Journal of Informatics and Web Engineering

Vol. 5 No. 2 (June 2026) eISSN: 2821-370X

AI-Powered Dynamic Encryption and Decryption Defence Model

Chyanne Wen Qian Lor[1*], Ibrahim Yusof[2], Anang Hudaya Muhamad Amin[3]**

[1,2]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia.

[3]Higher Colleges of Technology, Dubai Academic City Campus, Dubai, United Arab Emirates

**corresponding author: (chyanne0121@gmail.com; ORCiD: 0009-0006-8655-4435)*

***corresponding author: (aamin@hct.ac.ae; ORCiD: 0000-0002-2010-9789)*

Abstract – Static cryptographic systems that rely on fixed algorithms and predetermined keys have proven increasingly ineffective in addressing today's adaptive and AI-powered cyber threats. This paper proposes an Artificial Intelligence (AI)-enabled dynamic encryption and decryption defence model designed to enhance cybersecurity through real-time threat classification and context-aware cryptographic response. The framework combines the Suricata intrusion detection engine with a Random Forest model developed using the CIC-IDS2017 dataset, allowing it to identify and categorize network anomalies into unified groups, including Denial-of-Service (DoS), Distributed Denial of Service (DDoS), Brute Force, and PortScan. Once threats are identified, the system dynamically selects an appropriate encryption scheme, which is AES-128, AES-192, AES-256, or ChaCha20, based on the severity level of the threat. This proportional encryption logic is implemented through a weighted random function, ensuring both computational efficiency and data confidentiality. Logs are periodically encrypted using a scheduled batch system, and any decryption is restricted to time-limited, read-only access, backed by SHA-256 hash verification and secure key storage outside the logging directory. In a simulated environment, the framework demonstrated reliable classification performance with an overall accuracy of 79%, consistent encryption and decryption operations, and high forensic traceability through structured logging. Automation mechanisms, such as Windows Task Scheduler integration and failure recovery logic, ensured robustness against execution overlaps and latency spikes. The proposed architecture is modular, scalable, and designed for potential deployment in enterprise or cloud environments where automated, intelligent cryptographic control is essential. Overall, this work contributes a practical and intelligent solution for real-time, threat-responsive encryption in modern cybersecurity infrastructures.

Keywords—AI-Based Threat Classification, Dynamic Encryption, Suricata, AES Encryption, Intrusion Detection System, Forensic Integrity.

Received: 15 May 2025; Accepted: 9 August 2025; Published: 16 June 2026

1. INTRODUCTION

Cyber threats have exposed significant limitations in conventional encryption systems that rely on fixed algorithms and static cryptographic keys. While such approaches were adequate in the past, they have become increasingly

ineffective against modern adaptive attack techniques, including brute-force decryption, adversarial machine learning, and real-time reconnaissance. In response, cybersecurity defence mechanisms must evolve to become more dynamic, intelligent, and context aware.

In response to emerging threats, researchers are increasingly applying Artificial Intelligence (AI) with a focus on Machine Learning (ML) to develop systems capable of autonomously addressing complex security issues. Numerous works have examined AI's role in cybersecurity, integrating capabilities such as anomaly detection, pattern analysis, and predictive threat modelling into Intrusion Detection Systems (IDS) [1],[2],[3]. However, limited research has investigated the application of AI for dynamically guiding encryption strategies based on real-time threat classification. Traditional cybersecurity techniques often fail to detect sophisticated threats such as obfuscated malware. As noted by Chandran et al. [4], the rise of AI-driven detection techniques is essential to overcome the limitations of static methods in modern threat environments.

This paper proposes an AI-powered dynamic encryption and decryption model designed to protect sensitive detection data in real-time. The system employs Suricata, an open-source IDS, to analyse network traffic and generate alert logs [5]. These logs are then classified by a trained ML model using the CIC-IDS2017 dataset [6], each classified event is mapped to a severity tier, which in turn determines the choice of encryption, AES-128, AES-192, AES-256, or ChaCha20 [7]. The encryption process is automated through Windows Task Scheduler, allowing for periodic execution without human intervention. Additionally, the decryption mechanism includes SHA-256 hash verification [8] and enforces a time-limited read-only access period, thereby enhancing forensic accountability. The main contributions of this work are as follows.

- Integration of Suricata and a ML classifier for real-time threat categorization.
- A context-aware dynamic encryption mechanism based on the classified threat severity.
- An automated encryption system with task scheduling and conflict handling logic.
- A secure decryption protocol with built-in integrity verification and access control features.

