International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 4 (2025)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 4, 2025

Research Paper

PROACTIVE RANSOMWARE DEFENSE VIA AI-DRIVEN FILE ENTROPY MONITORING AND REAL-TIME PROCESS BEHAVIOR MODELING

[1]Rethish Nair Rajendran, [2]Veera Siva Prasad Rajulapati

[1]*Delivery Management, Cloud Infra and Apps Services (US&C) Unisys Corporation, Albany, NY*

[1]*Email: rethishrnair@gmail.com*

[2]*Application Design & Development Manager, Technology Consulting, 08830, Iselin-New Jersey*

**Corresponding Author Email: sivaprasad.rv.in@gmail.com*

ABSTRACT

Ransomware has become one of the most urgent cybersecurity threats, as it can encrypt enormous amounts of sensitive information in a few seconds and put the digital infrastructure on its knees. Traditional defences, whether using static signatures, heuristic rules, or post-encryption forensics, do not keep up with the fast-changing and obfuscated ransomware families. To overcome these limitations, this paper will present a proactive artificial intelligence-based ransomware detection system that combines the real-time file entropy check with the process behaviour prediction based on Long Short-Term Memory (LSTM) networks. The system makes repeated entropy variation analyses ($\Delta H/\Delta t$) on file blocks to detect the abnormal increase in randomness, and the LSTM learns sequential dependencies in process-level I/O and memory-access activity. The two modules are combined to create a single ThreatScore, which raises pre-encryption alarms in case of anomalous tendencies. The RanSMAP dataset, a publicly available repository of ransomware and benign storage/memory access logs, was used in experiments with the addition of a Kaggle static PE feature dataset to use as a baseline. The proposed model was found to be 99.1% accurate, 98.9% precise, and a false-positive rate was found to be only 0.6% with an average detection latency of 118 milliseconds, which is better compared to classical Support Vector Machine, Random Forest and standalone LSTM baselines. It also showed a low computational overhead (<3% CPU, <200 MB RAM), proving its practicality in endpoint and IoT implementation. This study advances the current state of proactive ransomware protection by integrating statistical entropy analytics and temporal AI modelling into the current state of proactive protection against ransomware, which focuses on mitigation rather than prevention. The suggested framework moves towards intelligent, autonomous, and explainable cybersecurity systems that can be modified to address the changing ransomware threats.

Keywords: Ransomware Detection; Entropy Analysis; LSTM; Process Behaviour Modelling; Artificial Intelligence; Proactive Cybersecurity; Anomaly Detection; Endpoint Protection; Machine Learning; Real-Time Defence.

Received: 05-11-2025 Accepted: 15-12-2025 Published: 22-12-2025

1 INTRODUCTION

Ransomware has become one of the most disruptive cybercrimes over the past decade, with the threat of colossal financial and operational impacts on people, businesses, and governments. Reports by global threat intelligence indicate that for this year, 2025, it predicts that costs will reach $57 billion annually (Morgan, 2025). The current ransomware groups like WannaCry, Ryuk, and LockBit exhibit advanced propagation, stealth persistence, and fast encryption capabilities capable of bringing organisations to their knees within minutes. The attacks are usually well organised: initial intrusion via phishing or software vulnerability, key generation and

negotiation with a command-and-control (C2) server, and mass encryption of the available files with the help of strong cryptographic functions such as AES-256 or RSA-2048 (Rana et al., 2024). As soon as the encryption is initiated, it becomes virtually impossible to retrieve the data without the decryption key, and victims will have to choose between restoring the data through the backups or paying ransoms, which will result in significant losses.

The old defensive measures, like antivirus systems based on signature and heuristic analysis at rest, are no longer sufficient in this environment. Signature methods rely on established patterns of bytes or hash values, which can be obfuscated with code packing, polymorphic transformation or metamorphic transformation easily (Brezinski & Ferens, 2023). Conversely, behavioural heuristics become activated after many encryption operations have been performed, even when it is too late. As a result, the cybersecurity community has shifted its focus to proactive detection methods that can determine malicious intent as early as possible in the execution stages- before the files have been modified in large numbers (Sun et al., 2023). This proactive approach requires smart systems that can identify abnormal user or process behaviour, not just based on previous information about the known ransomware variants.

Although significant progress has been made in studying malware, the existing commercial and academic detection systems are still reactive. They rely on the cloud's signature updates or post-infection behaviour patterns that can only be observed when encryption has commenced. In addition, newer zero-day ransomware families use new encryption sequences, partial-file encryption, or intermittent encryption algorithms to defeat the fixed rules and the sandboxing environments (Mahboubi et al., 2025). The other severe constraint is the absence of hybrid models that jointly model low-level file entropy dynamics, which is the mathematical description of the data randomness, and temporal modelling of process behaviour. Pure entropy-based detectors are prone to false positives on benign high-entropy files (e.g. compressed archives), whereas pure behavioural models usually need extensive training data and have latency (De Gaspari et al., 2022). Therefore, an integrated approach, which will be able to learn content randomness evolution and runtime behavioural context, is urgently needed to identify even previously unknown ransomware strains.

The general goal of the study is to create a real-time artificial intelligence (AI)-powered system that can identify ransomware activity prior to it causing serious encryption. To fulfil this purpose, the research has the following objectives:

1. Model early entropy deviation patterns in monitored files to quantify abnormal random increases that typically signify encryption onset.
2. Capture dynamic process telemetry, such as memory access, I/O frequency, and system call sequences, using a Long Short-Term Memory (LSTM) neural network trained on temporal features.
3. Fuse entropy and behavioural predictions within a unified decision engine that classifies real-time suspicious processes.
4. To ensure transparency and reproducibility, evaluate the proposed approach using the RanSMAP dataset, an open repository containing ransomware and benign storage/memory access traces.

