International Journal of Computer Science and Artificial Intelligence

DOI: 10.5281/zenodo.22682640 | ISSN: 3141-643X | Volume: 1 | Issue: 2 | 2026

Development of an AI-Powered Ransomware Early Warning System for Hospital Managements

Osifo Nosayaba Blessing[1], John Temitope Ogbiti[2]

[1,2]Department of Computer Science, Edo State University Iyamho, Nigeria
Corresponding Author: osifo22.nosayaba@edouniversity.edu.ng

Abstract

Ransomware poses an escalating threat to healthcare institutions, where attacks disrupt clinical workflows, compromise patient data, and directly endanger patient safety. Existing detection systems are reactive, binary in classification, and lack contextual awareness of hospital-specific risk. This paper presents the design and implementation of Leadeks REWS, an AI-powered, hospital-aware ransomware early warning system that classifies system behaviour into three stages, normal, pre-encryption, and encryption, using a Random Forest ensemble classifier trained on dynamic behavioural features. The system identifies the affected hospital subsystem (Electronic Health Records, Picture Archiving and Communication Systems, or Administrative Systems), assigns a clinical risk level, and surfaces confidence-scored alerts through a real-time Flask and MongoDB dashboard with stage-specific mitigation recommendations. A rule-based Hospital Security Monitor additionally detects brute-force, privilege-escalation, and cross-system attacks within three interactive hospital subsystems. The Random Forest model, trained on an 80/20 stratified split of a 2,000-sample dynamic ransomware dataset, achieved 91% overall classification accuracy and 87% recall for the pre-encryption stage, outperforming Logistic Regression, single Decision Tree, and Support Vector Machine baselines. Twenty-five structured test cases and a formal evaluation against eight non-functional requirements confirm that the system delivers low-latency, reliable, and clinically prioritised ransomware detection suited to resource-constrained hospital environments.

Keywords: Ransomware Detection; Machine Learning; Random Forest; Hospital Cybersecurity; Early Warning System.

1. Introduction

The rapid digitisation of healthcare, through Electronic Health Records (EHR), Picture Archiving and Communication Systems (PACS), and networked medical devices, has improved clinical efficiency while exposing hospitals to significant cybersecurity threats. Ransomware, malicious software that encrypts critical data and demands payment for its release, has emerged as one of the most damaging of these threats. The World Health Organization [15] reported that healthcare cyberattacks increased by 69% globally between 2020 and 2022, with ransomware accounting for over half of all incidents. Given the time-sensitive nature of clinical care, hospitals have become high-value targets for ransomware attackers.

In Nigeria, the Nigerian Communications Commission's Computer Security Incident Response Team [16] reported a sharp increase in ransomware attacks targeting critical infrastructure, including healthcare facilities, yet most Nigerian hospitals lack dedicated cybersecurity personnel and rely on basic antivirus software that is ineffective against modern, polymorphic ransomware variants [1]. Traditional signature-based and rule-based detection systems are reactive by design: they identify an attack only after encryption has commenced, by which point patient data has already been compromised and clinical workflows disrupted. These systems also operate as binary classifiers, offering no insight into the stage of an attack, and are not hospital-context-aware, treating an alert on an administrative billing system with the same urgency as one on an EHR system holding patient medication histories.

This study addresses these deficiencies by developing Leadeks REWS, an AI-powered, hospital-aware ransomware early warning system that detects ransomware behaviour during the pre-encryption stage, classifies attack progression across three distinct stages, and escalates alerts according to the clinical risk associated with the affected hospital subsystem. The specific objectives of this study are to: (i) develop an AI-powered ransomware early warning system for hospital environments using machine learning techniques; (ii) implement the system using Google Colab for model training and VS Code for application development; and (iii) evaluate the developed system using standard performance metrics including accuracy, precision, recall, and F1-score.

This paper is organised as follows: Section II presents the literature review; Section III presents the proposed methodology; Section IV presents the results and discussion; and Section V concludes the paper with recommendations for future work.

