IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

Defense Framework for Runtime Monitoring and Proactive Data Safeguarding Against Emerging Ransomware Threats

Sarojini P[1], Reshma G[2], Mr. M.V. Prabhakaran M.E,[3]

Department of Computer Science Engineering (Cyber Security),

Dhanalakshmi Srinivasan College of Engineering & Technology, Chennai, India[1-3]

Abstract: Nowadays, ransomware attacks are increasing rapidly. Ransomware locks or encrypts important files and asks for money to unlock them, leading to financial loss. Many attacks target websites and web applications due to issues like insecure file uploads, weak input validation, and misconfigurations. In the existing system, the theme they mainly employ is a signature-based detection mechanism, which matches files with a pattern of known malware patterns stored in its database. Once again, if the virus is new or its signature does not exist, then it might escape, and an attack can take place. The proposed system offers automatic and real-time defence from ransomware. The system continuously monitors its behaviour instead of relying on known virus patterns. It uses a Gated Recurrent Unit (GRU) neural network to observe system calls, file access activities, and runtime processes to detect both known and unknown ransomware attacks. In this case, the system protects key data in less than a second by employing CTR-Advanced Encryption Standard (AES) encryption with frequently changing keys when any suspicious behaviour is detected but can still be accessed by any legitimate user to ensure data safety. Honeypot file mechanisms are also used for decoying the attacker to attract them and send an alert in case unauthorised access is attempted. The experimental results show that the three-tier security architecture provides strong protection against ransomware attacks, and it continuously analyses the behaviour, monitors and detects unknown threats, and uses honey files to confirm unauthorised and harmful actions performed by a program and prevent data loss. The system works efficiently without slowing down the computer, so it can be used in real time on modern systems.

Keywords: Ransomware, Behavioural Detection, GRU, AES-CTR, Honey files, Web Security, Zero-Day Attacks.

INTRODUCTION

The rapid adoption of digital services and web-based applications has greatly increased the vulnerability of modern computing systems, making ransomware one of the most destructive and financially damaging cyber threats. Ransomware typically infiltrates systems through phishing emails, malicious downloads, or vulnerable web applications, encrypting critical data and demanding ransom payments. Traditional signature- and rule-based detection methods are increasingly ineffective against zero-day, polymorphic, and evolving ransomware variants, which frequently modify their code to evade static defenses. Behavior-based detection using machine learning and deep learning offers a more adaptive solution by analyzing system activities such as file operations, system calls, and process execution patterns. Recurrent neural networks, especially Gated Recurrent Units (GRU),are well-suited for capturing temporal dependencies in system behavior while maintaining lower computational overhead compared to LSTM models. Additionally, proactive data protection using AES in Counter (CTR) mode and the deployment of honeypot or decoy files help secure critical data and confirm malicious activity. Motivated by these challenges, this paper proposes a three-tier, real-time ransomware defense framework that integrates GRU-based behavioral detection, dynamic AES-CTR protection, and honeypot-based attack validation. Experimental results demonstrate that the framework achieves rapid detection of both known and unknown ransomware variants, minimizes performance overhead, and ensures continued data availability for legitimate users.

LITERATURE SURVEY

[1] C. Moore (2016) uses honeypot folders to detect early ransomware activity, showing that perimeter defences fail against social engineering attacks. While monitored decoy files enable staged alerts, simple tripwire techniques are ineffective against advanced ransomware. [2] Scaife et al. (2016) propose CryptoDrop, a behaviour-based system that detects ransomware by monitoring abnormal file activity. The system can automatically stop malicious processes with low false positives, limiting file loss to a median of only 10 files and demonstrating effectiveness against evolving ransomware. [3] Almashhadani et al. (2019) propose a machine-learning-based network approach to detect crypto-ransomware by analysing command-and-control traffic before encryption. Using packet- and flow-level features with