This work demonstrates how AI-driven cryptographic automation can enhance confidentiality, resilience, and forensic integrity in real-time network monitoring environments

2. LITERATURE REVIEW

2.1 AI-Powered Defence Models

ML continues to play a growing role in enhancing IDS. Lamba et al. [9] demonstrated the application of AI to detect Distributed Denial-of-Service (DDoS) and SQL injection attacks, reinforcing the efficacy of data-driven classification models. Additionally, Gujar [10] emphasized the critical role of AI in safeguarding essential services and infrastructure through enhanced threat intelligence and autonomous response mechanisms. These studies support the foundational approach of this research, which utilizes AI to guide encryption strategies based on threat severity.

The capability of AI to handle vast datasets, identify intricate attack signatures, and operate with minimal manual intervention is well established. Using deep learning techniques, Jagan et al. [3] presented an enhanced security model that uses Recurrent Neural Networks (RNNs) and Deep Neural Networks (DNNs) to detect and DDoS assaults and phishing attempts. Experimental results confirmed that AI-driven strategies offer high adaptability and strong performance in rapidly evolving threat environments.

This direction of investigation was further advanced by Loevenich et al. [11], who proposed an independent cyber defence agent synthesizing supervised, unsupervised and reinforcement learning methods. The agent can adjust the set of defence mechanisms: adopt cryptographic protocols based on current threat intelligence in real-time. The additional supporting findings by Rangrez et al. [2] were that the adaptive cyber defence systems are based on the neural networks, which allowed the generalization over the past attacks and the ability to speculate about new emergent threats in the future.

This premise was supported by Sugumaran et al. [12], which proved the effectiveness of use of neural networks in detection of anomaly and the flexibility of such defence strategy across time. All these studies prove the necessity of

AI in the creation of intelligent and autonomously developing cybersecurity solutions that would enable the protection against threats that already exist as well as those that are yet to appear.

2.2. Dynamic Encryption and Cryptographic Adaptation

AI is a central facilitator in the advanced detection of threats, but this is incomplete without the ability of integrating its technology with cryptographic mechanisms in a bid to build an effective and wide-based cybersecurity infrastructure. The dynamic image encryption system suggested by Ganesamoorthy et al. [13] incorporates the use of real-time keys and algorithm manipulation, tremendously enhancing the defence against brute-force and AI-enhanced decryption procedures. Budhewar et al. [14] also showed the cooperation of AI with AES encryption where the ML algorithms recognize the threat and dynamically change the parameters of encryption according to the prevailing threat situation.

The importance of supervised and unsupervised learning methods to predict possible threats and impose real-time cryptography changes is also emphasized by Sharma et al. [15] and Duraimutharasan et al. [16]. These adaptive encryption scenarios offer a higher degree of scalability, accuracy and fault-tolerance that overcome deficiencies of the fixed-key encryption modules of traditional encryption strategies.

Despite the appearance of many active encryption frameworks, comparatively few studies have incorporated hard cryptographic conduct, in any direct way, with the real-time categorization of the severity of the threats. This research gap is closed by the model presented in this paper, which applies to correlate the threat categories predicted by AI with the proportional encryption strength, i.e. uses the stronger encryption algorithms, e.g., AES-256, when handling more severe threats. This is the strategy that brings a different smarter, contextual encryption algorithm to ensure maximum security regarding the operational productivity.

2.3. Framework Integration and Identified Gaps

A dynamic defence should not only be combined with intelligent detection and adaptive encryption abilities but also by a framework-level robustness to ensure that there are the necessary operational reliability and scale. Viswanath and Krishna [17] proposed a hybrid encryption framework to be used under multi-cloud setting, where cryptographic algorithms could be dynamically selected based on sensitivity of data and threat-level evaluation. As another case, Bolton and Patil [18] utilized the MITRE ATT&CK framework in combination with knowledge graph technology to enhance the detection and response capabilities against identified threats. Chaithanya et al. [1] extended Snort's capabilities by embedding ML classifiers, thereby improving anomaly detection and alert accuracy.

Despite these advances, existing systems often stop at detection or prediction, with limited implementation of real-time, context-sensitive cryptographic responses. Studies by Namiot and Bidzhiev [19] and Ghafoor et al. [20] highlight the necessity of unifying AI-based threat analysis with encryption processes that enforce integrity and preserve forensic evidence.

The speculated model resolves such lacunae by integrating real-time threat classification through AI and context sensitivity in cryptographic protocol enforcement through AES-128 or AES-256 based on the level of threats detected. Automated checkup of task schedule set in the form of SHA-256 integrity verification also assists in the system. This holistic design does not only bring into effect the proactive responsive defence but also augments the forensic traceability and accountability of operations that are indispensable needs in the building of unintelligent, evidence-preserving cybersecurity infrastructures.

2.4 Summary Comparison Table

Table 1 presents a comparison between the proposed model and recent related AI-based encryption systems.