2. Literature Review

Herrera-Silva and Hernández-Álvarez [1] produced an open dataset of sandbox-extracted dynamic features, comprising approximately 2,000 samples and 50 dynamic features, demonstrating that ensemble classifiers such as Random Forest achieve robust accuracy and generalise well across ransomware variants; however, the dataset lacks explicit stage labels. Urooj et al. [2] and Alraizza et al. [5] each surveyed dynamic and machine-learning-based ransomware detection approaches, concluding that behavioural analysis is preferable for zero-day detection and that ensemble methods consistently outperform single-model baselines, while noting persistent challenges around real-time deployment, labelled data scarcity, and the absence of stage-aware, context-sensitive detection.

Almashhadani et al. [11] empirically characterised the ransomware attack lifecycle, identifying

distinct stages, including a detectable pre-encryption window, but did not implement a real-time detection system. Kim et al. [10] examined the security implications of the Internet of Medical Things, finding that hospital subsystems carry varying clinical criticality yet most security solutions treat all systems as equally important, motivating the hospital-aware design of the proposed system.

Sgandurra et al. [3] introduced EldeRan, a machine-learning system that detects ransomware from early behavioural patterns using a regularised logistic regression classifier, achieving high accuracy but operating strictly as a binary classifier with no stage awareness. Scaife et al. [6] developed CryptoDrop, a heuristic file-monitoring system that halted ransomware after a median loss of only 10 files, demonstrating that early intervention is operationally feasible, though its fixed heuristics cannot adapt to new variants. Ahmed et al. [4] presented Peeler, a kernel-level detection system capable of pre-encryption detection, but its deployment requires privileged system access that conflicts with hospital IT stability requirements. Pundir et al. [13] proposed RanStop, a hardware-assisted detector using microarchitectural features, offering low overhead but limited mapping to high-level, hospital-specific context.

Fernando et al. [9] compared classical machine learning against deep learning for ransomware detection and found that Random Forest remains highly competitive for moderate-sized behavioural datasets while being significantly easier to deploy, reinforcing the pragmatic model choice adopted in this study. More recent work by Ngoie et al. [14] and Zhang et al. [8] explored explainable and temporally-aware detection, underscoring the value of interpretability in high-stakes healthcare deployments, though both remain computationally intensive and unvalidated for real-time use.

Across the fourteen studies reviewed, dynamic behavioural features are consistently the most reliable indicators of ransomware activity, and ensemble methods consistently outperform single-model classifiers. However, all reviewed systems, including EldeRan, CryptoDrop, and Peeler, treat detection as a binary problem, provide no hospital-specific contextual intelligence, and lack an operational dashboard for security administrators. Table 1 compares the three most closely related systems against the system proposed in this study.

Table 1: Comparative Analysis of Existing Ransomware Detection Systems

System	**Detection Method**	**Classification**	**Pre-Encryption Detection**	**Hospital-Aware**
EldeRan [3]	ML (Logistic Regression)	Binary	No	No
CryptoDrop [6]	Heuristic rules	Binary	Partial	No
Peeler [4]	ML (Kernel-level)	Binary	Yes	No
Proposed Leadeks REWS	ML (Random Forest ensemble)	Multi-stage (3 classes)	Yes	Yes (EHR, PACS, ADMIN)

These gaps collectively justify the identified research gap: the absence of a system that combines multi-stage classification, hospital-aware contextual intelligence, clinical risk prioritisation, confidence-based alerting, and an operational dashboard, features that no single existing work provides.

3. Methodology

This research adopts a design-science approach combined with the Agile software development life cycle, chosen for its iterative refinement of both the machine learning component and the functional detection system. Development proceeded through three sprints: Sprint One covered dataset preprocessing, stage-label derivation, and training/validation of the Random Forest classifier; Sprint Two covered core system development, including the Flask application, the psutil-based real-time monitoring loop, and MongoDB integration; and Sprint Three covered the dashboard interface, mitigation recommendation module, and search/filter functionality.

3.1. Analysis of the Existing System

Existing hospital ransomware defences rely primarily on signature-based detection and rule-based intrusion detection systems such as Snort and Suricata. These approaches cannot detect novel or zero-day ransomware variants, as polymorphic techniques render signature databases perpetually obsolete, and rule-based systems require constant manual updates that many hospital IT teams lack the resources to sustain. Analysis reveals five specific shortcomings: reactive detection that occurs only after encryption commences; binary classification with no stage-level insight; lack of hospital-specific contextual awareness; high false-positive rates causing alert fatigue; and no clinical risk assessment to prioritise response.