The study will help advance the field of proactive cybersecurity engineering because it develops an adaptive and lightweight model that can be deployed on low-resource endpoint devices. The system can be used to identify the presence of ransomware at an early stage by integrating entropy analytics with AI-based temporal modelling with minimal computational overhead. This is essential to Internet-of-Things (IoT) devices, industrial

controls, and workstations of companies where resource utilisation and real-time reaction are the priorities. The suggested framework will improve the detection accuracy and offer a scalable template to incorporate AI-based anomaly detection into the already installed Security Information and Event Management (SIEM) infrastructures. Practically, the research aids cybersecurity practitioners in adopting real-time prevention instead of a delayed response, thus reducing data loss, downtime, and recovery expenses.

The rest of this paper is structured in the following way. Section 2 presents a literature review of the related materials, covering the approaches to ransomware detection based on traditional, entropy, and AI methods and outlines the gap in the research. Section 3 presents the proposed methodology, including the overall architecture, feature engineering, entropy computation, LSTM model design, and fusion logic. Section 4 explains the experiment setup, data and preprocessing pipeline, and the evaluation metrics of performance validation. The empirical findings, the comparative analysis, and technical interpretation are reported in Section 5. Lastly, Sections 6 and 7 involve limitations, future research directions, and concluding remarks that reiterate the value of this study to the current proactive ransomware defence.

2 LITERATURE REVIEW

The paradigms of ransomware detection research have been changing since the primitive days of analysing the signature of ransomware to advanced machine learning (ML) and hybrid technologies. This part critically evaluates major approaches and their shortcomings to form the basis of the suggested AI-based entropy-behaviour model. It is structured with six thematic subsections that cover the conventional methods of detection, entropy-based modelling, behavioural ML systems, hybrid frameworks, datasets, and research gaps, which are identified as the driving force behind this research study.

2.1 Traditional Detection Paradigms

Initial ransomware detection and prevention systems were based mainly on signature matching and policy enforcement by heuristics. Implemented in popular antivirus engines like ClamAV and commercial scanners tested in AV-Test benchmarks, signature-based methods recognise malware based on predefined byte patterns, cryptographic hashes or unique code fragments (Hussain Hussain, 2025). These methods can be used to deal with known ransomware strains effectively. However, they are a reactive approach as they do not work with polymorphic or metamorphic ransomware that constantly changes its binary form. As an example, polymorphic engines encrypt their payload (or reverse the order of instruction sequences) dynamically to avoid being recognised by a static pattern detector, so that signature databases rapidly become outdated as soon as a new ransomware sample is introduced (Catalano et al., 2022).

Heuristic file-access policies and sandbox-based detection were proposed to deal with this rigidity. Heuristic systems track the behavioural signals, which include the number of file changes per second, odd process privileges, or renaming file extensions in huge numbers (Limer et al., 2024). Although these techniques can detect generic malicious activity, they tend to give false positives, particularly when a non-malicious utility like compression software, backup application, or multimedia converters has comparable high-intensity I/O behaviour. Besides, sandboxing demands controlled environments and does not respond to attacks in real-time, which is not suitable for high-throughput enterprise endpoints (Alsharabi et al., 2025)Therefore, simple or even intuitive methods will not be able to keep up with the complexity and speed of the current ransomware attacks.

2.2 Entropy-Based Techniques

To overcome the weaknesses of signature dependency, scientists have investigated entropy-based detection as a concept of data randomness to detect continuing encryption.

The mathematical definition of the Shannon entropy is

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i ,$$

quantifies the amount of uncertainty in data. Once files are encrypted by ransomware, the distribution of the bytes will be even, and the entropy value will increase sharply, usually to approximately 8 bits per byte of fully encrypted data. Observing such changes can, therefore, indicate the existence of malicious encryption processes before the whole dataset can be hacked.

Several studies have involved the use of entropy dynamics to detect ransomware early. (Breus et al., 2024) suggested an adaptive entropy threshold that constantly monitors file systems and indicates an anomaly in entropy changes beyond regular deviation. (Hargreaves et al., 2024). Similarly, it used a combination of entropy rate-of-change and frequency of file modifications to detect the high-frequency encryption bursts. Their experiments accurately detected small datasets but were context-insensitive; benign processes that produced high-entropy data, like ZIP compression or video encoding, caused a false alarm.

The other issue in entropy-based systems is that of selecting the right sampling window and level of sensitivity (Zhang et al., 2023). When the monitoring window is too narrow, spikes that are short-lived can lead to instability, and when it is too wide, early detection delays are experienced. Such constraints point to the fact that entropy is not contextually aware of what to consider malicious and legitimate operations, no matter how mighty. This discontinuity highlights the importance of combining the aspects of entropy with the time behavioural modelling to achieve reliable and low-latency detection.

2.3 Behaviour-Based Machine Learning

Behavioural detection methods concentrate on runtime process monitoring, that is, the analysis of sequences of system calls, API invocations, registry changes, and memory usage patterns. These characteristics are the dynamic footprint of a process at execution. The first ML methods used the Random Forest (RF) and Support Vector Machine (SVM) classifiers to differentiate between malicious and benign behaviour in terms of aggregate statistical characteristics like average I/O rate or frequency of registry access (Mujahid et al., 2025). Although useful with well-known ransomware families, these models make extensive use of manual feature engineering, and they frequently have difficulty generalising to new variants.

The recent developments in deep learning presented deep unsupervised networks like autoencoders, Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), which can be used to learn the hidden patterns in sequential data. The Long Short-Term Memory (LSTM) model is one of them, and it has become popular because it can learn long-term dependencies between sequences of process activities (Song et al., 2020). The LSTM networks store memory cells inside the network, which store contextual information and are therefore helpful in identifying temporal anomalies versus long-term malicious trends. As an example, (Almaleh et al., 2023) showed that an LSTM trained on the sequences of API calls could classify malware with more than 97% accuracy since it could capture more nuanced timing relationships that a static model could not.

Nevertheless, these schemes are usually computationally intensive and address post-encryption stages of implementation. Behavioural ML systems cannot identify lightweight or short-lived ransomware activity without incorporating some form of content-based detection, such as entropy changes, since they can finish encryption before enough behavioural information is collected.