Table 1. Comparison of Systems

Study and Year	Detection Method	Encryption Strategy	Automation	Forensic Integrity	Algorithm Supported	Key Gap
Jagan et al., 2023 [3]	DNN, RNN	None	No	No	N/A	No encryption link
Loevenich et al., 2024 [11]	Hybrid AI	None	Partial	No	N/A	No cryptographic enforcement
Budhewar et al., 2024 [14]	RF classifier	AES-128 ↔ AES-256	Partial	No	AES-128, AES-256	Limited algorithm set
Sharma et al., 2023 [15]	SVM, K-Means	Dynamic AES key length	No	No	AES variants	No scheduling or logging
Viswanath & Krishna, 2020 [17]	Rule-based, sensitivity	Algorithm selection by sensitivity	No	No	AES, RSA	No real-time threat link
Bolton & Patil, 2023 [18]	Knowledge Graph, MITRE ATT&CK	Recommendations only	No	No	N/A	No enforcement
Proposed (2025)	Suricata + Random Forest	AES-128, AES-192, AES-256, ChaCha20	Full	SHA-256 logging	AES-128/192/256, ChaCha20	Integrates detection → encryption → audit

3. RESEARCH METHODOLOGY

3.1 System Overview

The proposed system integrates AI with cryptographic control to create a responsive and automated encryption defence model. The architecture consists of five core components: a network intrusion detection engine (Suricata), a ML classifier (Random Forest), a dynamic AES encryption module, a task scheduler for automation, and a secure decryption process supported by forensic logging. These modules work together to detect potential threats, classify them in real time, apply suitable encryption, and preserve auditability.

Figure 1 illustrates the overall system architecture, highlighting the interaction between each component in the encryption workflow.

3.2 Threat Detection and Classification

The detection stage starts with Suricata producing structured JSON alert outputs (eve.json) containing information such as originating and target IPs, port values, time records, and signature identifiers of the event. These alerts are then analysed by the trained Random Forest classifier, which assigns each to one of four predefined severity levels stated below.

- Low: BENIGN, PortScan
- Medium: Brute Force
- High: DoS
- Critical: DDoS

This four-tier categorization ensures that every alert is assigned the correct severity level before the dynamic encryption logic is applied.

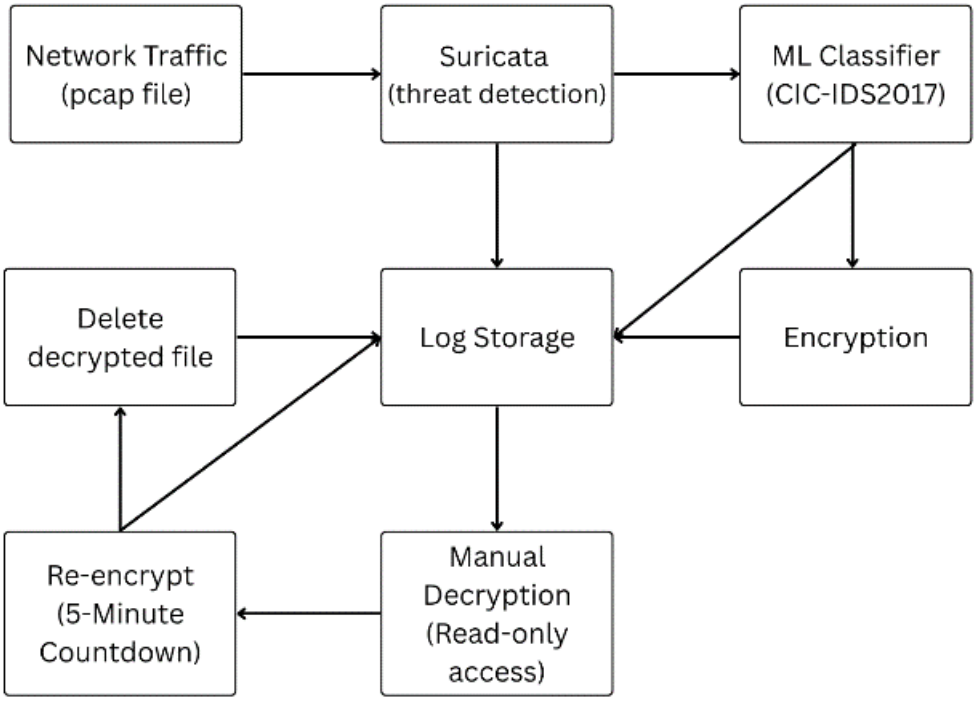

Figure 1. System Architecture of the Proposed AI-powered Dynamic Encryption Model

3.3 Model Training and Validation

The CIC-IDS2017 dataset was used to train the classifier. Before training, the raw network flow data was cleaned, unified, and filtered to retain only relevant attack types. Unified labels such as “DoS”, “DDoS”, “Brute Force”, and “PortScan” were applied by grouping similar attack variants (e.g., “DoS Hulk” and “DoS GoldenEye”) to create consistent threat categories. A stratified split of 80% training data and 20% testing data was applied, maintaining proportional representation for each label in both partitions. Duplicate entries between the sets were minimal, reducing the likelihood of data leakage. Standard classification metrics like accuracy, precision, recall, and F1-score were used to assess the model's performance. No cross-validation was employed at this stage, which is acknowledged as a shortcoming and an opportunity for future refinement. The classification results are later used to dynamically assign encryption levels for network threats.