3.2. Proposed System Overview and Architecture

The proposed Leadeks REWS system continuously monitors system behaviour, classifies activity into three stages (normal, pre-encryption, and encryption), identifies the affected hospital subsystem, and assigns a clinical risk level to guide incident response. The architecture is organised into six integrated layers: a Monitoring Layer that collects behavioural data every 60 seconds via a psutil-based agent; a Processing Layer containing a Feature Extraction Module and a Random Forest ML Engine that outputs stage predictions with confidence scores; a Context and Risk Layer that identifies the affected hospital subsystem (EHR, PACS, or ADMIN) and assigns a clinical risk level (Critical, High, or Medium); an Application Layer that surfaces enriched alerts through a Flask web server and dashboard; and a Data Persistence Layer using MongoDB for historical analysis and audit trails.

3.3. System Requirements

Functional and non-functional requirements were derived from the study objectives and the gaps identified in Section II. Table 2 summarises key functional requirements and Table 3 summarises the non-functional requirements.

Table 2: Selected Functional Requirements

ID	Requirement
FR1	Provide a login interface with username/password authentication.
FR2	Continuously collect behavioural features (process counts, API/registry/network activity) every 60 seconds via psutil.

FR3	Classify behaviour into normal, pre-encryption, or encryption using a trained Random Forest model.
FR4	Display a confidence score (0-1) for each classification.
FR5	Detect and alert on pre-encryption activity when thresholds are met.
FR6	Identify the affected hospital subsystem (EHR, PACS, or ADMIN) from behavioural patterns.
FR7	Assign a clinical risk level (Critical, High, or Medium) to each alert.
FR8	Provide a real-time dashboard with auto-refresh, event counts, and threat summaries.
FR9	Display paginated alert history (10 per page) with timestamp, stage, confidence, system, and risk.
FR10	Allow search and filter of alerts by keyword, stage, risk level, and status.
FR11	Provide stage- and risk-specific mitigation recommendations via a modal.
FR12	Persist all alerts in MongoDB for historical retrieval.
FR13	Display doughnut charts of alert distribution by stage and severity.
FR14	Provide a settings page with system info and an alert-reset function.
FR15	Provide interactive EHR, PACS, and ADMIN subsystem interfaces with role-based login.
FR16-18	Allow authorised users to add, view, and delete patient, imaging, and staff records respectively.
FR19	Support password-based encryption/decryption of subsystem files, logging failed attempts.
FR20	Enforce role-based access control over encryption and deletion operations.
FR21	Detect brute-force, privilege-escalation, and cross-system attacks and surface them as [HOSPITAL]-prefixed alerts.

Table 3: Non-Functional Requirements

ID	**Requirement**
NFR1	Complete a detection cycle (collection to alert storage) in under 10 seconds.
NFR2	Return a model prediction in under 50 milliseconds per feature vector.
NFR3	Load dashboard pages in under 2 seconds under normal network conditions.
NFR4	Consume no more than 20% CPU and 500 MB RAM during active monitoring.
NFR5	Write all alerts to MongoDB without data loss.
NFR6	Redirect unauthenticated users to login for any protected route.
NFR7	Render correctly on Chrome, Firefox, and Edge.
NFR8	Maintain a modular architecture separating collection, detection, identification, and visualisation.

3.4. AI Detection Model and Algorithm

A Random Forest ensemble classifier (200 estimators, maximum depth 12, Gini criterion, bootstrap sampling, random state 42) was selected for its robustness to noisy behavioural data, effective handling of high-dimensional feature spaces, reduced overfitting, and native support for probability estimation via predict_proba(), which enables confidence-based alerting. Since the

