**OPEN ACCESS**

Volume: 13

Special Issue: 1

Month: February

Year: 2026

P-ISSN: 2321-4643

E-ISSN: 2581-9402

Citation:
Vimala, B "An Explainable Hybrid Attention–Temporal Ensemble Framework For Early Ransomware Detection in Hospitality Information Systems Using Kernel-Level File I/O Behaviour." *Shanlax International Journal of Management*, vol. 13, no. S1, 2026, pp. 59–72.

DOI:
https://doi.org/10.34293/management.v13iS1-i1-Feb.10332

# An Explainable Hybrid Attention–Temporal Ensemble Framework For Early Ransomware Detection in Hospitality Information Systems Using Kernel-Level File I/O Behaviour

**Dr. B. Vimala**
*Teaching Assistant, Department of Tourism and Hotel Management*
*Alagappa University, Karaikudi*

**Abstract**

*Ransomware attacks increasingly threaten hospitality information systems such as property management software, booking platforms, payment gateways, and guest data repositories, disrupting operations and compromising sensitive information before traditional defenses respond. This study proposes an explainable Hybrid Attention–Temporal Ensemble (HATE) framework for early ransomware detection tailored to hotel IT environments. Kernel-level file I/O events are captured using Windows Event Tracing (ETW) within a sandbox simulating realistic hospitality workloads. The logs are pre-processed and encoded into behavioural features, then analysed using three complementary deep learning models: a Temporal Fusion Transformer for time-dependent patterns, a Graph Attention Network for relational event structures, and an Informer Transformer for long behavioural sequences. Soft voting combines predictions into a final ransomware probability score. SHAP, LIME, and Integrated Gradients provide global and local interpretability. Experiments show 97.8% detection, 98.1% recall, and a 2.1% false positive rate, identifying r*

**Keywords: Ransomware Detection, Hospitality Information Systems Security, Kernel-Level Behaviour, Explainable AI, Early-Stage Threat Detection, ETW Monitoring.**

## Introduction

Ransomware refers to a malicious software that intends to deny access to a user's data or system by encrypting files and then requiring a ransom to be sent to be decrypted [1-2]. It is essential to identify ransomware at its initial stage, as it is possible to prevent the loss of numerous data or other disturbance of the system effectively. A runtime analysis, especially the patterns of storage access (e.g., file read and write operations, access sequence, logical block addressing and entropy change) is one of the most practical methods of detecting ransomware at the first stages [3-4]. These relation (process-to-file

interactions) and time (sequence and timing of operations) structures can often indicate malicious intent in advance of large-scale encryption, allowing proactive defensive actions to be undertaken [6-8]. Conventional detection mechanisms, such as signature-based mechanisms, analysis of the static files as well as rule-based detection mechanisms, have severe drawbacks. They find it difficult to detect new, polymorphous or zero-day ransomware with different signatures [9-10]. Models built with machine learning can either deal with temporal sequences or relational interactions but not both, which makes it hard to achieve the complete range of ransomware behaviors, leading to false positives or slow response time. Also, most of the available models are not robust to noisy and heterogeneous data and cannot offer interpretable explanations of their predictions, making them less useful and trusted in the real world [11-14]. Such issues demonstrated the necessity of combining methods through hybridization and making use of advanced learning methods and off

**Key Contribution**

- Introduces a new HATE architecture that incorporates the Temporal Fusion Transformer, Graph Attention Network, and Informer Transformer as the means of detecting ransomware promptly and accurately.
- Presents a kernel-level behavioural data acquisition model based on Windows ETW, which captures finer-grained File I/O events, making it possible to model ransomware encryption behaviour realistically.
- Alphabetic encoding and feature engineering strategy to encode low-level file system events into discriminative behavioural representations.
- Enhances early notification of the initial 30 file-I/O files so that timely action may be taken before massive file encryption.
- Combines SHAP, LIME, and Integrated Gradients to offer the best transparency and trust in the ransomware detection systems because it gives a more detailed explanation of its behaviour around the globe and locally.

