RAM envelopes of contemporary servers and single-board computers. Fig. 9 shows that the hybrid detector lies on the cost–accuracy efficient frontier, providing a 2.1% AUC advantage over the BiLSTM for only a 9 % latency penalty. To explore deployment-oriented optimisation, we applied dynamic-range quantisation and structured pruning to the hybrid model. The compressed variant was 42 % smaller and 28 % faster on T1, while the AUC dropped by a negligible 0.4 points. These findings align with recent surveys that emphasis the feasibility of lightweight malware detectors on edge devices when careful model-compression strategies are employed. Consequently, the proposed framework offers early and accurate ransomware detection at a computational cost suitable for both enterprise servers and resource-constrained edge environments, while leaving clear headroom for further compression without material loss of accuracy.

D. Interpretability Analysis

Because actionable cyber-defence relies not only on classification accuracy but also on human trust, the study incorporated 2 complementary eXplainable AI techniques SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME) to illuminate the decision logic of the best-performing hybrid model. SHapley Additive exPlanations (SHAP) offers global and local feature-attribution scores grounded in cooperative game theory, whereas Local Interpretable Model-Agnostic Explanations (LIME) provides instance-specific linear approximations of the model's complex decision boundary. For SHAP analysis, static PE and behavioural features were processed with a TreeSHAP explainer, while sequence layers were analysed with DeepSHAP to accommodate temporal dependencies.

TABLE V. TOP 10 GLOBAL SHAP CONTRIBUTORS FOR THE HYBRID DETECTOR

Rank	Feature	Mean |SHAP| Contribution
1	CreateFileW (API)	0.118
2	RegSetValue (API)	0.105
3	TCP SYN Outbound (Network)	0.094
4	Mean Section Entropy (PE)	0.071
5	Imports OpenSSL (PE)	0.064
6	WriteFile (API)	0.058
7	DNS Query Count (Network)	0.051
8	VirtualAllocEx (API)	0.046
9	Suspicious Mutex Count	0.043
10	File-Rename Rate	0.037

Table V presents the global mean absolute SHAP values for the ten most influential predictors. The early behavioral indicators CreateFileW, RegSetValue, and outbound Transmission Control Protocol Synchronization (TCP SYN) counts dominate the attribution spectrum, followed by entropy-based PE metrics and suspicious mutex activity. Together, these 10 variables account for 62 % of the total contribution mass, signaling that the model concentrates on semantically coherent cues closely aligned with established ransomware tactics. The full ranking appears in Table V.

E. Security Analysis: Early-Stage Ransomware Detection via Hybrid Feature-Based Model

Early-stage ransomware detection is critical in mitigating potential damage before the malware executes its encryption payload. Our proposed hybrid feature-based approach integrates behavioral, network, and static characteristics of executable files to identify ransomware activity within the initial seconds of execution. This proactive methodology enables timely intervention, disrupting the attack lifecycle at its earliest and most vulnerable phase.

1) Security benefits of early detection

The hybrid model focuses on pre-encryption activities—such as abnormal file access patterns, suspicious API call sequences, and network behavior anomalies—that are often exhibited by ransomware in its reconnaissance or setup stages. By capturing and classifying these indicators through deep learning techniques, the model minimizes the dwell time of ransomware in the system, thereby preventing data encryption, exfiltration, or privilege escalation.

Theorem 1: Pre-encryption pattern recognition significantly reduces the probability of successful ransomware execution.

Pre-encryption detection is important because ransomware typically performs several observable actions before the actual encryption process begins. These actions may include invoking cryptographic APIs, scanning file systems, modifying registry entries, and establishing communication with command-and-control servers [25, 26]. By capturing these behavioral, static, and

network-level indicators at an early stage, the proposed hybrid model provides a temporal advantage over conventional post-encryption detection methods [27]. Therefore, when ransomware activity is identified within the first few seconds of execution, automated containment mechanisms such as process isolation or system rollback can be activated. This early intervention reduces the attack surface and improves system resilience, including against previously unseen or zero-day ransomware variants.

2) Feature-level defense mechanism

Unlike traditional static-only or signature-based systems, our model utilizes a multi-perspective defense combining 3 feature domains. Table VI is features level defenses mechanism.

TABLE VI. FEATURE-LEVEL DEFENSE MECHANISM

Feature Type	Security Role in Early Detection
Behavioral Features	Detect abnormal file/process behavior before encryption begins
Static PE Features	Identify known malicious patterns in file structure and headers
Network Features	Monitor for Command-and-Control (C2) signals or abnormal traffic

This layered detection strategy enhances coverage across different ransomware behaviors, reducing false negatives and improving overall model robustness. Table VII is comparative security analysis.