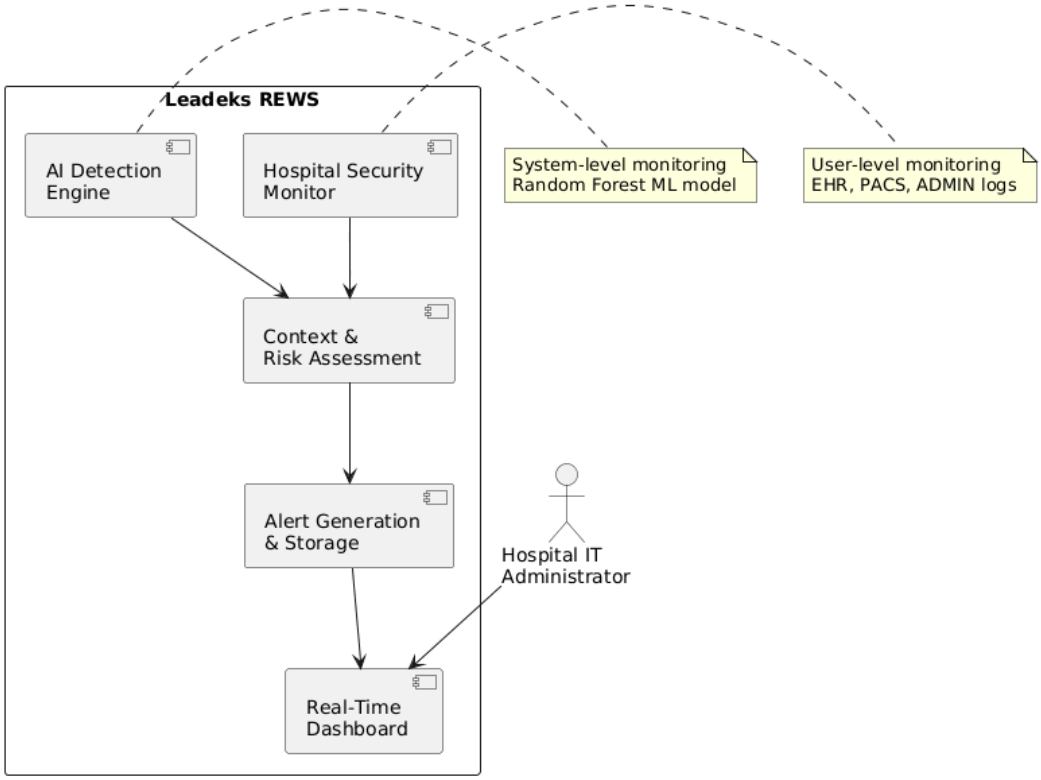

Figure 1: Proposed System Architecture Overview Diagram

source dataset lacked explicit stage labels, a weighted composite score, combining file creation events (weight 3), file read operations (weight 2), and entropy measurements (weight 4), was computed for each sample, with the 33rd and 66th percentiles used as thresholds to assign normal, pre-encryption, and encryption labels respectively.

The detection algorithm applies tiered classification logic to the model's probability outputs: a pre-encryption alert is raised when pre-encryption probability is at least 0.10 and encryption probability is below 0.90; an encryption alert is raised only when encryption probability reaches at least 0.85 and pre-encryption probability is below 0.10; a lower-confidence pre-encryption alert is raised when encryption probability reaches at least 0.40 or pre-encryption probability reaches at least 0.05; otherwise, the sample is classified as normal. This tiered design deliberately prioritises pre-encryption detection over encryption detection, reflecting the system's early-warning philosophy that a pre-encryption alert worth investigating is preferable to an irreversible, confirmed encryption event. A complementary rule-based Hospital Security Monitor analyses hospital subsystem access logs to detect brute-force attacks (three or more failed logins), privilege escalation, and cross-system access attempts, surfacing these as [HOSPITAL]-prefixed alerts within the same dashboard.

4. Results and Discussion

The Leadeks REWS system was fully implemented using Python, Flask, scikit-learn, MongoDB, and psutil, comprising eight integrated modules: data collection and feature extraction, ransomware detection, hospital system identification, real-time monitoring, web dashboard and alert visualisation, mitigation recommendation, and interactive EHR/PACS/ADMIN hospital

subsystems with role-based access control and file encryption/decryption.