**Literature Review**

Recent studies in ransomware detection have started to move more towards behavior-sensitive, decentralized and explainable AI-driven systems to overcome the shortcomings of conventional signature-based approaches. Ekaterina Starchenko et al. [1] suggested the Decentralized Entropy-Driven Detection (DED) algorithm, which uses autonomous neural graph representations with entropy-based anomaly scoring on a decentralized architecture, a system with strong resilience, scalability and high detection accuracy by modeling complex system interactions, but the primary weakness is that the algorithm is quite computationally and implementation-heavy with high implementation complexity. Equally, Ignatius Rollere et al. [2] proposed Algorithmic Segmentation and Behavioral Profiling based on Temporal-Correlation Graphs (TCGs), a time-conscious graph of system activities, to identify polymorphic, never-seen ransomware with great accuracy, but that requires continuous and high-quality monitoring data, which adds significant overhead and decreases effectiveness in noisy settings. Elodie Mutombo Ngoie et al. [3], focusing on explainability, introduced an interpretable hybrid framework based on the use of BERT, RoBERTa, and DeBERTa with features-to-text conversion and feature-to-text and fine-tuning large models, which allow explainability (via features) as well as strong performance in detection; however, the complexity of feature-to-text conversion and the computational cost of fine-tuning large models restrict its application to real-time systems. On another note, Lafedi Svet et al. [4] introduced the Zero-Space Detection framework which exploits the principles of unsupervised clustering, deep learning and

ensemble to develop scalable low-latency detection in high-velocity conditions, yet the paper was withdrawn after a contested authorship, casting doubt on both the credibility and generalizability of the results. On the system level, Manabu Hirano and Ryotaro Kobayashi [5] proposed RanSMAP, an open data and hypervisor-based detector based on low-level storage and memory access patterns, which is more robust to evasion techniques and boosts the detection rate through memory features, but requires virtualization support and can add performance overhead. To resolve the problem of privacy and cross-organization cooperation, Daniel M. Jimenez-Gutierrez et al. [6] offered Federated Cyber Defense framework which is built on Federated Learning, which allows privacy-aware detection of ransomware across distributed systems, with system performance similar to centralized models, and the core weakness is higher communication and system complexity. All these works illustrate a big leap towards scalable, adaptive and behavioral ransomware detection and point to ongoing issues of high costs of computation, deployment complexity, data quality and system overhead.

**Challenges**

- Most of the models are based on post-encryption signature or the properties of the files, which are not easy to identify the existence of ransomware before they cause a lot of harm.
- The classical models tend to miss the relational interactions and a consecutive pattern of execution, resulting in the overlook of behavioral anomaly.
- Most of the current methods are not able to identify the newer variants of ransomware because they are not adaptable and use known signatures.
- Traditional models are prone to heterogeneous or noisy systems behavior, leading to high false-positive rates and lower reliability under real-world conditions.

**Methodology**

The primary objective of this study is to come up with a strong and explicable ransomware detection framework capable of detecting malicious file-encryption operations in their early stages of execution through analysis of kernel-level behaviour. To do that, the framework monitors low-level File I/O events by using Windows Event Tracing for Windows (ETW) and executes ransomware code and benign applications in a controlled sandbox. The resulting logs are purged, organized, coded alpha-numeric, and converted to discriminative behavioral features and are normalized and labeled to create an equally balanced training set. They are evaluated against a Hybrid Attention-Temporal Ensemble (HATE) model which combines Temporal Fusion Transformer, Graph Attention Network and Informer Transformer to learn to jointly capture temporal dependencies, relational event structure and long-sequence encryption burst. These deep learning models are then combined in a soft-voting ensemble process that yields a final ransomware probability score of the individual predictions made by the models. In order to achieve transparency and trust, Explainable AI methods such as SHAP, LIME, and Integrated Gradients are integrated so as to understand model choices´

surpasses a predetermined threshold, the system generates an alert, isolates the malicious process and generates a forensic-ready behavioral report with detailed explanations to allow timely and informed security measures.