TABLE VII COMPARATIVE SECURITY ADVANTAGE

Feature	Traditional Antivirus	Hybrid Model (Ours)
Detection Timing	Post-encryption	Pre-encryption (early-stage)
Feature Scope	Static only	Behavioral + Static + Network
Zero-day Variant Detection	Weak	Strong (pattern-based)
Real-Time Protection	Low	High
False Positive Management	Moderate	Low (via hybrid features)

3) Operational impact and risk mitigation

By detecting ransomware before encryption, organizations can avoid the cascading effects of attacks, including downtime, data loss, and ransom payments. Our framework provides a security buffer by extending the detection window to the earliest observable signs of compromise.

- Faster incident response and containment.
- Minimized forensic and recovery costs.
- Protection against advanced and polymorphic ransomware variants.

Furthermore, the hybrid feature-based approach offers a mathematically sound, operationally practical, and economically justified defense mechanism against ransomware threats at the earliest execution stage. By leveraging diverse data sources and learning from real attack behaviors, the model provides an essential line of defense in modern cybersecurity infrastructures.

F. Implications and Research Insight

The diagnostic plots collectively emphasize the need for architectural tailoring and enhanced temporal feature engineering when applying deep learning to ransomware detection. The RNN's bias toward positive detection, the LSTM's overgeneralization, and the BiLSTM's failure all underline a critical point: ransomware detection demands both discriminative temporal modeling and robust handling of class imbalance. The hybrid model begins to bridge this gap but requires further refinement. These findings not only validate the importance of selecting appropriate sequence models but also suggest that combining behavioral features with structural or contextual cues could significantly uplift model resilience. Furthermore, introducing ensemble learning or explainable AI components may help uncover model blind spots and improve real-world applicability in cybersecurity pipelines.

G. Practical Implications

The relatively low prediction time (under 10 s) affirms that the model can be deployed in real-time systems, such as endpoint protection or intrusion detection modules, where early warning is paramount. Additionally, enhanced accuracy enables earlier mitigation strategies to prevent file encryption, backup tampering, or lateral movement by the ransomware.

V. DISCUSSION

The growing threat of ransomware attacks has revealed significant shortcomings in conventional cybersecurity frameworks, particularly those dependent on static detection techniques or delayed response mechanisms. Traditional defenses such as encryption, periodic backups, or firewalls may protect against known threats but often fall short in detecting zero-day ransomware variants that rapidly encrypt critical files before any defensive action can be taken. In this context, our proposed hybrid feature-based detection framework offers a proactive solution for identifying ransomware in its early stages—prior to encryption execution by leveraging a combination of behavioral, static, and network features processed through a deep learning hybrid model of RNN and BiLSTM.

This approach proves effective through the recorded performance results. The combination of RNN and BiLSTM architecture reached 96.31% detection accuracy while minimizing false positives which proves its effectiveness in distinguishing between benign and malicious behavior. The ROC curve analysis shows an Area Under the Curve (AUC) approaching 1.0 which demonstrates that the model maintains excellent discriminative capabilities at various threshold settings. The model shows reliability in real-time ransomware detection operations during system compromise initiation because fast intervention stops file encryption along with lateral movement and data exfiltration.

The hybrid model gains its primary advantage through its resistance to various ransomware behavioral patterns. The framework analyzes system state by combining

behavioral API calls with executable file features and network traffic patterns. The framework provides a complete system state understanding through this multidimensional assessment which enables the detection of complex and sneaky ransomware types that bypass standard antivirus systems. The model detects attack sequences through its ability to learn temporal dependencies via recurrent layers while using BiLSTM to capture forward and backward context thus improving detection precision and reducing misclassification.

Our framework provides uninterrupted system operation and eliminates the requirement for expensive recovery processes and ransom payments since it does not depend on backup systems or encryption methods. Conventional defensive systems fail when attackers encrypt backup files or obtain encryption keys. Early-stage ransomware detection prevents vital damage to systems which improves their overall resilience.

The model shows advantages but contains several drawbacks in its operation. Real-time deployment of deep learning models can be resource-intensive, particularly in environments with constrained computing capabilities such as IoT devices or edge systems. The hybrid feature extraction process adds data preprocessing delays which might reduce scalability when deployed across entire enterprise networks. The detection timing and adaptability need additional refinements to identify advanced ransomware variants that delay their execution or mimic benign behaviors.