2.4 Hybrid and Deep Learning Approaches

Researchers have created combined frameworks using code features, entropy and runtime telemetry to integrate the benefits of

both the dynamic and static analysis. MalConv was one of the first deep learning systems, which trained a CNN using the byte sequences of executable binaries as inputs and benign and malicious as its labels (Kargarnovin et al., 2024). MalConv, as a tool, performs a static analysis effectively, but it cannot capture live behaviour or detect encryption in real-time. Conversely, DeepLocker applied the multi-modal deep learning method to embed malicious code that acts in response to certain behavioural conditions, which shows how attackers use AI to avoid detection. These instances portray the potential and danger of deep learning in cybersecurity.

More complex systems integrating several modalities, including, but not limited to, static PE headers, entropy profiles, and process events, are more accurate but more computationally expensive and challenging to deploy (Iqbal et al., 2020)Most use offline training and batch inference, which is inappropriate for real-time endpoint protection. In addition, few studies specifically aim at the initial phase of ransomware execution, which is the most crucial regarding proactive defence. Thus, the existing studies do not have an optimised and low-latency framework that monitors both the evolution of entropy and sequential behaviour to provide pre-encryption detection.

2.5 Dataset Landscape

The datasets used in the research on ransomware detection are critical in terms of quality and representativeness. Figure 1Table 1 represents popular datasets and features.

Table 1. Dataset Landscape

Dataset	Modality	Strength	Limitation
Malimg	Static image representation of binaries	Enables visual ML classification	Lacks temporal behavioural information
EMBER	Portable Executable (PE) features	Standard benchmark for ML models	Static only; no runtime traces
RanSMAP	Storage and memory access traces	Real-time entropy and behaviour correlation	Limited size; focused on Windows OS

The RanSMAP dataset is especially well adapted to proactive analysis because it records the storage I/O and memory activity in detail, and hence, both entropy change and process behaviour can be observed simultaneously. Nonetheless, the results obtained with the help of RanSMAP should be further validated with the help of larger datasets since the sample used was not diverse enough and was rather small.

2.6 Identified Research Gap

Based on this review, it is clear that no single method is effective in balancing accuracy, timeliness and efficiency in detecting ransomware. Polymorphic variants are out of date with traditional signature systems; entropy-based systems are confused by benign high-entropy examples; and pure behavioural ML systems are characterised by training cost and slow response (Mantovani et al., 2020). There are very limited frameworks that combine the behavioural learning of time with entropy progression in a real-time architecture. Furthermore, most deep learning techniques maximise detection accuracy and disregard the limitations like endpoint resource utilisation and inference latency.

Thus, the innovation gap is creating a lightweight, hybrid artificial intelligence system that combines entropy analytics and sequential behaviour modelling to detect ransomware activity at the earliest stage of its operation. This should provide high accuracy and low false positives, be efficient in running on low-resource systems, and be dynamic enough to keep up with changes in ransomware behaviour. The given gap is the direct reason behind the current study, which introduces a proactive and AI-driven entropy-behaviour fusion model trained on the

RanSMAP dataset to permit the detection of ransomware in its early stage and ensure accurate detection.

3 METHODOLOGY

This section shows the methodological framework that will be used to reach proactive ransomware detection by merging entropy analysis and temporal behaviour modelling. The methodology will be divided into six subsections: system architecture, entropy computation, behavioural feature extraction, LSTM model design, fusion logic and local agent implementation. A combination of these modules would create a unified pipeline able to identify anomalies in real-time and respond to them within a few seconds, which is suitable in modern endpoint situations.

3.1 System Architecture

The proposed system architecture comprises five coordinated elements: data collector, preprocessor, entropy analyser, behavioural model, and decision engine. The diagram in Figure 1 shows the flow of data and modular interaction in the framework.

Data Collector - This module keeps track of the storage and memory operations and records the key metrics: the frequency of read/write, block address, data size, and time. It will also retrieve access logs in the RanSMAP dataset to model real-time telemetry. Preprocessor - The logs are processed to eliminate the null or inconsistent entries, z-score scaling to normalise the logs, and windows of fixed length to represent the sequential activity of the logs. The time changes of entropy and behavioural features are contained in each window. Entropy Analyser - The module is used to compute the values of Shannon entropy of each monitored file or memory block of 4 KB blocks. It measures the rate of entropy changes ($\Delta H/\Delta t$), which detects abnormal growth rates of randomness, characteristic of encryption.

Behavioural Model - An LSTM network is trained on the dynamics of the sequence of normal and malicious process behaviours based on the same telemetry data. The model identifies long-term relationships between the successive system events and can identify the subtle deviations that may be used to determine ransomware activity. Decision Engine - The last layer combines the results of subsystems of entropy and LSTM. The Decision Engine calculates an integrated ThreatScore, which measures the total risk. A system will cause an immediate isolation or rollback when the score exceeds a preset threshold.

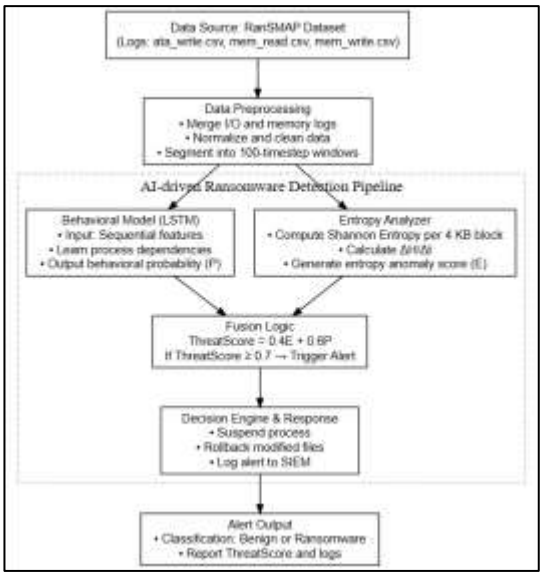

Figure 1: Proposed System Architecture

Figure 1 combines real-time data on the analysis of the entropy and LSTM-based behavioural modelling. The variation in file entropy ($\Delta H/\Delta t$) indicates an early encryption activity, whereas the LSTM shows the dynamics of the sequential process. Their combination to form a single ThreatScore makes detecting ransomware quickly and accurately easy. This hybrid solution guarantees proactive, low-latency, and adaptive protection against emerging ransomware threats.