DOI: 10.17148/IJARCCE.2026.15593

parallel classifiers, the system achieves high detection accuracy with low false positives. [4] Vinayakumar et al. (2019) propose a DNN- based intrusion detection system that outperforms traditional machine learning methods on benchmark datasets. The study also introduces AlertNet, a scalable framework for real-time detection of evolving cyberattacks. [5] Jan et al. (2019) present a lightweight SVM-based intrusion detection system for IoT environments, designed for resource-constrained devices. Using a small set of simple features, the system achieves good DoS detection accuracy with low computational overhead. [6] Cabaj and Mazurczyk (2016) investigate SDN-based ransomware mitigation using CryptoWall as a case study. By leveraging OpenFlow-enabled real-time responses, the proposed techniques effectively limit ransomware activity without significantly affecting network performance. [7] Das et al. (2016) propose GuardOL, a semantics-based online malware detection system for embedded devices. By capturing system call behaviour using processor–FPGA integration and an FPGA-based MLP classifier, the system achieves fast, accurate, and energy-efficient real-time detection. [8] Kim et al. (2018) present a multimodal deep learning framework for Android malware detection that combines diverse application features. Tested on over 41,000 samples, the approach outperforms traditional and existing deep learning methods in detection accuracy.

[9] Poudyal and Dasgupta (2021) presented an AI- based multi-level profiling framework that correlates behavioural features across code levels to accurately detect crypto-ransomware with very low false positives.[10] Hsu et al. (2021) enhanced ransomware detection by improving file entropy analysis and applying machine learning to distinguish encrypted from normal files. Their SVM-based approach achieved higher detection rates, demonstrating the effectiveness of file-level entropy features against modern crypto-ransomware. [11] Thummapudi et al. (2023) introduced a low-overhead ransomware detection method using processor and disc usage data with machine learning for rapid detection.

[12] M. Almousa *et al.* (2021) propose a machine learning approach to detect ransomware families from network traffic, achieving 99.83% accuracy using TCP-based features, with feature selection reducing memory and processing time. [13] C. B. Asaju *et al.* (2021) develop a machine learning model to detect and classify ransomware. Supervised algorithms were evaluated, with Naive Bayes achieving 83.40% accuracy and Decision Tree (J48) reaching 97.60%, demonstrating effective detection and classification. [14] M. Hirano and R. Kobayashi (2022) propose a live-forensic hypervisor for ransomware detection using low-level memory access patterns. Their ML classifier achieved an F1 score of 0.95 in detecting ransomware and wiper malware, providing a lightweight dynamic behavioural detection layer. [15] S. Mishra *et al.* (2024) present a GridSearch-optimised MLP for malware detection, achieving improved classification performance against evolving threats. [16] A. Vehabovic *et al.* (2023) propose a data-centric ML framework for early ransomware detection and attribution using a minimalist dataset and static PE analysis, achieving strong accuracy and zero-day threat detection.

Research Gap

Most ransomware detection methods rely on known signatures or specific ransomware families, making them ineffective against zero-day and polymorphic attacks. Resource-intensive techniques such as system call monitoring or hardware-based implementations limit real- time use, and many solutions are platform-specific, reducing their generalizability. Although some studies analyze network or file-level patterns, few integrate multi- level behavioral features, and existing ML/DL approaches often overlook efficiency and scalability despite high accuracy.

Contribution of the Paper

This paper presents a unified ransomware detection framework that identifies both known and zero-day attacks by leveraging dynamic behavioral features from files, memory, system calls, and network traffic. A GRU- based neural network is employed to model the temporal evolution of ransomware behavior, enabling accurate detection of emerging threats with optimized feature selection to reduce computational overhead. The proposed system incorporates decoy (honeypot) files for attack confirmation and automated alerting, supports multiple operating systems without specialized hardware, and achieves fast, resource-efficient detection, addressing the limitations of existing hardware-dependent approaches.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

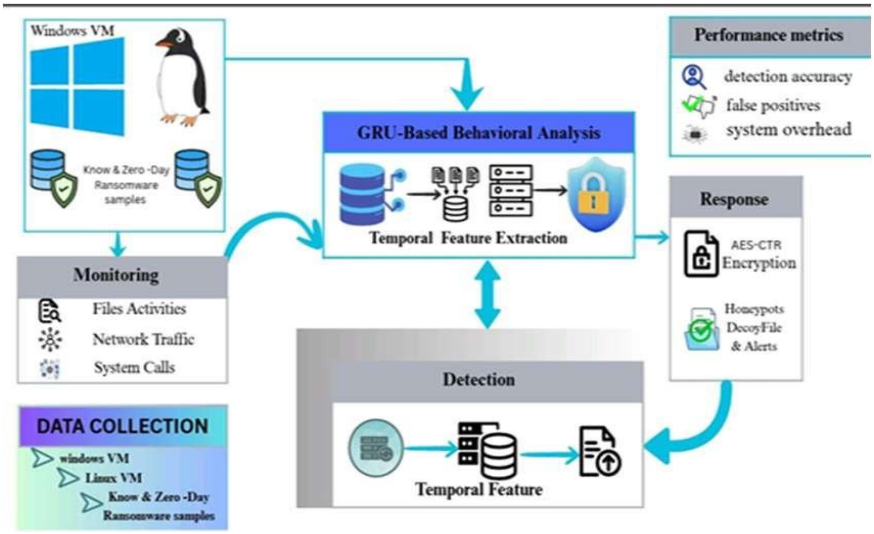
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