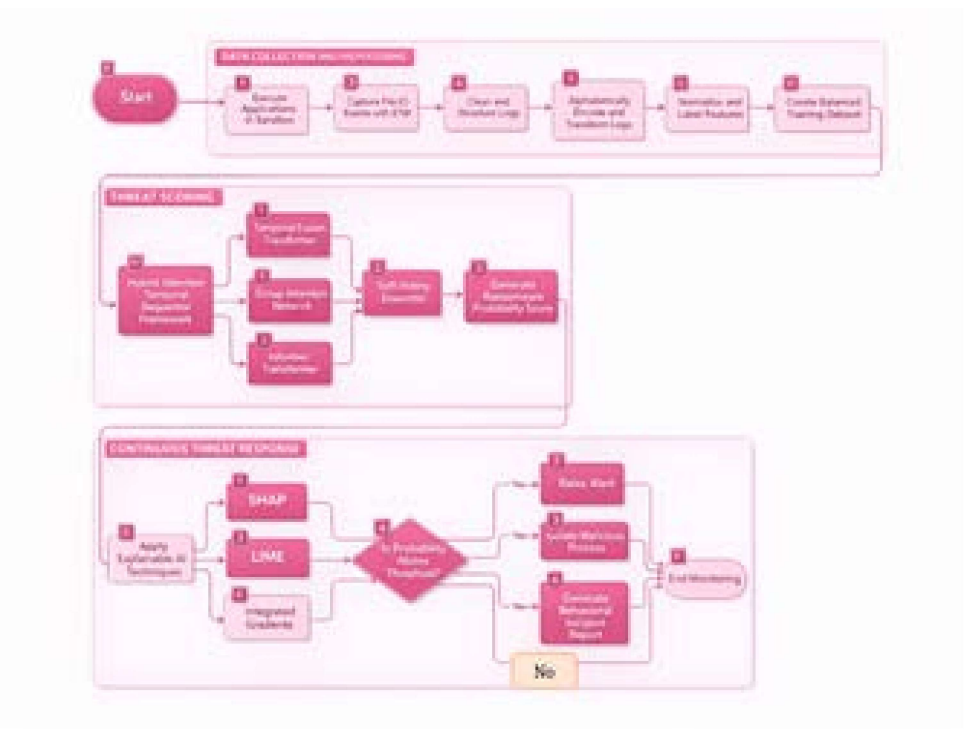


**Figure 1 Architecture for the Proposed Methodology**

**Input Collection – ETW Kernel Event Acquisition**

The kernel level file system behaviour under investigation in the proposed framework is monitored with the help of the integrated Windows Event Tracing to Windows (ETW) mechanism which is a low-overhead and fine-grained monitoring of system events. NT Kernel Logger is set to subscribe to the File I/O providers in particular such that all important file activities of all running processes are recorded in real time. Such actions are file Create, Read, Write, Rename, Delete, Cleanup and Close, which are the main signs of the ransomware encryption process. The acquisition of data is conducted continuously as the ransomware samples and legitimate applications are run within an isolated sandbox system and thus realistic mixed behavioral patterns are captured. The recorded kernel events are subsequently exported into formatted CSV log files which are then the ra±„

**Sandbox Environment Setup**

A sandbox environment is created as a virtualized environment where the ransomware samples can be safely executed in a controlled environment and the realistic system behaviour is collected without the risk of infecting the host. The testbed is composed of virtual machines executing Windows 10 and Windows Server 2019, which are a typical target platform of enterprises and desktops. In order to exclude any interference with the behavioral patterns under analysis, the built-in security systems like Windows Defender and antivirus services are turned off. The virtual machines are brought back to a clean snapshot after each execution cycle in order to provide consistency and remove the remnants of previous executions. In every experiment, an assortment of benign applications such as Office applications, web browsers, compression programs, and code editors run simultaneously with ransomware samples, thus creating a mix of actual activity profiles. Such a configuration allows one to generate a balanced dataset of both malicious and legitimate file su>:!