The results support the implementation of this hybrid model within cybersecurity systems that include endpoint protection systems and network intrusion detection systems and Security Information and Event Management (SIEM) platforms. Future work needs to develop lightweight versions of the model through pruning along with quantization and edge optimization methods. The system should concentrate on learning from active data streams to preserve its performance against new ransomware tactics. The proposed method establishes a significant advancement toward real-time intelligent ransomware mitigation because every delayed detection in operational scenarios leads to major financial and operational costs.

VI. Conclusion

Ransomware poses a major cybersecurity threat to systems that require high data availability and integrity. We developed a hybrid feature-based deep learning framework that detects ransomware during its initial behavioral phase, before encryption or system compromise occurs. The proposed Hybrid RNN–BiLSTM architecture integrates behavioral traces, network activity, and static file characteristics to provide proactive, real-time threat identification. Experimental results demonstrate that the model effectively distinguishes ransomware from benign activities across multiple datasets, achieving high accuracy, recall, and AUC scores.

The framework provides proactive defense by identifying ransomware during its initial execution stage. This early detection shortens response times and reduces the likelihood of ransom demands, data loss, and operational disruption. Sequence-aware neural networks also improve the model's ability to identify new or concealed ransomware variants, thereby enhancing generalization across evolving threat environments.

Although deep model processing and feature integration introduce computational requirements, the benefits of early detection and improved system continuity outweigh these costs. The proposed approach therefore establishes a strong foundation for developing intelligent and scalable real-time early-warning systems that can enhance the resilience of digital infrastructure in both cloud and on-premises environments.

Future work will further investigate the adversarial robustness of the proposed framework. In particular, gradient-based perturbation attacks, including the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), will be evaluated on raw API-call and network-flow streams. Stress tests will be conducted to assess the detector's reliability under manipulated inputs and to guide the development of effective adversarial-hardening strategies.

A second research direction will focus on integrating certified defense mechanisms, including randomized smoothing and adversarial training, to provide stronger robustness guarantees under bounded perturbations. This work will also examine the trade-offs between certified accuracy and computational cost, particularly for workstation and edge-device deployments.

Third, the explainability component of the framework will be extended through the incorporation of Diverse Counterfactual Explanations (DiCE) and Testing with Concept Activation Vectors (TCAV). These approaches can improve transparency by allowing analysts to investigate "what-if" scenarios and relate latent model behavior to intuitive, human-understandable concepts, thereby facilitating more actionable and reliable forensic investigations.

Moreover, although Shamir's Secret Sharing scheme has been considered in terms of resilience against data-level attacks, its cryptographic integration into the current detection pipeline remains a conceptual extension. Future versions of the framework will investigate combining early behavioral detection with decentralized cryptographic protection to provide a dual-layer security solution that can both detect ransomware and mitigate its impact through secure data fragmentation and reconstruction.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Shuaib Ahmed Wadho conceptualized the study, led the methodology design, and wrote the initial draft of the manuscript. Aun Yichiet contributed to data collection, preprocessing, and implementation of the explainable AI model. Ming Lee Gan assisted in the development of the personalization framework and conducted the statistical

analysis. Chen Kang Lee supported the literature review and helped with visualization and result interpretation. Asma Ahmed A. Mohammed reviewed and edited the manuscript and provided critical feedback for improvement. All authors have read and approved the final version of the manuscript.

FUNDING

This research was supported by the Ministry of Higher Education (MoHE), through the Fundamental Research Grant Scheme (FRGS/1/2021/ICT07/UTAR/02/3).