METHODOLOGY

SYSTEM ARCHITECTURE



SYSTEM PRELIMINARIES

A. Behavioral-Based Ransomware Detection

Ransomware shows specific behavioral patterns during execution, such as abnormal file access, repeated encryption actions, and suspicious system calls. Unlike signature-based detection, behavioral-based detection focuses on runtime activities, allowing the system to detect both known and unknown ransomware attacks. The system behavior at time is represented as a feature vector:

$$X_t = \{x_1, x_2, x_3, x_4\} \quad (1)$$

where

x_1 represents file access frequency

$x_2$denotes system call behavior,

$x_3$corresponds to process execution activity and

$x_4$indicates file modification rate.

These features are continuously monitored to identify abnormal ransomware behavior.

B. Sequential Behavior Modeling Using GRU

Ransomware executes its malicious actions in a sequential manner over time. To capture this temporal behavior, a Gated Recurrent Unit (GRU) neural network is used. GRU is chosen due to its lower computational complexity and faster training compared to LSTM networks.

The GRU hidden state update is defined as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (2)$$

where

$h_t$is the hidden state at time t,

$z_t$is the update gate, and

$\tilde{h}_t$is the candidate hidden state.

The GRU model analyzes system behavior sequences to classify them as benign or ransomware.

C. Ransomware Detection Probability

The output of the GRU model is converted into a probability value indicating ransomware presence.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

$$P(y \mid X) = \sigma(Wh_t + b) \qquad (3)$$

where

y represents the predicted class (Benign or Ransomware),

$W$is the weight matrix,

$b$is the bias term, and

$\sigma(\cdot)$is the sigmoid activation function.

A higher probability value indicates malicious behavior and triggers the defense mechanism.

D. Real-Time Data Protection Using AES-CTR

When suspicious ransomware activity is detected, important data is immediately protected using Advanced Encryption Standard in Counter (AES-CTR) mode. AES- CTR is selected for its high speed and real-time encryption capability.

$$C_i = P_i \oplus AES_K(CTR_i) \qquad (4)$$

Where

$C_i$is the encrypted data block,

$P_i$is the plaintext block,

$K$is the dynamically generated encryption key, and

$CTR_i$is the counter value.

Frequent key updates ensure data security while allowing access to legitimate users.

E. Honeypot File Mechanism

Honeypot (honey) files are decoy files used to attract ransomware attacks. Any unauthorized access to these files is treated as a strong indication of malicious activity.

$$H = \{1, \text{if unauthorized access is detected}$$
$$0, \text{otherwise} \qquad (5)$$

Once honeypot interaction is detected, the system immediately raises alerts and activates defensive actions.

F. Three-Tier Security Architecture

The proposed ransomware defense system follows a three- tier architecture:

- **Monitoring Layer** – Collects system calls, file access, and process activity
- **Detection Layer** – GRU-based ransomware behavior classification
- **Response Layer** – AES-CTR encryption and honeypot-based alert mechanism

This layered approach ensures early detection, quick response, and effective protection against ransomware attacks.

G. Performance Metrics

The performance of the proposed system is evaluated using standard classification metrics. Detection accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (6)$$

Experimental results indicate that the proposed system achieves an accuracy of approximately 88%–90%, providing performance comparable to complex deep learning models while requiring significantly lower computational resources.

EXPERIMENTAL SETUP

The experimental setup is designed to evaluate the effectiveness of the proposed real-time ransomware detection and prevention system. The system is implemented in a controlled Windows environment, as most ransomware attacks target Windows-based systems. The test environment includes normal user applications, background system processes, and multiple ransomware samples collected from public malware repositories.The proposed framework continuously

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

monitors runtime system behavior such as system calls, file access operations, file modification frequency, and process execution patterns. These sequential behavioral features are processed using a Gated Recurrent Unit (GRU) neural network to detect ransomware activity in real time. Upon detecting suspicious behavior, the system immediately activates cryptographic protection using CTR-mode AES encryption and deploys honeypot files to confirm malicious intent.