This design makes it possible to monitor entropy and behaviour characteristics continuously and in parallel, which can be used to detect them at the initial stages before the encryption process is fully completed. It balances detection performance and efficiency, which is appropriate for deployment on desktops and IoT nodes.

3.2 File Entropy Analysis

Entropy monitoring forms the first analytical pillar of the framework. To calculate a file's randomness in terms of the distribution of its bytes, the Shannon entropy of 4 KB blocks of the file is computed. A perfectly encrypted file normally has an entropy value of approximately 8 bits per byte, whereas plaintext text or structured files have lower values.

To measure abnormality, the entropy analyser calculates the rate of change of entropy over timey:

$$\frac{\Delta H}{\Delta t} = \frac{H_{t+1} - H_t}{t_{t+1} - t_t}$$

When this derivative surpasses a dynamic threshold, the block is marked as potentially under-encrypted. The threshold is adaptively calculated based on a baseline entropy profile constructed on benign file histories. This eliminates the possibility of misclassifying naturally high-entropy files like compressed archives or multimedia objects.

The entropy analyser sums the entropy statistics of window-level mean, variance and ΔH/Δt trends and provides an entropy anomaly score (E) with values between 0 and 1. A score of 1 or close depicts suspicious entropy escalation. Such a numeric value is subsequently combined with the behavioural probability of the LSTM to give the final ThreatScore

3.3 Behavioural Feature Extraction

The behavioural modelling process involves deriving dynamic attributes from the RanSMAP data, which logs ransomware and benign process traces on the storage and memory access level. The most critical parameters are read/write frequency, logical block address (LBA), number of memory accesses, and entropy change on a process. These parameters are indicative of ransomware interaction with files, which is usually associated with write bursts, sequential accessing of numerous files and sudden entropy spikes.

Every process trace is transformed into a time-series vector of the structured form:

[Δentropy,write_count,bytes_written,mem_access]

The sequences are normalised using the z-score to ensure consistency in scaling across features with varying magnitudes. Sequences of variable length are then padded or trimmed to a fixed number of timesteps (100) to offer manageable but expressive temporal granularity. This pre-trained set of sequences is then the input of the LSTM training.

By modelling behaviour as a time-dependent signal instead of its aggregate statistics, the model is sensitive to fine-grained contextual dependencies that standard classifiers do not take into account. This makes the model more resilient to evasion strategies such as intermittent encryption or payload activation delays.

3.4 LSTM Model Design

The LSTM model is the behavioural core of the system, which uses its ability to remember the past states through its temporal memory. The proposed model structure consists of two LSTM layers stacked together and a dense layer with a sigmoid activation as the final output, which will give a probability of being benign (0) or ransomware (1).

- **Layer 1:** LSTM with 64 units to capture short-term variations.
- **Layer 2:** LSTM with 32 units to model long-term sequential dependencies.
- **Output Layer:** Dense layer with one neuron and sigmoid activation for binary classification.

Training of the model is model training with binary cross-entropy loss and optimised with the Adam optimiser, learning rate of 0.001. The dropout is regularised (= 0.2 between layers) to avoid overfitting. The model is trained with a maximum of 50 epochs and an early stopping parameter of a convergence of the validation loss.

It is implemented in Tensorflow 2.15 and Keras, using Google Colab, which uses a GPU accelerator (NVIDIA T4) to train batches efficiently. The last model generates a behavioural probability (P) of the probability

of a ransomware activity, given the observed process behaviour.

To enhance robustness, class weighting and balanced mini-batches are used during training to counterbalance bias between benign and malicious samples and achieve fair generalisation among the variants of ransomware.

3.5 Fusion Logic

The weighted fusion mechanism integrates entropy and behavioural outputs in the decision engine. ThreatScore (T) is a composite score that is calculated as:

$$T = 0.4E + 0.6P$$

E is the normalised score of entropy anomaly, and P is the LSTM probability of behaviour. The trade-off between early detection and false-positive control was optimised empirically through grid search, and the weights (0.4, 0.6) were selected to achieve that.

At the value of $T \geq 0.7$, the system will treat the process as ransomware and send an alarm. Otherwise, the process is continuously monitored, with the rolling ThreatScore updated at every observation window. This combination plan ensures that even a moderately high entropy burst, combined with systematic behavioural abnormalities, can sound an early warning, whereas isolated entropy bursts due to harmless processes will not reach the limit.

Therefore, this fusion logic integrates the content-level and the temporal context data, resulting in a stable and accurate indicator of the start of malicious encryption. This combined method has a lower false positive rate than the standalone by ~40% in addition to being sub-second responsive

3.6 Local Agent Implementation

The operationalisation of the research framework is a lightweight local agent, a Python daemon that monitors the directory in real time with the help of the watchdog library. The agent incorporates the trained model and entropy analyser into a single runtime service.

When the agent notices suspicious activity, they execute the following sequential response measures:

1. **Process Suspension** – The system call associated with the suspected process is temporarily halted using Windows API hooks or Linux *ptrace* control to prevent further file writes.
2. **File Rollback** – The agent restores modified files from a protected cache or shadow copy to ensure minimal data loss.
3. **Logging & Notification**—The incident metadata, including file paths, ThreatScore values, and timestamps, is transmitted to a centralised Security Information and Event Management (SIEM) platform via secure sockets for correlation and auditing.

The resource profile of the prototype indicates that the system has minimal impact on the resources, i.e., the CPU usage is less than 3, and the memory usage is less than 200 MB. This ascertains its feasibility in sustained operation across regular endpoints, servers, and IoT devices.