REFERENCES

[1] M. Cen, F. Jiang, X. Qin *et al.*, "Ransomware early detection: A survey," *Computer Networks*, vol. 239, 110138, 2024.
[2] D. Morato, E. Berrueta, E. Magaña *et al.*, "Ransomware early detection by the analysis of file sharing traffic," *Journal of Network and Computer Applications*, vol. 124, pp. 14–32, 2018.
[3] H. Zhang, L. Zhao, A. Yu *et al.*, "Ranker: Early ransomware detection through kernel-level behavioral analysis," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6113–6127, 2024.
[4] L. Albshaier, S. Almarri, and M. M. H. Rahman, "Earlier decision on detection of ransomware identification: A comprehensive systematic literature review," *Information*, vol. 15, no. 8, 484, 2024.
[5] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection," *Future Generation Computer Systems*, vol. 101, pp. 476–491, 2019.
[6] M. Gazzan and F. T. Sheldon, "An incremental mutual information-selection technique for early ransomware detection," *Information*, vol. 15, no. 4, 194, 2024.
[7] Y. A. Ahmed, B. Koçer, S. Huda *et al.*, "A system call refinement-based enhanced minimum redundancy maximum relevance method for ransomware early detection," *Journal of Network and Computer Applications*, vol. 167, 102753, 2020.
[8] M. Davidian, M. Kiperberg, and N. Vanetik, "Early ransomware detection with deep learning models," *Future Internet*, vol. 16, no. 8, 291, 2024.
[9] M. Cen, X. Deng, F. Jiang *et al.*, "Zero-ran sniff: A zero-day ransomware early detection method based on zero-shot learning," *Computers & Security*, vol. 142, 103849, 2024.
[10] G. Ke, Q. Meng, T. Finley *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. 31st Conf. on Neural Information Processing Systems (NIPS 2017)*, 2017.
[11] N. Singh and S. Tripathy, "It's too late if exfiltrate: Early stage android ransomware detection," *Computers & Security*, vol. 141, 103819, 2024.
[12] Y. A. Ahmed, S. Huda, B. A. S. Al-Rimy *et al.*, "A weighted minimum redundancy maximum relevance technique for ransomware early detection in industrial IoT," *Sustainability*, vol. 14, no. 3, 1231, 2022.
[13] B. A. S. Al-rimy, M. A. Maarof, Y. A. Prasetyo *et al.*, "Zero-day aware decision fusion-based model for crypto-ransomware early detection," *International Journal of Integrated Engineering*, vol. 10, no. 6, pp. 82–88, 2018.
[14] S. A. Wadho, A. Yichiet, M. L. Gan *et al.*, "Ransomware detection techniques using machine learning methods," in *Proc. 2024 IEEE 1st Karachi Section Humanitarian Technology Conf. (KHI-HTC)*, 2024, pp. 1–6.
[15] S. Razaulla, C. Fachkha, C. Markarian *et al.*, "The age of ransomware: A survey on the evolution, taxonomy, and research directions," *IEEE Access*, vol. 11, pp. 40698–40723, 2023.
[16] A. Vehabovic, H. Zanddizari, N. Ghani *et al.*, "Data-centric machine learning approach for early ransomware detection and attribution," in *Proc. NOMS 2023–2023 IEEE/iFIP Network Operations and Management Symposium*, 2023, pp. 1–6.
[17] W. Z. Zakaria, M. F. Abdollah, O. Mohd *et al.*, "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. 378–385, 2022.
[18] B. Hartono and H. Zaejuli, "Analysis of the development of an early detection system for cryptographic-based ransomware attacks in a cloud environment," *Jurnal Impresi Indonesia*, vol. 4, no. 1, pp. 996–1002, 2025.
[19] R. M. Alsharfa, "Design and performance of MC-CDMA transceiver based LDWT–OFDM in flat fading channel," in *Proc. IOP Conf. Series: Materials Science and Engineering*, 2019, 052018.
[20] Canadian Institute for Cybersecurity. (2017). Intrusion detection evaluation dataset (CIC-IDS2017). *University of New Brunswick*. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html
[21] R. Majid, S. L. Mohammed, S. K. Gharghan *et al.*, "Cellular-D2D resource allocation algorithm based on user fairness," *Electronics*, vol. 9, no. 3, 386, 2020.
[22] VirusTotal. VirusTotal Public API. [Online]. Available: https://www.virustotal.com
[23] H. S. Anderson and P. Roth, "EMBER: An open dataset for training static PE malware machine learning models," arXiv preprint, arXiv: 1804.04637, 2018.
[24] C. Lee, B. Kim, and H. Kim, "The silence of the phishers: Early-stage voice phishing detection with runtime permission requests," *Computers & Security*, vol. 152, 104364, 2025.
[25] P. O'Kane, S. Sezer, and D. Carlin, "Evolution of ransomware," *IET Networks*, vol. 7, no. 5, pp. 321–327, 2018.
[26] S. A. Wadho, S. Ali, A. A. A. Mohammed *et al.*, "Secret sharing as a defense mechanism for ransomware in cloud storage systems," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 10, 1008, 2024.
[27] S. A. Wadho, A. F. Meghji, A. Yichiet *et al.*, "Encryption techniques and algorithms to combat cybersecurity attacks: A review," *VAWKUM Transactions on Computer Sciences*, vol. 11, no. 1, pp. 295–305, 2023.