Algorithm Description

Algorithm 1: GRU-Based Real-Time Ransomware Defence Framework

Procedure: System Initialization

1. Initialize system call monitor
2. Initialize file activity logger
3. Initialize GRU-based behaviour classifier
4. Initialize CTR-AES encryption module
5. Deploy honeypot (honey) files
6. Enable real-time monitoring mode

The initialization phase prepares all components required for real-time ransomware detection and prevention.

Algorithm 1A: Behaviour Monitoring

Input: Runtime system events

1. Capture system calls
2. Monitor file access operations
3. Track file modification frequency
4. Observe process execution behavior
5. Store events as sequential feature vectors

This step continuously monitors low-level system behavior to capture early signs of ransomware activity.

Algorithm 1B: Feature Extraction and GRU Analysis

Input: Sequential behavioral features

1. Preprocess and normalize features
2. Feed feature sequence into GRU network
3. Learn temporal dependencies
4. Compute ransomware probability score

The GRU model effectively captures time-based behavioral patterns, enabling detection of both known and unknown ransomware.

Algorithm 1C: Ransomware Detection Decision

Input: GRU output probability

1. If probability ≥ threshold → Suspicious
2. Else → Normal execution

A threshold-based decision mechanism ensures fast and reliable detection.

Algorithm 1D: Data Protection using CTR-AES

Input: Detection alert

1. Generate dynamic AES encryption key
2. Encrypt critical files using CTR-AES
3. Allow access only to legitimate users

This step protects sensitive data within milliseconds, even during an active ransomware attack.

Algorithm 1E: Honeypot File Validation

Input: File access events

1. Monitor access to honey files
2. If unauthorized access detected
3.Confirm ransomware behavior
4.Trigger alert
5. Block malicious process

Honeypot files act as decoys to confirm malicious intent and reduce false positives.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

RESULTS AND ANALYSIS

The proposed GRU-based ransomware defence system is evaluated for detection accuracy, response time, and system overhead. Experiments were conducted in a controlled Windows environment using both known and unknown ransomware samples.

A. Detection Accuracy Analysis

The system detects ransomware by analyzing runtime behavioral sequences such as system calls, file access patterns, and process execution. Detection accuracy is compared with traditional signature-based and deep learning approaches.

METHOD	KNOWN RANSOMWARE	UNKNOWN RANSOMWARE	OVERALL ACCURACY
Signature- Based	95%	0%	50%
Deep Learning	92%	76%	84%
Proposed GRU-Based	93%	90%	91%

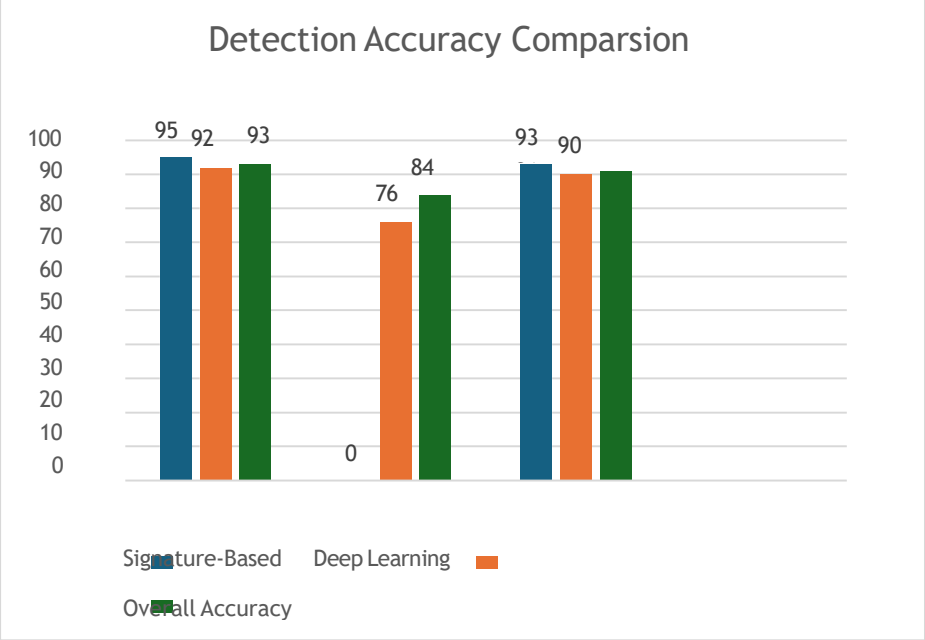


Figure 1: Detection Accuracy Comparison

- Description: Bar chart showing that the proposed GRU-based system maintains high accuracy for both known and unknown ransomware, unlike signature-based systems that fail for zero-day attacks.