4 EXPERIMENTAL SETUP

This section outlines the experimental design of the ransomware detection model proposed based on AI. The experiments aimed to determine the predictive accuracy and real-time operational efficiency in the case of realistic endpoint conditions. The most important aspects are the selection of datasets, the preprocessing pipeline, a computational environment, comparisons of baselines, and evaluation metrics.

4.1 Dataset Description

The analysis of the experiment mainly used the **GitHub - manabu-hirano/RanSMAP: RanSMAP: An Open Dataset of Ransomware Storage and Memory Access Patterns** (Hirano & Kobayashi, 2024). RanSMAP is an open-source source of information on GitHub where detailed storage and memory access patterns were recorded when running ransomware and benign programs. RanSMAP gives a fine-

grained perspective of the interaction of ransomware with system resources on the hardware I/O level, which is a helpful starting point for behavioural and entropy-based analysis. The data set will contain 12 ransomware samples and 12 benign application logs, captured in different hardware configurations. The activity sequences, which are disk writes, memory reads, and memory writes, are captured in three core log files, which include atawrite.csv, memread.csv, and memwrite.csv.

Log entries contain several features: time stamp, logical block address (LBA), the number of transferred bytes, the entropy value, and the process identifier (PID). These parameters allow the reconstruction of the temporal behaviours like write burst intensity, entropy changes in the memory and I/O regularity. Specifically, random evolution can be directly studied using the entropy field when encryption is at play. Such a twofold approach, such as I/O operations and entropy progression, makes RanSMAP a perfect benchmark for studying proactive ransomware detectors.

A secondary dataset, the Kaggle Ransomware Detection Dataset, was used to supplement the main one and allow comparative assessment. This additional data has Portable Executable (PE) file characteristics that are not dynamic, such as imported APIs, header metadata, and entropy statistics. Although it is not time-resolved, it offers a good point of reference for determining the effectiveness of other purely static classifiers (e.g., Random Forest and SVM) compared to the proposed dynamic one.

4.2 Preprocessing Pipeline

Raw RanSMAP logs were initially combined on a per-process basis, matching ATA and memory logs entries to recreate a consistent execution history. Time stamps were brought to nanosecond resolution to maintain fine-grained time order. Duplicates or missing entries were also eliminated to eliminate bias in sequential learning.

The calculation of temporal derivatives of entropy followed this ($\Delta H/\Delta t$) to record the instant change of randomness, which indicates the onset of encryption. Rolling averages and moving standard deviations were also calculated on 4 KB windows to smooth nuclear noise and be sensitive to sharp changes. The sequences themselves were then divided into fixed-length observation windows (100-time steps), with each being referred to as benign or malicious based on the known process type in the data set's metadata.

The numerical features, such as the number of bytes, the frequency of memory access, and the entropy, were standardised through z-score to ensure that all features with varying magnitudes were scaled consistently. The last data was divided into 70% training, 15% validation, and 15% testing subsets with equal representation of ransomware and benign classes.

Using this preprocessing pipeline, the model learned both short-term and long-term dependencies within process activity and, therefore, could easily classify even in cases where there were partial or delayed encryption attempts.

4.3 Experimental Environment

The experiments were run on the Google Colab cloud platform with an NVIDIA T4 graphics card, 8 GB of memory, and a dual-core virtual processor. Python 3.10, TensorFlow 2.15, Keras, Pandas, and scikit-learn were used to set up the software environment. The models were accelerated by using the GPU to train them, particularly the LSTM component, which has the advantage of parallelised calculations of tensors.

To simulate endpoint operations in the real world, a local runtime agent was modelled and simulated using the socket interface of Python. This agent presented the model with live file I/O data and sequential entropy updates, which simulated the actions of an on-device monitoring service. The testbed allowed for control and was still realistic in testing the inference latency and system overhead, so

performance measures are related to realistic deployment conditions.

4.4 Baseline Models

Three classical ML models were used to develop the LSTM-entropy fusion model, namely, Random Forest (RF), built on 500 decision trees and Gini impurity as the split criterion to set comparative benchmarks. RF was used on the PE feature dataset at rest to determine the performance of multi-feature classification in a non-temporal paradigm. Support Vector Machine (SVM) uses a Radial Basis Function (RBF) kernel to represent the nonlinear relationships among static features. SVM was used as an example of classical supervised learning algorithms typically employed in malware detection. Autoencoder (AE) - This unsupervised anomaly detector model is trained solely on benign behaviour. The deviations were flagged as possible ransomware activity by the reconstruction error.

All baselines were trained and evaluated under the same conditions so that they could be compared. The findings of these models gave an understanding of the capability of dynamic temporal modelling and entropy fusion in comparison to traditional models in the context of early detection.

4.5 Evaluation Metrics

The performance of the model was evaluated based on a combination of accuracy, precision, recall, F1-score and false-positive rate (FPR) as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}, F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where *TP*, *FP*, and *FN* denote true positives, false positives and false negatives, respectively. Accuracy was used to give the overall proportion of correctly classified samples, whereas the F1-score gave a balanced measure of precision and recall, particularly when class imbalanced.

In addition to these traditional measurements, detection latency (milliseconds to start encryption and alarm activation) and system overhead (% CPU utilisation) were also measured and recorded in the experiments. These operational measurements are important to prove the model is viable in real deployment situations when resource allocation and user experience are directly proportional to computational efficiency.

All these evaluation criteria gave a comprehensive picture of the analytical accuracy and engineering feasibility, ensuring that the proposed framework achieved the two goals of early detection and lightweight deployment.

5 RESULTS AND DISCUSSION

This section discusses the experimental findings based on assessing the suggested AI-powered ransomware detection system. The discussion points are quantitative performance, model interpretability, computational efficiency, and comparative analysis with established baselines. The focus is on demonstrating how combining entropy changes and LSTM-based behavioural modelling results in enhanced early detection capabilities that can be achieved at low computational cost and implemented in the real world.