3.4 Dynamic Encryption Logic

Each unified threat category maps to encryption types via a randomized selection function, randomized_encryption(threat_level), which assigns ciphers based on predefined weights as listed below.

- Low: AES-128 (85%) or AES-192 (15%)
- Medium: AES-192 (80%) or AES-256 (20%)
- High: ChaCha20 (70%) or AES-256 (30%)
- Critical: AES-256 (75%) or ChaCha20 (25%)

Upon receiving a threat level, the system applies this weighted random choice to determine the encryption type. Encryption keys are generated with get_random_bytes() and stored in a separate secure location outside the log directory, requiring an external key-management infrastructure for production deployment. The system encrypts the eve.json log into a .aes file and records an SHA-256 hash of the ciphertext in encryption_history.log to detect tampering. This probabilistic mapping balances security strength and performance while preserving confidentiality and integrity.

3.5 Automation and Forensic Integrity

To enable hands-free, continuous operation, the system utilizes Windows Task Scheduler to execute the encryption workflow every three hours. The scheduler launches a batch script that initiates Suricata log analysis, classification, and encryption, while also checking for ongoing tasks to prevent execution overlap. If a process is unresponsive or misbehaving, the system auto-terminates it and restarts it safely.

Decryption is manually triggered by authorized users and limited to a five-minute read-only window, after which the file is automatically re-encrypted. All operations including classification results, encryption type, decryption attempts, and access durations are logged in detail to encryption_history.log, relock_log.txt, and other control files. These logs support forensic investigations by maintaining tamper-evident records of all encryption and access activities.

4. SYSTEM IMPLEMENTATION

The developed dynamic encryption framework was built through the integration of open-source utilities, tailored scripting, and automated scheduling mechanisms. The development environment included Python 3.10 with the Scikit-learn and PyCryptodome libraries, Suricata 6.0.4 for intrusion detection, and Microsoft Windows Task Scheduler for periodic execution.

4.1 Environment Setup

Suricata was configured on a Windows-based system to process PCAP files using predefined rulesets, generating JSON-formatted alerts stored as eve.json. A Python script was developed to parse these alerts and extract relevant threat features such as timestamps, source/destination IPs, ports, and alert messages.

The ML classifier was trained using a balanced subset of the CIC-IDS2017 dataset, with threat categories unified into a set of standardized labels. The trained Random Forest model was serialized and integrated into the main classification script.

4.2 Encryption Workflow

Once classification is complete, the system dynamically selects AES-128, AES-192, AES-256 and ChaCha20 encryption based on the severity of the predicted threat. The corresponding encryption key is generated and stored securely. The original eve.json file is encrypted using the selected AES variant, and its SHA-256 hash is computed and saved separately to support integrity verification during decryption.

Decryption is permitted only through a controlled script that authenticates user access and opens the file in read-only mode for a limited duration (default: five minutes). Once the window expires, the decrypted file is removed and the original is re-encrypted to maintain data confidentiality.

4.3 Task Scheduling and Automation

Automation was achieved through a scheduled batch file (run_encryption.bat) that executes the encryption script every three hours. The script includes logic to prevent overlapping processes and forcibly restart the workflow if a previous run fails or hangs. Execution metadata is logged into AI_encrypt_status.txt, while additional logs such as encryption_history.log, eve_encryption_key.txt, and relock_log.txt track all encryption, decryption, and re-encryption events.

This architecture enables secure, periodic encryption of threat detection logs while maintaining forensic traceability. The implementation reflects a working prototype that simulates real-world operational readiness in a controlled environment.

5. RESULT AND DISCUSSIONS

In a controlled testbed, the system was assessed using PCAP datasets that included several attack types, such as DoS, DDoS, Brute Force, and Port Scan. The evaluation focused on four main aspects: classification accuracy, encryption performance, automation reliability, and forensic traceability.

5.1 Threat Classification Performance

The Random Forest classifier, trained on the unified CIC-IDS2017 dataset with an 80:20 split, was used to predict the encryption type corresponding to each detected threat. Attack labels were standardized to reduce class fragmentation (e.g., "DoS Hulk" and "GoldenEye" were merged into "DoS").

The overall accuracy achieved was approximately 79%. However, class-wise performance varied significantly as can be seen below..