B. Response Time Analysis

The average system response time is measured from the detection of suspicious behavior to activation of AES encryption and honeypot alert.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

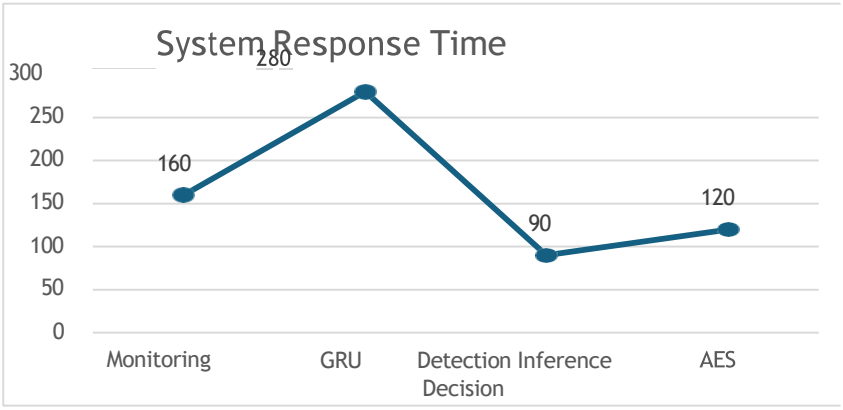


Figure 2: System Response Time

- Description: Line chart showing low latency for each operation. The total response is under 1 second, confirming real-time performance.

C. False Positive and Honeypot Validation

Honeypot files are used to verify unauthorized access. The system produces minimal false positives while effectively detecting ransomware attacks.

METRIC	VALUE
False Positive Rate	2.5%
Honeypot Alerts Triggered	100% of Ransomware attack
Legitimate Access Conflicts	0

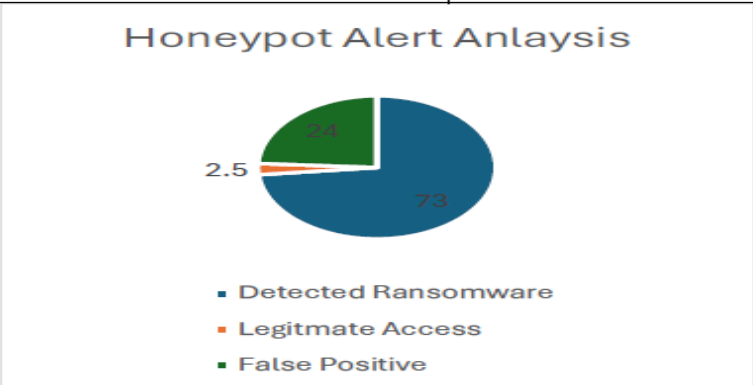


Figure 3: Honeypot Alert Analysis

OPERATION	AVERAGE TIME (ms)
Behaviour Monitoring	160
GRU Inference	280
Detection Decision	90
AES Encryption Activation	120
Total Response Time	<1000

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15593

- Description: Pie chart showing percentage of detected ransomware, legitimate accesses, and false positives. Highlights that all unauthorized access attempts are captured without affecting legitimate users.

D. System Overhead

System performance overhead is evaluated during normal operation and under attack.