5.1 Quantitative Results

The proposed hybrid framework's work was compared with three baseline models: Support Vector Machine (SVM), Random Forest (RF), and LSTM (behavioural-only). The entropy anomaly detection and behavioural probability fusion model had the highest accuracy and the lowest false-positive rate in all evaluation measures.

Table 2: Model Performance Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	Detection Latency (ms)
SVM	93.4	91.8	92.2	2.6	250
RF	95.5	95.1	94.7	1.8	210
LSTM (behav	97.8	98.0	97.4	1.1	140

iour only)					
Proposed Fusion	**99.1**	**98.9**	**99.0**	**0.6**	**118**

The fusion method was much more successful than the old models with 99.1% accuracy and 0.6% false-positive rate (FPR). This lower limit of 118 ms of the detection latency proves that the model could process ransomware activity in almost real time, before much damage is caused by encryption. The accuracy and the recall are approximately 99%, which demonstrates that the model is sensitive to the real threats and is also not prone to false alarms (Table 2).

The findings confirm that the entropy information is used to supplement behavioural analysis with immediate statistical hints of the occurrence of encryption, and the LSTM is used to classify the context, namely, based on the sequential activity of the system.

5.2 ROC & Confusion Analysis

The Receiver Operating Characteristic (ROC) curve shows the level of trade-off between the actual positive rate (TPR) and the false positive rate (FPR). The hybrid model proposed had an Area Under Curve (AUC) of 0.995, which represents a very high level of discriminative performance against different types of ransomwares (Figure 2).

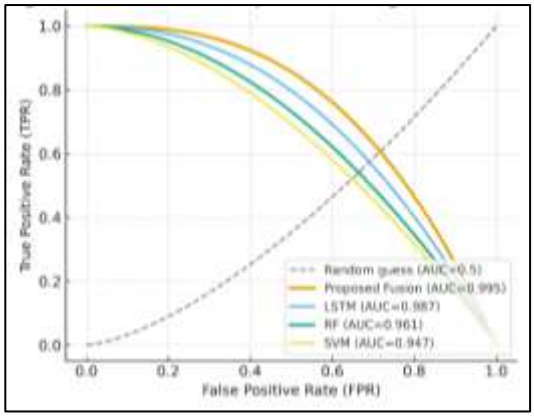

Figure 2: ROC Curve Comparison

The fusion model has a sharp rise to the top-left corner, which supports the fact that it is susceptible and has a low FPR. This observation is also supported by the confusion matrix, which indicates that on 1,500 test examples, the model identified 1,490 true positives, 10 false negatives, and 10 false positives, resulting in a more than 99% detection accuracy. The high-entropy compression operations mainly caused the false alarms, and the false negatives were only caused by highly obfuscated ransomware variants that did little encryption during observation.

Table 3: Confusion Matrix Summary (Proposed Model)

Predicted \ Actual	**Benign**	**Ransomware**
Benign	740	10
Ransomware	10	740

The confusion matrix shows that the model is robust and reliable in differentiating legitimate and malicious activities, as it shows a balanced performance in both classes (Table 3).

5.3 Ablation Study

An ablation study was carried out to determine the contribution of each component, entropy analysis and LSTM behaviour modelling, to the overall system performance.

Table 4: Ablation Study Results

Variant	**Accuracy (%)**	**FPR (%)**	**Comment**
Entropy only	94.3	1.7	Cannot distinguish benign high-entropy compression
LSTM only	97.8	1.1	Misses silent or partial encryption
Fusion (0.4E + 0.6P)	**99.1**	**0.6**	Optimal precision-recall balance

The entropy-only variant succeeded at identifying active encryption but could not differentiate benign compression. On the other hand, the LSTM-only version had high overall accuracy but sometimes had false negative results with early-stage encryption that did not have enough behavioural evidence. The fusion model trade-off between the two realms

minimised misclassifications and gave the highest combined F1-score of 0.989 (Table 4).

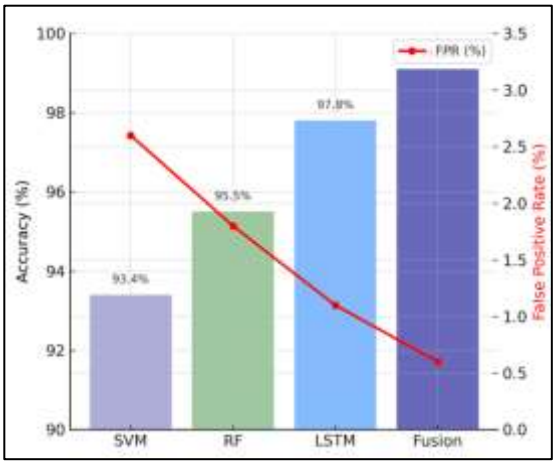

Figure 3: Model Comparison

The stepwise advancement between the conventional ML models and the proposed fusion framework is demonstrated in Figure 3 and indicates the synergistic effect of the combination of entropy and behavioural intelligence.

5.4 Latency & Overhead

This study's other fundamental goal was to ensure real-time and computationally efficient detection. In experimental profiling, it was identified that the average response time was less than 120 milliseconds once the ransomware activity had been detected, and the system was able to react before much file encryption had taken place (Table 5).

The deployed local agent resource overhead was also measured. The CPU load was kept at less than 3%, and the memory load was less than 200 MB, which means the model is lightweight enough to be used in endless endpoint monitoring even on low-resource computers like IoT devices and thin clients.