**Data Pre-Processing**

The raw ETW File-I/O logs that are obtained in the sandbox are full of noisy and redundant data that is not directly relevant to analysis of ransomware behavior. Thus, a pre-processing stage of data cleaning and organization is done in a systematic manner. Attributes not reflecting behavior like ProcessID, FileObject, FileKey and FileName is eliminated as it is runtime specific and does not offer meaningful information on patterns of malicious activity. It only retains behavior-oriented fields, i.e. EventName, FileType and Timestamp. Events are connected by the same FileKey and ProcessName to create ordered sequence of activities in order to rebuild the sequence of actions done on each file. Sequences with fewer than three events are dropped (as they cannot be decrypted by minimal activity) and sequences that lack enough operations related to encryption like Create, Read, Write, Delete or Rename are also deleted. The output of this process is a collection of clean and structured behavioral file-I/O sequences that can be used in feature extraction and deep learning analysis.

**Alphabetic Encoding of File I/O Events**

Once the files are pre-processed, alphabetic encoding of each file system operation is done to make it represented in a symbolic form that is easy to analyse in terms of behavioural patterns. In particular, the operations Create, Read, Write, Rename, Delete and Close are analyzed by characters C, R, W, N, D, and L respectively. As an example, a series of file operations like Create Rename Write Read Delete is encoded as CRWND. This encoding transforms detailed traces of events happening at the kernel into small streams of symbols which maintain the original file activity order of execution. Such symbolic streams of ransomware behavior are easier to identify repetitive malicious behavior, permit efficient feature extraction, and permit deep learning models to su

**Feature Engineering**

The behavioral sequences that are alphabetically encoded are converted into the quantitative form in the feature engineering stage, which reflects the unique properties of the activity of ransomware. The eid_count feature is the accumulation of the number of file operations made on a file, and it indicates the burst-like nature as is usually seen during the encryption process. The count of Read_Write frequency and the Rename_Read_Write pattern are read out to record the repetitive access and renaming patterns highly related to ransomware. Moreover, the Create, Delete, Cleanup and Close activity frequencies are calculated to indicate the abnormal file handling intensity. The entropy feature of file type indicates the diversity of types of files that have been affected, meaning that there have been large-scale encryption operations of heterogeneous data. Lastly, the sequence length represents the time richness of any interaction pattern of the files. These properties combined to create numerical vectors of behavioral features that are effective representations of ransomware encryption behzä learning models.

**Normalization & Dataset Formation**

The extracted behavioral features are normalized with Minmax scaling to make sure that every behavior attribute has an equal contribution to the training of the model since all the attribute values are brought to the fixed range 0-1. This avoids characteristics that have large number ranges to take over the learning process and enhances the convergence properties of deep learning models. Once it is normalized, the feature vectors are labeled with a class, with 0 (benign activity) and 1 (ransomware behavior) being the possible classes. The entire data is then split into two subsets with an 80:20 split with 80 percent of the data being used to train the models and the other 20 percent

of the data being used to test and evaluate the performance. This distance guarantees an impartial evaluation of the detection framework.

**Hybrid Attention–Temporal Ensemble (HATE) Framework**

The HATE model works based on the analysis of ransomware behaviour through three complementary viewpoints to provide early, accurate, and explainable detection. First, the Temporal Fusion Transformer learns the time-varying dynamics of file-I/O operations and predicts file access, the development of repetitive read-write-rename, and the increased weight of temporally significant events on ransomware operations. Simultaneously, the Graph Attention Network takes into consideration the structural dependencies between file manipulations by encoding event sequences into interaction graphs and demonstrating suspicious transition patterns that are characteristic of an encryption workflow. The Informer Transformer is an extension of the long behavioural sequence processing with probabilistic sparse attention that allows efficient detection of rapid, high-frequency bursts of encryption that take place during large-scale file compromise. Each of the models generates a ransomware probability score, which is fused by a soft-voting ensemble decision layer to generate a single and strong classification. The SHAP, LIME and Integrated Gradients are used in conjunction in order to be more transparent and to explain the global feature importance as well as the local and time-step level contribution. Once the last probability is greater than a specified threshold, the framework automatically sends an alarm, isolates the rogue process, and creates an easily readable behavioural report, which allows ransomware to be prevented in ti