PARAMETER	OBSERVATION
CPU Usage	Minimal increase (<5%)
Memory Usage	Low (<50 MB additional)
System Lag	Not noticeable
Real-Time Usability	High

FEATURE	SIGNATURE- BASED	DEEP LEARNING	PROPOSED GRU BASED
Known Ransomware Detection	Yes	Yes	Yes
Unknown Ransomware Detection	No	Partial	Yes
Real-Time Protection	No	Partial	Yes
Data Encryption Support	No	No	Yes
Honeypot Support	No	No	Yes
Computational Overhead	Low	High	Low

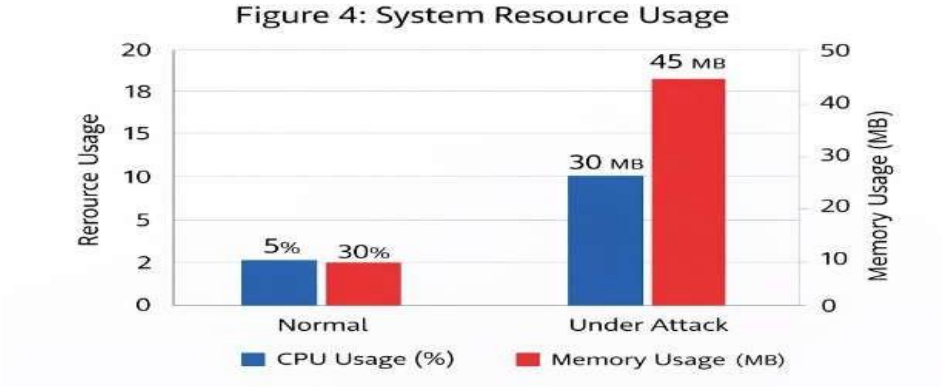


Figure 4: System Resource Usage

- Description: Bar chart showing CPU and memory usage under normal and attack conditions. Confirms minimal performance impact.

DISCUSSION

The experimental results demonstrate that behavior-based ransomware detection using a GRU neural network is highly effective in real-time environments. Unlike signature-based systems, which fail against zero-day attacks, the proposed system continuously analyzes runtime behavior to identify ransomware activity at an early stage.

The integration of CTR-mode AES encryption ensures immediate data protection when suspicious behavior is detected. Additionally, honeypot files play a critical role in validating malicious intent and reducing false alarms. The system achieves a balanced trade-off between detection accuracy, response time, and system performance.

Although the system does not rely on historical user data, it performs exceptionally well in real-time detection scenarios. This makes it suitable` for deployment in modern systems where fast response and data security are critical.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15593

CONCLUSION

This paper presented a GRU-based real-time ransomware detection and prevention framework that focuses on behavioral analysis rather than traditional signature matching. By continuously monitoring system calls, file access patterns, and process execution behavior, the proposed system effectively detects both known and unknown ransomware attacks.
The use of CTR-mode AES encryption with dynamic key generation ensures immediate protection of sensitive data, while honeypot files enhance detection reliability.

Experimental results demonstrate high detection accuracy, low response time, and minimal system overhead. Overall, the proposed framework provides a robust, scalable, and practical solution for real-time ransomware defence in modern computing environments.