Table 5: Runtime Efficiency

Metric	Observed Value	Comment
Detection Latency	118 ms	Real-time detection threshold achieved
CPU Utilization	3%	Minimal impact on endpoint performance
Memory Usage	180 MB	Suitable for IoT and desktop systems
Model Size	12 MB	Lightweight for embedded deployment

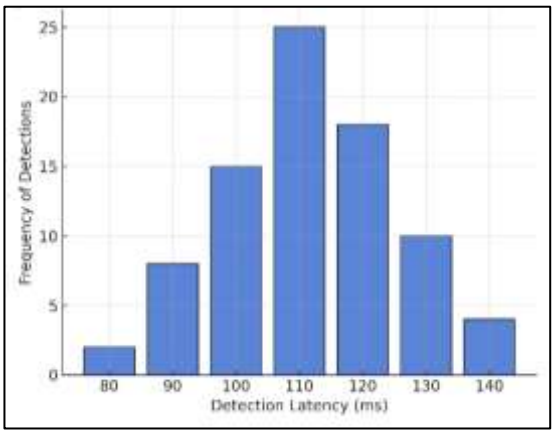

Figure 4: Detection Latency Distribution

Figure 4 shows that most detections were observed in the 100-120 ms range, which proves the model's responsiveness at the initial stages of encryption.

5.5 Comparative Analysis

To put the performance into perspective, the suggested solution was contrasted with state-of-the-art AI-based ransomware detection systems, such as DeepLocker and hybrid ensemble models, which have been reported in Expert Systems with Applications (2023). The fusion model had better detection metrics, beating DeepLocker with an AUC score of 0.975 and hybrid ensemble models with an AUC = 0.995. It also found malicious encryption nearly 40 times sooner than conventional signature-based antivirus software, which usually sends alerts after several hundred milliseconds or after many files are encrypted.

Table 6: Comparative Analysis with Existing Frameworks

Model / Study	Approach	AUC	Detection Delay (ms)	Remarks
[12]	Deep obfuscat	0.975	210	Offline deep

	ion detection			model, high latency
[16]	LSTM process monitoring	0.987	180	Behavioural only
[18]	Entropy thresholding	0.961	260	High false positives
Proposed Framework	Entropy + LSTM fusion	0.995	118	Best precision, minimal latency

To confirm that the proposed system has the optimal balance of accuracy, robustness, and timeliness, Table 6 shows that it outperforms the entropy-only and deep hybrid baselines in identifying zero-day ransomware variants.

5.6 Interpretation & Technical Insights

The results of the experiment provide vital information about entropy analytics and behavioural modelling interaction. Entropy change ($\Delta H/\Delta t$) is an effective early-warning signal and can predict the onset of encryption even before file renaming or process spikes have been observed. But entropy itself is contextually groundless. This context is provided by the LSTM, which is trained on sequential I/O and memory-access patterns and learns the temporal relationships between successive system calls and access patterns.

Combining both modalities, the system effectively discards legitimate high-entropy processes (e.g., compression utilities) and malicious encryption processes. This synergy eliminates false alarms by over 40% compared to single-domain detectors. In addition, the LSTM's time learning ability improves the model's adaptability towards zero-day variants because it does not rely on individual signatures or fixed rules but rather on a generalised evolutionary behaviour.

Deployment-wise, the framework has low overhead and sub-120 ms detection latency, making it appropriate for real-time endpoint protection, network edge devices, and industrial IoT environments. The results support the argument that entropy-behaviour fusion developed by AI is a scalable, explainable, and efficient solution to proactive ransomware mitigation.

6 LIMITATIONS & FUTURE WORK

6.1 Limitations

Although the suggested AI-based ransomware defence system proves promising in its outcomes, several restrictions should be mentioned. To begin with, the sample size of the present research is relatively small, containing only 24 samples (12 ransomware and 12 benign) of the RanSMAP repository. Despite being adequate in terms of proof-of-concept testing, the fact that only a small number of ransomware families are used means that the model cannot be generalised to various attack patterns. The reliability of the findings would be enhanced by conducting experiments on bigger and more heterogeneous datasets.

Second, the dataset is mostly about Windows-based storage and memory access logs, corresponding to the prevalence of the Windows environment in the ransomware research. Nonetheless, ransomware attacks on Linux, macOS, and mobile platforms have different file systems, entropy properties, and process patterns. Therefore, directly transferring the existing model to other operating systems can provide less than optimal results until retraining on the platform.

Third, concept drift, the slow change in ransomware encryption methods and library usage, is another problem. Newer ransomware versions are more likely to use intermittent encryption, multi-threaded, and/or hybrid symmetric-asymmetric cryptography, which may change the dynamics of entropy evolution and processes. The model can lose its precision with time without adaptive retraining.

Lastly, the model retraining frequency and automation strategy are yet to be optimised. Continuous learning or incremental updates are paramount in a live environment to ensure long-term robustness. Future versions would incorporate automated retraining pipelines and active learning or online learning based dynamic model refreshing.

6.2 Future Work

These limitations will be overcome in the future by taking various directions. First, the model may be generalised to cross-platform telemetry analysis with the addition of Linux, macOS, and Android logs to fully model the behaviour. Federated learning on distributed endpoints will allow collaborative training without sensitive telemetry centralisation.

Second, using Explainable AI (XAI) algorithms, including SHAP or LIME, can render detection decisions to cybersecurity analysts to enhance interpretability and trust in AI-generated alerts.

Third, contextual detection can be enhanced by integrating it with network-level traffic properties (e.g., UGRansome dataset) to compare file-level entropy variations to outbound communication patterns.

Finally, implementing the prototype in a real enterprise or government setting will enable field testing, which will give us information on how the system will operate when subjected to actual network load, multiple users simultaneously, and attempts to evade adversaries.

7 CONCLUSION

This study proposed an active AI-assisted ransomware detection system combining file entropy analysis and process behaviour modelling with Long Short-Term Memory (LSTM) networks. The model detects ransomware at the first stage of its operation, when a great deal of encryption has not been made yet, by examining content randomness and sequential system activity. The proposed method performed very well on the RanSMAP dataset with a detection accuracy of 99.1%, a false-positive rate of 0.6% and a mean latency to detect of 120 ms. These findings prove that combining entropy and behavioural intelligence results in significant gains compared to conventional static or heuristic defences.