The TFT is used to simulate the complicated time-dependent action of the ransomware encryption activities in file I/O sequences. In contrast to traditional sequential models, TFT can simultaneously represent the short-term and long-term temporal dependence, with the help of a combination of attention processes and gated residual networks. Within the framework proposed, TFT learns how ransomware gradually conducts repetitive read-write-rename operations during the course of time and dynamically allocates importance weights to each significant event, enabling the model to pay attention to the most significant steps in the encryption process. Such a capacity to emphasize operationally important temporal changes facilitates the framework to identify ransomware actions even in its initial implementation stages, enhancing the detection effectiveness and reaction speed.

The GAT is used to model the structural relationships between file system operations by modeling behavioral sequences as event-interaction graphs. In these graphs, the nodes are identifiable as certain file operation like: Create, Read, Write, Rename or Delete and the edges are the transitions between the two consecutive events within a sequence. GAT uses its attention mechanism to weight suspicious transitions with greater importance and which are often correlated with ransomware behavior like repetition read-write-rename patterns. This allows the model to concentrate on the relational patterns which have the greatest impact instead of considering all event interactions equally and thus increase the likelihood of detecting malicious activity which might not be evident b

The framework has the Informer Transformer to effectively analyze long file-operation sequences that are produced during ransomware execution. In contrast to regular transformers, Informer also uses a probabilistic sparse attention mechanism which greatly decreases computational complexity but retains the most informative temporal dependencies. This is especially applicable to detecting fast and bursting bursts of encryption traffic, which are indicative of a ransomware attack. The Informer model can be used to identify ransomware strains that act fast and are early and accurately detected by concentrating on the most relevant sections of a long time series of data where a large-scale file encryption can occur within a short time.

The Ensemble Decision Layer is the one that combines the separate results gained out of the Temporal Fusion Transformer, Graph Attention Network, and Informer Transformer to produce a single classification result. Every model provides a score of the ransomware likelihood in accordance with the representation learned throughout its study of the input conduct. A combination of these probability values is done through a soft-voting ensemble fusion strategy, whereby the scores are averaged to give one final ransomware probability score. The strategy exploits the synergy between temporal modeling, relational graph reasoning, and long-sequence attention analysis and, in this way, enhances the detection strength and minimizes the chances of false positives or misclassification.

**Explainable AI Integration – SHAP, LIME & Integrated Gradients**

Three Explainable AI (XAI) methods SHAP, LIME, and Integrated Gradients are introduced to address the issue of black-box nature of deep learning models and improve trust in the detection framework. SHAP is employed to offer both global and localized explanations quantifying the role of each feature to the model predictions, which allows attaining an overall idea of what behavioral patterns are most likely to be considered the most predictive of ransomware activity. LIME is a perturbation-based technique that tries to explain individual predictions by perturbing the input features about a particular example and trying to select the features that had the greatest impact on that particular classification. Integrated Gradients supplements these techniques by examining the internal gradients of deep learning models to point out the most important time-steps in the sequence of events that led to the decision. A combination of these XAI methods guarantees that each“”””\ZY