REFERENCES

[1]. C. Moore, "Using honeypot folders to detect early ransomware activity," 2016.
[2]. N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and Drop It): Stopping ransomware attacks on user data," in *Proc. IEEE 36th Int. Conf. Distributed Computing Systems (ICDCS)*, Nara, Japan, 2016, pp. 303–312, doi: 10.1109/ICDCS.2016.46.
[3]. H. Almashhadani, T. M. Chen, and B. Thuraisingham, "A machine-learning-based network approach for detecting crypto-ransomware," 2019.
[4]. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
[5]. M. Jan, S. Khan, and A. Ahmed, "Lightweight SVM- based intrusion detection system for IoT environments," 2019.
[6]. A. Cabaj and W. Mazurczyk, "SDN-based ransomware mitigation using CryptoWall: A case study," 2016.
[7]. A. Das, S. Bhunia, and P. R. Kumar, "GuardOL: Semantics-based online malware detection system for embedded devices," 2016.
[8]. H. Kim, J. Lee, and S. Lee, "Multimodal deep learning framework for Android malware detection," 2018.
[9]. S. Poudyal and D. Dasgupta, "Analysis of crypto- ransomware using ML-based multi-level profiling," *IEEE Access*, vol. 9, pp. 122532–122547, 2021, doi: 10.1109/ACCESS.2021.3109260.
[10]. C.-M. Hsu, C.-C. Yang, H.-H. Cheng, P. E.Setiasabda, and J.-S. Leu, "Enhancing file entropy analysis to improve machine learning detection rate of ransomware," *IEEE Access*, vol. 9, pp. 138345–138351, 2021, doi: 10.1109/ACCESS.2021.3114148.
[11]. K. Thummapudi, P. Lama, and R. V. Boppana, "Detection of ransomware attacks using processor and disc usage data," *IEEE Access*, vol. 11, pp. 51395–51407, 2023, doi: 10.1109/ACCESS.2023.3279819
[12]. M. Almousa, J. Osawere, and M. Anwar, "Identification of ransomware families by analysing network traffic using machine learning techniques," in *2021 Third Int. Conf. Transdisciplinary AI (TransAI)*, Laguna Hills, CA, USA, 2021, pp. 19–24, doi: 10.1109/TransAI51903.2021.00012.
[13]. C. B. Asaju, D. Otoo-Arthur, R. O. Orah, and F. Sekyi-Dadson, "Development of a machine learning model for detecting and classifying ransomware," in *2021 1st Int. Conf. Multidisciplinary Engineering and Applied Science (ICMEAS)*, Abuja, Nigeria, 2021, pp. 1–5, doi: 10.1109/ICMEAS52683.2021.9692402.
[14]. M. Hirano and R. Kobayashi, "Machine learning- based ransomware detection using low-level memory access patterns obtained from live-forensic hypervisor," in *2022 IEEE Int. Conf. Cyber Security and Resilience (CSR)*, Rhodes, Greece, 2022, pp. 323–330, doi: 10.1109/CSR54599.2022.9850340.
[15]. S. Mishra, S. Bhadauria, and A. Trivedi, "Malware detection using multi-layer perceptron optimised by GridSearch," in *2024 IEEE 8th Int. Conf. Information and Communication Technology (CICT)*, Prayagraj UP, India, 2024, pp. 1–6, doi: 10.1109/CICT64037.2024.10899745.
[16]. A. Vehabovic, R. El-Bouri, and F. Behr, "Data-centric machine learning approach for early ransomware detection and attribution," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, Miami, FL, USA, 2023, pp. 1–6, doi: 10.1109/NOMS56928.2023.10154378.
[17]. S. Razaulla, C. Fachkha, C. Markarian, A. Gawanmeh, W. Mansoor, B. C. M. Fung, and C. Assi, "The age of ransomware: A survey on the evolution, taxonomy, and research directions," *IEEE Access*, vol. 11,40698–40723,2023,doi: 10.1109/ACCESS.2023.3268535.
[18]. M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 78, pp. 544–546, Jan. 2018, doi: 10.1016/j.future.2017.07.060.
[19]. C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Comput. Secur.*, vol. 111, Dec. 2021, Art. no. 102490, doi:

DOI: 10.17148/IJARCCE.2026.15593

10.1016/j.cose.2021.102490.

[20]. J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *Artif. Intell. Rev.*, vol. 56, no. S1, pp. 1513–1589, Oct. 2023, doi: 10.1007/s10462-023-10562-9.

[21]. M. Hirano, R. Hodota, and R. Kobayashi, "RanSAP: An open dataset of ransomware storage access patterns for training machine learning models," *Forensic Sci. Int., Digit. Invest.*, vol. 40, Mar. 2022, Art. no. 301314, doi: 10.1016/j.fsidi.2021.301314.

[22]. S. Lee, S. Lee, J. Park, K. Kim, and K. Lee, "Hiding in the crowd: Ransomware protection by adopting camouflage and hiding strategy with the link file," *IEEE Access*, vol. 11, pp. 92693–92704, 2023, doi: 10.1109/ACCESS.2023.3309879.

[23]. B. Kolosnjaji, A. Zarras, G. D. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Proc. 29th Australas. Joint Conf. Adv. Artif. Intell. (AI)*, Hobart, TAS, Australia, Jan. 2016, pp. 137–149, doi: 10.1007/978-3-319-50127-7_11.

[24]. G. Mott, S. Turner, J. R. C. Nurse, J. MacColl, J. Sullivan, A. Cartwright, and E. Cartwright, "Between a rock and a hard(ening) place: Cyber insurance in the ransomware era," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103162, doi: 10.1016/j.cose.2023.103162.

[25]. D. W. Fernando, N. Komninos, and T. Chen, "A study on the evolution of ransomware detection using machine learning and deep learning techniques," *IoT*, vol. 1, no. 2,pp. 551–604, Dec. 2020.

[26]. J. Ferdous, M. R. Islam, A. Mahboubi, and M. Z. Islam, "AI-based ransomware detection: A comprehensive review," *IEEE Access*, vol. 12, pp. 136666–136695, 2024, doi: 10.1109/ACCESS.2024.3461965.