4.1. System Interface

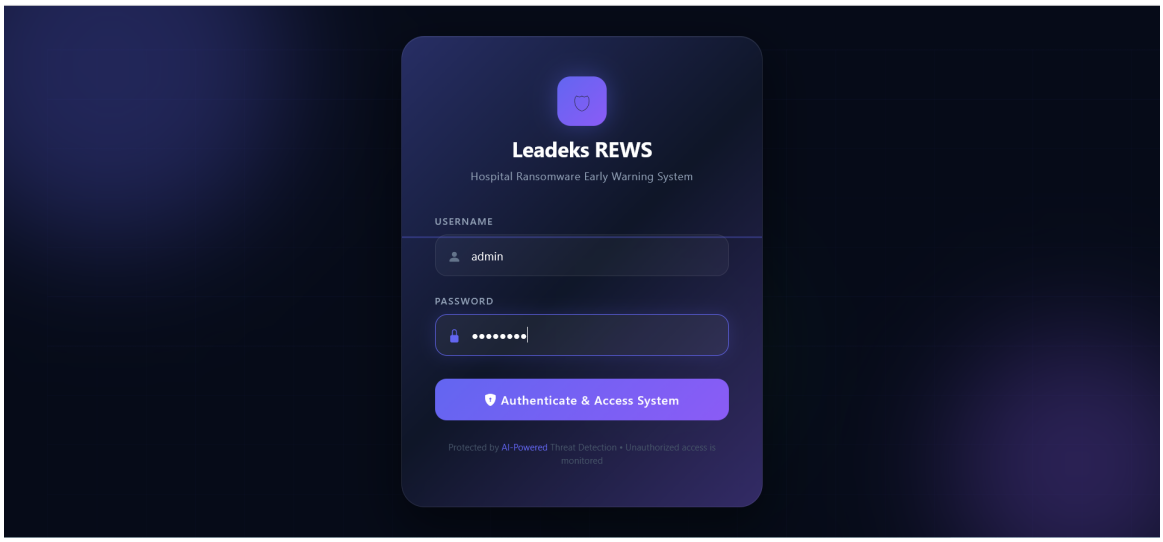

Figure 2: Login Interface

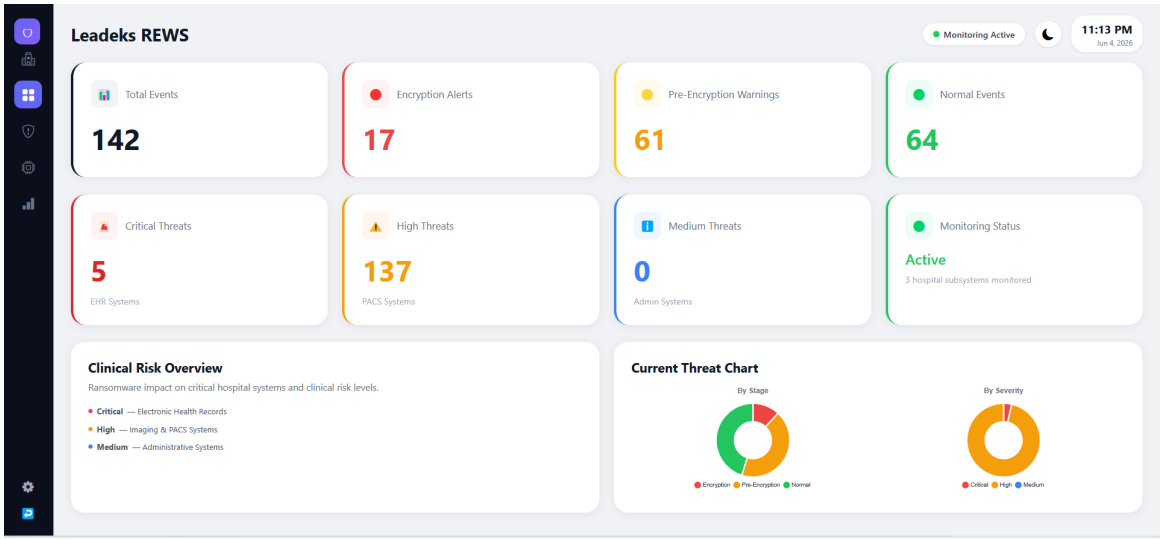

Figure 3: Main Dashboard Landing Page

The dashboard displays Total Events, Encryption Alerts, Pre-Encryption Warnings, Normal Events, and threat counts by clinical risk level, alongside a Clinical Risk Overview bar chart and stage/severity doughnut charts, with a live monitoring status indicator.