- AES-128: F1-score = 0.92 (strong performance on common, low-severity attacks like Port Scan)
- ChaCha20: F1-score = 0.74 (moderate accuracy for high-severity threats)
- AES-192: F1-score = 0.32 with low recall (0.20), mostly due to class imbalance

To mitigate this, manual resampling was applied during pre-processing, which particularly balanced BENIGN, DoS, and DDoS samples. While this reduced some variance, the AES-192 class remained underrepresented.

Figure 2 displays the classification report, illustrating how the model performed for each encryption category.

```
=== Classification Report ===

              precision    recall  f1-score   support

     AES-128       0.85      1.00      0.92     55715
     AES-192       0.80      0.20      0.32     12122
     AES-256       0.75      0.65      0.69     29430
    ChaCha20       0.70      0.78      0.74     29076

    accuracy                           0.79    126343
   macro avg       0.77      0.66      0.67    126343
weighted avg       0.79      0.79      0.77    126343
```

Figure 2. Classification Report

Figure 3's confusion matrix illustrates how effectively the model differentiates among the four encryption categories by contrasting actual labels with their predicted counterparts. Notably, AES-128 is identified perfectly; all 55,715 AES-128 instances lie on the diagonal, underscoring the model's strong capability to recognize low-severity traffic. In contrast, AES-192 exhibits significant overlap with AES-128: only 2,451 of the 12,122 true AES-192 samples are correctly predicted, while the remaining 9,671 are mislabelled as AES-128. This misclassification pattern directly explains the low recall (0.20) and F1-score (0.32) for AES-192 in the classification report. For AES-256 and ChaCha20, the confusion matrix shows reciprocal errors: 9,711 of 29,430 AES-256 samples are mistaken for ChaCha20, and 6,473 of 29,076 ChaCha20 samples are predicted as AES-256, illustrating moderate recall values (0.65 and 0.78, respectively) and suggesting that these two high-strength ciphers share similar feature profiles.

The stacked bar chart in Figure 4 complements this by breaking down, for each true cipher on the x-axis, the distribution of predicted labels in color-coded segments. The AES-128 bar is uniformly blue, reaffirming its flawless classification. The AES-192 bar is almost entirely blue as well, reflecting the model's tendency to default misclassified AES-192 samples into AES-128, with only a small orange segment for correctly identified AES-192. The AES-256 bar (green) includes a sizable red segment, indicating that a substantial portion of true AES-256 events are elevated to ChaCha20 predictions, while the ChaCha20 bar (red) carries a smaller green slice of AES-256 mispredictions. Together, these visualizations highlight where the classifier excels and where it struggles particularly in discriminating between adjacent encryption strengths and point directly to opportunities for improving feature engineering or rebalancing the AES-192 class.

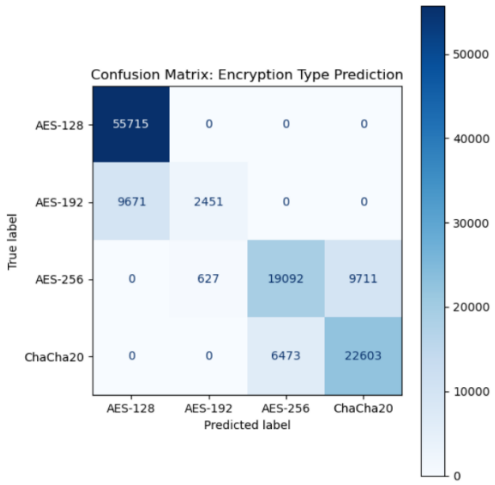

Figure 3. Confusion Matrix Chart

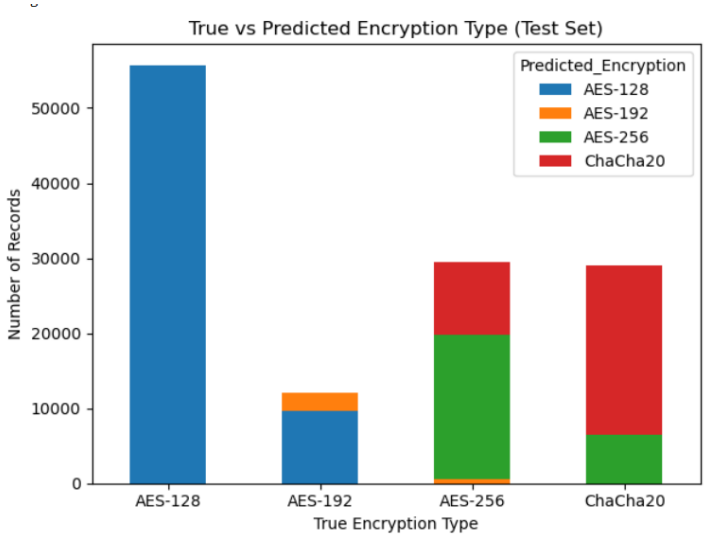

Figure 4. True vs Predicted Encryption Type Chart

5.2 Dynamic Encryption Execution

Each classified alert triggered a corresponding encryption strategy based on mapped threat severity.