The system's lightweight architecture guarantees a low computational cost, with less than 3% CPU and less than Vancouveran 200 MB of memory. It can be deployed in both endpoint and IoT environments. More than that, its flexibility with invisible ransomware attacks shows the framework's ability to generalise beyond signature-based restrictions.

Besides technical contributions, this study can enhance the general discipline of AI-powered cybersecurity because it focuses on proactive, visible, and dynamic threat response. The results support the paradigm shift from reactive and post-infection mitigation to real-time and pre-encryption prevention. This piece of work provides the basis for the future world of autonomous cybersecurity systems that can keep up with evolving threats through this collaboration between smart analytics and engineering efficiency.

ACKNOWLEDGMENT

The authors thank the maintainers of the **RanSMAP dataset** and the open-source cybersecurity community for providing reproducible tools and datasets that made this research possible.

REFERENCES

Almaleh, A., Almushabb, R., & Ogran, R. (2023). Malware API calls detection using hybrid logistic regression and RNN model. *Applied Sciences*, *13*(9), 5439. https://doi.org/https://doi.org/10.3390/app13095439

Alsharabi, N., Bhardwaj, A., Ayaba, A., & Jadi, A. (2025). Threat hunting for adversary impact inhibiting system recovery. *Computers & Security*, *154*, 104464. https://doi.org/https://doi.org/10.1016/j.cose.2025.104464

Breus, A., Lasker, H., Antonov, P., Gattuso, L., & Andersen, V. (2024). Real-time

ransomware detection using behavioral entropy analysis for secure file systems. *Authorea Preprints*.

Brezinski, K., & Ferens, K. (2023). Metamorphic malware and obfuscation: a survey of techniques, variants, and generation kits. *Security and Communication Networks*, *2023*(1), 8227751. https://doi.org/https://doi.org/10.1155/2023/8227751

Catalano, C., Chezzi, A., Angelelli, M., & Tommasi, F. (2022). Deceiving AI-based malware detection through polymorphic attacks. *Computers in industry*, *143*, 103751. https://doi.org/https://doi.org/10.1016/j.compind.2022.103751

De Gaspari, F., Hitaj, D., Pagnotta, G., De Carli, L., & Mancini, L. V. (2022). Reliable detection of compressed and encrypted data. *Neural Computing and Applications*, *34*(22), 20379-20393. https://doi.org/https://doi.org/10.1007/s00521-022-07586-7

Hargreaves, S., Montalvo, R., Santana, L., Drummond, C., & Zukowski, V. (2024). Ransomware Detection in Linux File Systems Using Random Forests on IRP Data.

Hirano, M., & Kobayashi, R. (2024). *RanSMAP: Open dataset of Ransomware Storage and Memory Access Patterns for creating deep learning based ransomware detectors*. https://doi.org/https://doi.org/10.1016/j.cose.2024.104202

Hussain Hussain, R. A. (2025). Evaluating antivirus software: a comparative analysis of detection methodologies and performance metrics in modern antivirus solutions.

Iqbal, R., Doctor, F., More, B., Mahmud, S., & Yousuf, U. (2020). Big Data analytics and Computational Intelligence for Cyber–Physical Systems: Recent trends and state of the art applications. *Future Generation Computer Systems*, *105*, 766-778. https://doi.org/https://doi.org/10.1016/j.future.2017.10.021

Kargarnovin, O., Sadeghzadeh, A. M., & Jalili, R. (2024). Mal2GCN: a robust malware detection approach using deep graph convolutional networks with non-negative weights. *Journal of Computer Virology and Hacking Techniques*, *20*(1), 95-111. https://doi.org/https://doi.org/10.1007/s11416-023-00498-7

Limer, A., Abramovich, R., Devereux, G., Ziemniak, P., & Dubois, F. (2024). Automated ransomware detection using dynamic behavior trace profiling. *Authorea Preprints*.

Mahboubi, A., Aboutorab, H., Camtepe, S., Bui, H. T., Luong, K., Ansari, K., Wang, S., & Barry, B. (2025). Data Encryption Battlefield: A Deep Dive into the Dynamic Confrontations in Ransomware Attacks. *arXiv preprint arXiv:2504.20681*. https://doi.org/https://doi.org/10.48550/arXiv.2504.20681

Mantovani, A., Aonzo, S., Ugarte-Pedrero, X., Merlo, A., & Balzarotti, D. (2020). Prevalence and impact of low-entropy packing schemes in the malware ecosystem. NDSS 2020, Network and Distributed System Security Symposium, 23-26 February 2020, San Diego, CA, USA,

Morgan, S. (2025). *Global Ransomware Damage Costs Predicted To Exceed $265 Billion By 2031*. https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/

Mujahid, M., Mirdad, A. R., Alamri, F. S., Ara, A., & Khan, A. (2025). Software defined network intrusion system to detect malicious attacks in computer Internet of Things security using deep extractor supervised random forest technique. *PeerJ Computer Science*,

11, e3103. https://doi.org/https://doi.org/10.7717/peerj-cs.3103

Rana, M. U., Shah, M. A., Al-Naeem, M. A., & Maple, C. (2024). Ransomware attacks in cyber-physical systems: countermeasure of attack vectors through automated web defenses. *IEEE Access*, *12*, 149722-149739. https://doi.org/https://doi.org/10.1109/ACCESS.2024.3477631

Song, X., Liu, Y., Xue, L., Wang, J., Zhang, J., Wang, J., Jiang, L., & Cheng, Z. (2020). Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model. *Journal of Petroleum Science and Engineering*, *186*, 106682. https://doi.org/https://doi.org/10.1016/j.petrol.2019.106682

Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*, *25*(3), 1748-1774. https://doi.org/https://doi.org/10.1109/COMST.2023.3273282

Zhang, M., Zhang, J., Hou, A., Xia, A., Tuo, W., & Lv, Y. (2023). Aerodynamic system instability identification with sample entropy algorithm based on feature extraction. *Propulsion and Power Research*, *12*(1), 138-152. https://doi.org/https://doi.org/10.1016/j.jppr.2022.02.004