The alerts table displays paginated detection events with colour-coded stage badges, confidence scores, affected hospital system, clinical risk level, and alert status, with server-side search and filtering by stage, risk level, alert status, and source.

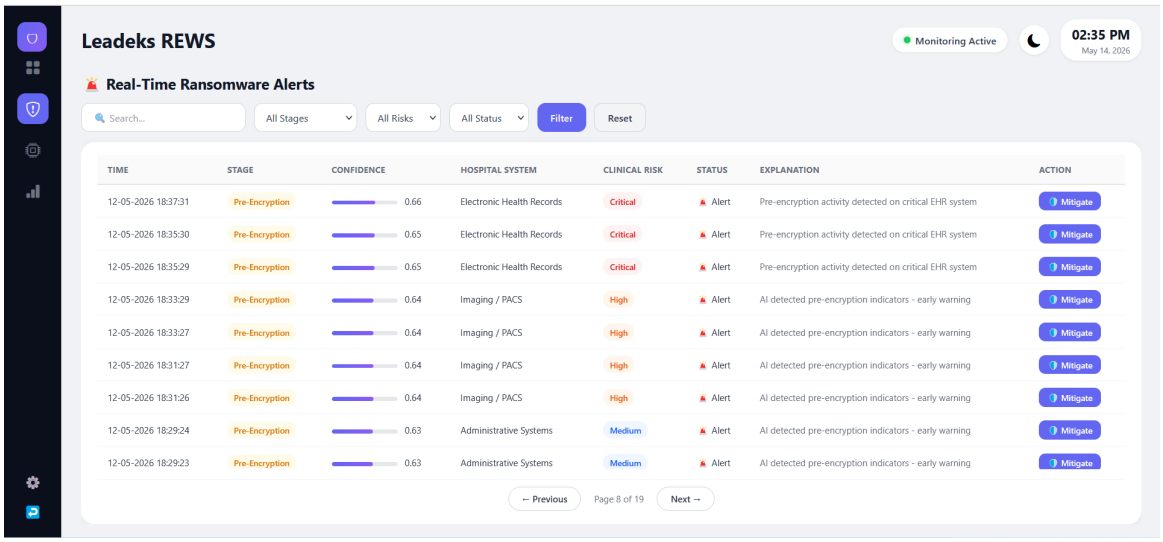

Figure 4: Real-Time Alerts Table

4.2. Model Training and Evaluation

The Random Forest model was trained on an 80/20 stratified split of the 2,000-sample dynamic ransomware dataset (1,600 training samples; 400 test samples), achieving an overall classification accuracy of 91%. Stage-specific performance was: Normal, precision 0.92, recall 0.94, F1 0.93; Pre-encryption, precision 0.89, recall 0.87, F1 0.88; Encryption, precision 0.91, recall 0.92, F1 0.91. Critically, the confusion matrix revealed zero instances of normal behaviour misclassified as encryption and zero instances of encryption misclassified as normal, an asymmetric error profile that is desirable for a security system prioritising early, investigable warnings over missed critical events. Compared against Logistic Regression (78% accuracy, 65% pre-encryption recall), a single Decision Tree (82% accuracy, showing overfitting), and an SVM with RBF kernel (84% accuracy, nearly double the training time), the Random Forest classifier outperformed all baselines across every metric, with a 13-percentage-point improvement in pre-encryption recall over Logistic Regression.

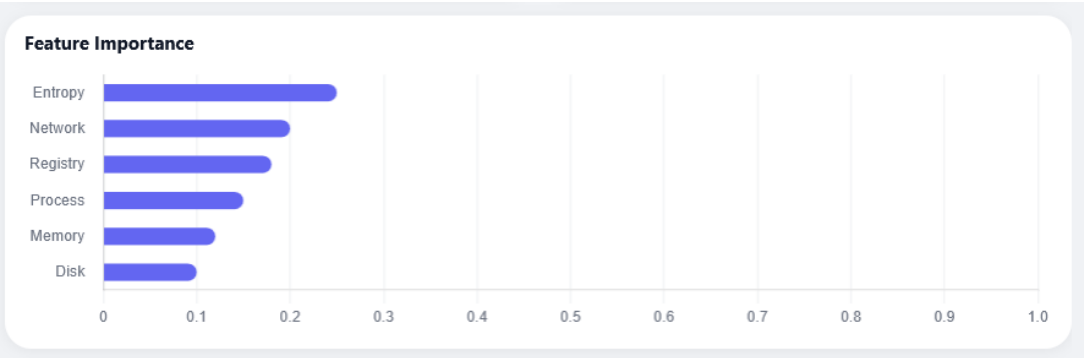

Figure 5: Feature Importance Ranking

Registry write operations emerged as the most important feature, consistent with ransomware's characteristic persistence behaviour, followed by API call frequency and DLL loading patterns, validating the feature selection decisions made during system design.