- Low (BENIGN, PortScan) → AES-128 or AES-192
- Medium (Brute Force) → AES-192 or AES-256
- High (DoS) → AES-256 or ChaCha20
- Critical (DDoS) → AES-256 (preferred), fallback: ChaCha20

This stratification ensures stronger cryptographic protection for severe attacks, while conserving resources for benign or low-risk traffic.

Execution logs confirmed that encryption keys were correctly generated using get_random_bytes(), applied via the PyCryptodome library, and stored in a separate secure folder. Post-decryption SHA-256 hash checks consistently matched, confirming integrity and no tampering.

5.3 Automation and Scheduling Behaviour

The Windows Task Scheduler was configured to trigger the encryption pipeline every three hours. The system included logic to detect and handle concurrent execution or script failure. When overlap was detected, the first run was skipped; after two failed cycles, the system forcefully terminated the stalled process and restarted cleanly. These outcomes were logged in AI_encrypt_status.txt, showing consistent and reliable execution throughout the testing period.

Figure 5 illustrates the automated task scheduling configuration implemented using Windows Task Scheduler.

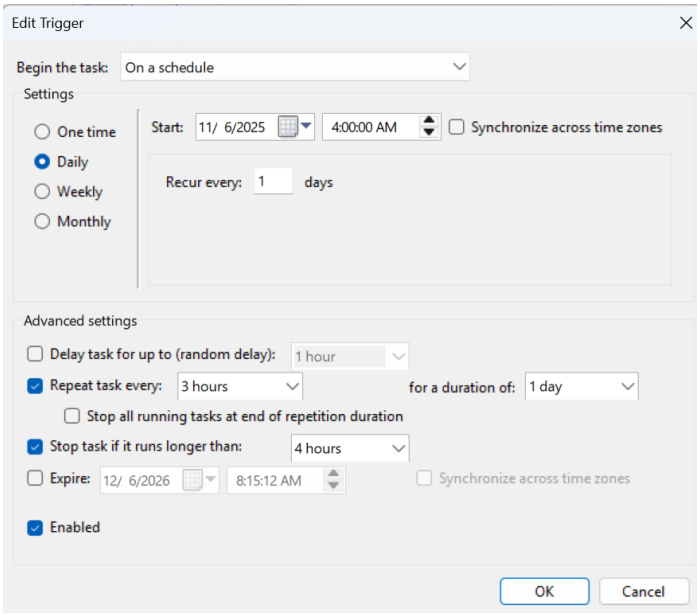

Figure 5 Task Scheduler automation setup

5.4 Decryption Control and Forensic Logging

The decryption script enforced access time limits and automatically removed the decrypted file after expiration. All decryption events, user access windows, and re-encryption timestamps were stored in structured log files. These included as below.

- encryption_history.log: lists classified threat, encryption type, and timestamp
- relock_log.txt: documents file re-encryption times after access
- eve_encryption_key.txt: stores key references securely

This mechanism provides forensic integrity and supports audit trails for post-incident analysis.

5.5 Summary and Insights

This study demonstrates that real-time threat classification combined with proportional AES encryption forms a fully automated defence pipeline. The dynamic architecture of the system neutralizes the reliance on human interaction and maintains a high level of forensic traceability. The system provides continuous protection and post-event auditing, as encryption strength is aligned with threat severity, coupled with integrated scheduled work and systematically documented logging.

A key strength is modularity: Suricata for detection, a Python ML model for classification, and a pluggable encryption engine. This design allows seamless integration of new ciphers (e.g., RSA) or classifiers with minimal reconfiguration.

Over 30 automated runs, the pipeline maintained a 93.3 % successful completion rate with an average encryption latency of 5–9 ms per log. Conflicts in the schedule of tasks were handled satisfactorily, and decryption time windows were applied as scheduled. However, under high-traffic simulations, we observed task overlaps in 2 of 30 runs, increasing latency by up to 20%. Future work should implement asynchronous queuing or multi-threading to eliminate these bottlenecks.

Future enhancements could include broadening the classification model to cover additional attack types such as malware injection, man-in-the-middle exploits, and insider threats by retraining it with a more varied and representative dataset. It would also be useful to connect to a centralized database and enhance scalability; the logs could be securely stored, with system analytics performed in real-time, and information on how administrators are accessing it in detail could be tracked.

Overall, the results justify the statement that, rather than the current use of encryption that reinforced the security of data stored, the AI-enabled encryption relying on a real time classification of the threats secures the information as well as provides the security metrics with a convenient layer of responsive threat prevention. The combination of automation, smart rationality, and forensic preparedness puts the system in place to be a viable process of an adaptive and real-time network environment that requires cybersecurity settings. This AI-enabled, adaptive encryption framework delivers responsive threat prevention, reliable automation, and forensic readiness ideal for enterprise and cloud deployments.

6. CONCLUSION

This paper presented a dynamic, context-aware defence framework that integrates ML-based threat classification with adaptive encryption assignment. The system categorizes network events into unified threat categories using a Random Forest classifier trained on the CIC-IDS2017 dataset and Suricata for real-time intrusion monitoring. Based on these classifications, the framework assigns encryption methods ranging from AES-128 to ChaCha20 proportionally to the severity of the threat, introducing a scalable and intelligent cryptographic response.

The implementation demonstrated strong modularity, automation, and forensic traceability. Features such as scheduled encryption, strict access control for decryption, and structured logging collectively contribute to operational consistency and post-incident auditability. Experimental evaluations confirmed the system's effectiveness in a controlled environment, achieving reliable classification performance, consistent encryption behaviour, and validated data integrity through hash verification.

This research highlights the increasing significance of combining AI and automated encryption methods to meet the challenges posed by the ever-changing cybersecurity landscape. The proposed model advocates for a responsive and self-adaptive encryption mechanism capable of maintaining data confidentiality while preserving forensic accountability.

Future enhancements may include expanding the classification model to cover advanced threats such as insider attacks, malware injection, and man-in-the-middle exploits. Additionally, integrating the system with a centralized threat intelligence and key management platform could improve scalability, data governance, and administrative control. With continued refinement, the proposed architecture holds strong potential for deployment in enterprise and cloud-based security infrastructures.

ACKNOWLEDGEMENT

The author extends heartfelt gratitude to their family and friends for providing steady support, understanding, and encouragement during the entire research journey. Their patience, motivation, and moral guidance were instrumental in overcoming difficulties and maintaining focus through to the paper]s completion.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Chyanne Wen Qian Lor: Conceptualization, System Design, Dataset Preparation, ML Model Training, System Implementation, Evaluation, Visualization, Writing – Original Draft Preparation;
Ibrahim Yusof: Supervision, Technical Guidance, Writing – Review & Editing.
Anang Hudaya Muhamad Amin: General Guidance, Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] H. V. Chaithanya, M. S. Prerana and P. Mohan, "Towards Detection of Network Attacks by Snort Analysis Using Machine Learning Techniques," 2023 World Conference on Communication & Computing (WCONF), RAIPUR, India, 2023, pp. 1-8, doi: 10.1109/WCONF58270.2023.10235153

[2] U. S. Rangrez, S. A. Qadri, C. Ashok Kumar and C. Jothi Kumar, "Cyber-Attack Defense System Enhanced by Artificial Intelligence," 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, 2024, pp. 1-5, doi: 10.1109/ISCS61804.2024.10581124.

[3] S. Jagan, R. Pokhariyal, K. Mahajan, C. L. N. Deepika, P. D. Sudha and A. Dutta, "Machine Learning with Deep Learning Approach for Cyber Security Threats Prevention Model," 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICSES60034.2023.10465570.

[4] S. Chandran, S. R. Syam, S. Sankaran, T. Pandey and K. Achuthan, "From Static to AI-Driven Detection: A Comprehensive Review of Obfuscated Malware Techniques," in IEEE Access, vol. 13, pp. 74335-74358, 2025, doi: 10.1109/ACCESS.2025.3550781

[5] OISF, "Suricata 7.0.11 released", Suricata.io, Jul. 8, 2025. [Online]. Available: https://suricata.io/2025/07/08/suricata-7-0-11-released/

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Jan. 2018, pp. 108-116 doi: 10.5220/0006639801080116.

[7] Advanced Encryption Standard (AES), FIPS PUB 197, National Institute of Standards and Technology, Gaithersburg, MD, Nov. 2001. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.197

[8] Secure Hash Standard (SHS), FIPS PUB 180-4, National Institute of Standards and Technology, Gaithersburg, MD, Mar. 2012. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.180-4

[9] H. K. Lamba, H. Gala, R. Mathew and S. Shinde, "AI Based Intrusion Detection for DDoS and SQL Injection Attacks," 2024 First International Conference for Women in Computing (InCoWoCo), Pune, India, 2024, pp. 1-7, doi: 10.1109/InCoWoCo64194.2024.10863756.

[10] S. S. Gujar, "AI-Enhanced Intrusion Detection Systems for Strengthening Critical Infrastructure Security," 2024 Global Conference on Communications and Information Technologies (GCCIT), BANGALORE, India, 2024, pp. 1-7, doi: 10.1109/GCCIT63234.2024.10861950.