4.3. Test Cases and Performance Evaluation

Twenty-five structured test cases were executed covering normal operation, boundary conditions, error handling, and hospital subsystem security features; Table 4 summarises a representative subset, with all cases ultimately passing (one case required a timestamp-format fix before passing).

Table 4: Representative System Test Cases and Results

TC ID	Objective	Result
TC-01	Login with valid credentials	Redirected to dashboard — Pass
TC-02	Login with invalid password	Error message shown — Pass
TC-03	Login with non-existent username	Error message shown — Pass
TC-04	Dashboard metric cards match database counts	Counts matched exactly — Pass
TC-05	Alerts page shows 10 alerts per page by default	10 rows displayed — Pass
TC-06	Server-side search filters alerts correctly	Filtered results correct — Pass
TC-07	Stage filter works correctly	Only matching stage shown — Pass
TC-08	Clinical risk filter works correctly	Only matching risk shown — Pass
TC-09	Mitigation modal displays correct recommendations	Modal displayed correctly — Pass
TC-10	Mitigation API handles invalid timestamp gracefully	Fallback JSON returned — Pass
TC-11	Monitoring page displays live system metrics	Values updated every 2s — Pass
TC-12	Analytics page renders all four charts	All charts rendered — Pass
TC-13	Settings page displays system information	Information displayed correctly — Pass
TC-14	Reset alerts deletes all documents	Database emptied — Pass
TC-15	System handles zero-alert state (boundary)	Empty state displayed — Pass
TC-16	Mitigation modal handles missing alert timestamp (error)	Fixed after initial failure — Pass
TC-17	EHR subsystem login with valid credentials	Login accepted — Pass

The system was further evaluated against the non-functional requirements in Table 3, covering response time, resource utilisation, reliability, security, and cross-browser compatibility. Table 5 summarises the evaluation; all measured requirements were met, and brute-force/privilege-escalation detection achieved 100% accuracy under controlled test conditions.

Table 5: Non-Functional Performance Evaluation Results

Metric	Requirement	Result	Status
Detection latency	Under 10 seconds per cycle	Completed within limit	Pass

Model inference time	Under 50 ms per prediction	Well within limit	Pass
Dashboard load time	Under 2 seconds	Loaded within limit	Pass
CPU utilisation	Below 20%	Within limit during monitoring	Pass
Memory usage	Below 500 MB	Within limit during monitoring	Pass
Alert persistence reliability	No data loss	All alerts stored correctly	Pass
Session security	Redirect unauthenticated users	Redirected correctly	Pass
Cross-browser compatibility	Chrome, Firefox, Edge	Rendered correctly on all three	Pass
Brute-force/privilege-escalation detection	Accurate detection	100% under controlled conditions	Pass

4.4. Discussion of Results

The results demonstrate that ransomware exhibits detectable behavioural precursors, elevated API calls, registry modifications, and file operations, that can be distinguished from normal system activity using ensemble machine learning. The 87% pre-encryption recall confirms that a meaningful intervention window exists before mass encryption begins, directly addressing the reactive nature of traditional signature-based and rule-based systems. The system's hospital-aware contextual intelligence and clinical risk prioritisation extend beyond any of the three closely related systems reviewed in Section II, none of which provides stage classification, hospital subsystem identification, or an operational dashboard. The dual-engine architecture, combining the AI detection model with the rule-based Hospital Security Monitor, further extends protection to user-level threats such as brute-force and privilege-escalation attacks within the embedded hospital subsystems. Limitations remain: the model was trained on sandbox-derived data, and validation in a live hospital environment is required before deployment, while behavioural features are approximated from system-level metrics rather than direct API interception, a deliberate trade-off favouring deployability over exact feature replication.

5. Conclusion

This paper has presented the design and implementation of Leadeks REWS, an AI-powered, hospital-aware ransomware early warning system that detects ransomware activity during the pre-encryption stage and generates clinically prioritised alerts for hospital subsystems. The system fully achieves its stated aim, evidenced by 91% overall classification accuracy, 87% pre-encryption recall, successful identification of affected hospital subsystems, a real-time dashboard with confidence-scored and clinically prioritised alerts, and a dual-engine architecture extending detection to brute-force, privilege-escalation, and cross-system threats. The study demonstrates that ransomware exhibits detectable behavioural precursors distinguishable from normal activity through ensemble machine learning, and that clinical risk integration is both

feasible and valuable for tiered incident response in healthcare settings. While the reliance on sandbox-based training data and the need for live-environment validation remain limitations, the stage-aware, context-aware paradigm demonstrated here provides a solid foundation for future work and is applicable beyond ransomware to other multi-phase cyber threats facing resource-constrained hospital environments.

Funding

The authors received no specific funding for this work.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Herrera-Silva, J. A., & Hernández-Álvarez, M. (2023). Dynamic feature dataset for ransomware detection using machine learning algorithms. *Sensors, 23*(3), 1053.

[2] Urooj, U., Al-Rimy, B. A. S., Zainal, A., Ghaleb, F. A., & Rassam, M. A. (2021). Ransomware detection using dynamic analysis and machine learning: A survey and research directions. *Applied Sciences, 12*(1), 172.

[3] Sgandurra, D., Muñoz-González, L., Mohsen, R., & Lupu, E. C. (2016). Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. *arXiv preprint arXiv:1609.03020*.

[4] Ahmed, M. E., Kim, H., Camtepe, S., & Nepal, S. (2021). Peeler: Profiling kernel-level events to detect ransomware. In *Proceedings of the European Symposium on Research in Computer Security*.

[5] Alraizza, A., Alotaibi, B., Alshammari, R., & Alshammari, A. (2023). Ransomware detection using machine learning algorithms: A survey. *Big Data and Cognitive Computing, 7*(3), 143.

[6] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. B. (2016). CryptoDrop: Stopping ransomware attacks on user data. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*.

[7] Zahoora, U., Qureshi, M. A., & Babar, M. A. (2022). Ransomware detection using deep learning-based unsupervised feature extraction and a cost-sensitive Pareto ensemble classifier. *Scientific Reports, 12*, 16567.

[8] Zhang, Y., Wang, X., & Li, Z. (2025). Algorithmic segmentation and behavioral profiling for ransomware detection using temporal-correlation graphs. *arXiv preprint arXiv:2501.17429*.

[9] Fernando, S., Komninos, N., & Chen, S. (2020). *A study on the evolution of ransomware detection using machine learning and deep learning techniques*. City, University of London.

[10] Kim, S., Lee, J., & Park, H. (2023). Security implications of the Internet of Medical Things in healthcare environments. *Journal of Medical Systems, 47*(2), 25.

[11] Almashhadani, A. O., Kaiiali, M., Sezer, S., & O'Kane, P. (2019). A multi-classifier network-based crypto ransomware detection system: A case study of Locky ransomware. *IEEE Access, 7*, 47053–47067.

[12] Agrawal, R., Verma, S., & Sharma, A. (2022). Ransomware detection and prevention in healthcare systems: A systematic review. *Computers & Security, 120*, 102805.

[13] Pundir, N., Tehranipoor, M., & Rahman, F. (2020). RanStop: A hardware-assisted runtime crypto-ransomware detection technique. *arXiv preprint*.

[14] Ngoie, C., Wa Nkongolo, M., Azugo, P., & Tokmak, M. (2025). Hybrid large language models for interpretable ransomware detection. *arXiv preprint*.

[15] World Health Organization. (2022). *Cybersecurity and privacy considerations in digital health*. World Health Organization.

[16] Nigerian Communications Commission Computer Security Incident Response Team. (2022). *Ransomware advisory for critical infrastructure*. Nigerian Communications Commission.

Copyright and License

Journal Website: https://ktrendjournals.org/