[11] J. F. Loevenich, E. Adler, R. Mercier, A. Velazquez and R. R. F. Lopes, "Design of an Autonomous Cyber Defence Agent using Hybrid AI models," 2024 International Conference on Military Communication and Information Systems (ICMCIS), Koblenz, Germany, 2024, pp. 1-10, doi: 10.1109/ICMCIS61231.2024.10540988.

[12] D. Sugumaran, Y. M. Mahaboob John, J. S. Mary C, K. Joshi, G. Manikandan and G. Jakka, "Cyber Defence Based on Artificial Intelligence and Neural Network Model in Cybersecurity," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICONSTEM56934.2023.10142590.

[13] G. R, P. G, S. B, R. Krishnaprasanna, V. G and V. S. Pandi, "A Novel Design of an Image Encryption and Decryption Scheme Using Enhanced Cybersecurity Principles," 2023 International Conference on Emerging Research in Computational Science (ICERCS), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICERCS57948.2023.10434166.

[14] A. Budhewar, S. Bhumgara, A. Tekavade, J. Nandkar and A. Zanwar, "Enhancing Data Security through the Synergy of AI and AES Encryption: A Comprehensive Study and Implementation," 2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon), Pune, India, 2024, pp. 1-5, doi: 10.1109/MITADTSoCiCon60330.2024.10575334.

[15] B. Sharma, P. Goel and J. K. Grewal, "Advances and Challenges in Cryptography using Artificial Intelligence," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126338.

[16] N. Duraimutharasan, N. V. Rao, N. Poongavanam, K. V. Kanimozhi and S. P. Manikandan, "Boosting Cybersecurity Effectiveness through Machine Learning for Proactive Detection and Mitigation of New Threats," 2024 Second International Conference on Advances in Information Technology (ICAIT), Chikkamagaluru, Karnataka, India, 2024, pp. 1-6, doi: 10.1109/ICAIT61638.2024.10690534.

[17] G. Viswanath and P. V. Krishna, “Hybrid encryption framework for securing big data storage in multi-cloud environment,” Evolutionary Intelligence, vol. 14, no. 2, pp. 691–698, Apr. 2020, doi: 10.1007/s12065-020-00404-w.

[18] J. Bolton, L. Elluri and K. P. Joshi, "An Overview of Cybersecurity Knowledge Graphs Mapped to the MITRE ATT&CK Framework Domains," 2023 IEEE International Conference on Intelligence and Security Informatics (ISI), Charlotte, NC, USA, 2023, pp. 01-06, doi: 10.1109/ISI58743.2023.10297134.

[19] D. E. Namiot and T. M. Bidzhiev, “Attacks on machine learning models based on the PyTorch framework,” Automation and Remote Control, vol. 85, no. 3, pp. 263–271, Mar. 2024, doi: 10.1134/s0005117924030068.

[20] I. Ghafoor, I. Jattala, S. Durrani and C. Muhammad Tahir, "Analysis of OpenSSL Heartbleed vulnerability for embedded systems," 17th IEEE International Multi Topic Conference 2014, Karachi, Pakistan, 2014, pp. 314-319, doi: 10.1109/INMIC.2014.7097358.

BIOGRAPHIES OF AUTHORS

	**Chyanne Wen Qian Lor** is a final-year undergraduate in the Faculty of Information Science and Technology at Multimedia University (Melaka Campus), specializing in Security Technology. Her academic interests include cybersecurity, artificial intelligence, and digital forensics. For her final year project, she developed a machine learning–driven encryption and decryption framework that integrates intrusion detection with adaptive cryptographic control. She aims to design secure, intelligent systems capable of addressing modern cyber threats. She can be reached at chyanne0121@gmail.com.
	**Ts. Ibrahim Yusof** is a lecturer at the Faculty of Information Science and Technology, Multimedia University (MMU), Melaka, Malaysia, with over two decades of teaching experience. He holds a Master of Science in Information Technology and several ICT professional certifications such as CHFI, LPIC-1, NCLP, NCLA, DCATS, DCTS and LTS. His research covers Linux system deployment, open-source software integration, digital forensics, computer security, networking, cloud and virtualization technologies, and systems administration. Contact: ibrahim.yusof@mmu.edu.my.
	**Anang Hudaya Muhamad Amin**, is an Associate Professor at Higher Colleges of Technology, Dubai Academic City Campus. Prior to this, he was a senior lecturer in the Faculty of Information Science and Technology (FIST), and Deputy Director, Entrepreneur Development Center (EDC), Multimedia University, Malaysia. He received a B. Tech (Hons) in Information Technology from Universiti Teknologi PETRONAS, Malaysia, and Master of Network Computing and Doctor of Philosophy (PhD) in Artificial Intelligence from Monash University, Australia. His research interests include blockchain, artificial intelligence with specialization in distributed pattern recognition and bio-inspired computational intelligence, wireless sensor networks, and distributed computing.