**Final Decision & Alert Generation**

The final step in the framework is that the probability score of the ransomware generated by the ensemble decision layer is checked against a predefined detection threshold. Once such probability surpasses the threshold, the system automatically identifies the activity as malicious and sends a security alert. The detected malicious process is then determined to stop the further encryption of files and their movement across. In addition to the alert, the detailed explanations produced by SHAP, LIME, and Integrated Gradients are provided, making it evident which features and time-steps have contributed to the decision to detect. Ultimately, an all-inclusive forensic-ready behavioral report is generated, which allows the security analysts to comprehend the pattern of the attack, confirm the alert, and d□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

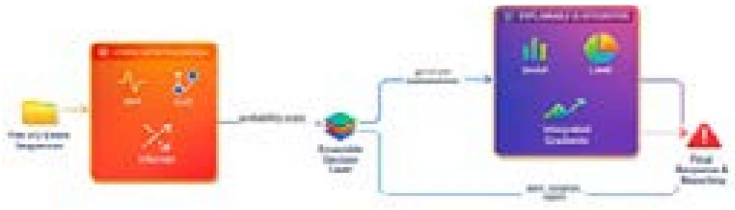

**Figure 2 Architecture for the Proposed Hybrid Attention–Temporal Ensemble Model**

**Results and Discussion**

In this part, the performance of the proposed HATE ransomware detection framework is examined based on the measures of the ability to detect files through I/O behavioural indicators at the kernel level, which are recorded with the help of ETW.

## Experimental Setup

**Table 1 Experimental Setup of the Proposed HATE Framework**

| Component | Description |
|---|---|
| Host System | Intel Core i7 CPU, 16 GB RAM, Windows 10 (64-bit) |
| Virtualization Platform | VMware Workstation |
| Guest Operating Systems | Windows 10 and Windows Server 2019 |
| Sandbox Environment | Isolated virtual machines with snapshot restoration after each execution |
| Security Controls | Windows Defender and antivirus services disabled to avoid interference |
| Malware Dataset | Real-world ransomware samples executed inside sandbox |
| Benign Applications | MS Office tools, web browsers, compression utilities, code editors |
| Data Acquisition Tool | Windows Event Tracing for Windows (ETW) with NT Kernel Logger |
| Captured Events | File Create, Read, Write, Rename, Delete, Cleanup, Close |
| Log Format | CSV structured event logs |
| Preprocessing | Noise removal, sequence reconstruction, alphabetic encoding |
| Feature Engineering | Read–Write frequency, Rename–Read–Write patterns, FileType entropy, eid_count, sequence length |
| Feature Normalization | Min–Max scaling in range [0,1] |
| Dataset Split | 80% Training, 20% Testing |
| Deep Learning Models | Temporal Fusion Transformer, Graph Attention Network, Informer Transformer |
| Ensemble Strategy | Soft-voting based probability fusion |
| Explainability Methods | SHAP, LIME, Integrated Gradients |
| Detection Threshold | 0.5 |
| Performance Metrics | Accuracy, Precision, Recall, F1-score, FPR, AUC |
| Implementation Tools | Python, PyTorch, Scikit-learn |
| Evaluation Mode | Early-stage detection and full sequence analysis |

## Detection Performance Comparison

Table 2 indicates clearly that the proposed HATE framework outperforms the conventional machine learning as well as the single deep learning systems. Traditional classifiers, including the Random Forest and SVM record the lower accuracy of 91.6 and 89.8, respectively, showing their weak proficiency in identifying the compound temporal and behavioural dynamics of ransomware work. CNN-LSTM model is more effective with 94.1 percent; however, it does not have the efficiency of accessing the entire temporal context and relationship event structures that are long. More sophisticated attention-based models such as TFT, GAT and Informer also increase the detection accuracy to 95.1-96.0, indicating their ability to capture time dependencies, interaction relationships, and long encryption bursts. The proposed HATE model has the best overall accuracy (97.8%), the best precision (97.6%), recall (98.1%), and F1-score (97.8%), as well as the lowest false positive (2.1%), and the highest AUC (0.989). These findings support the hypothesis that a combination of time, structure and long-sequence attention processing via ensemble fusion offers a more discrimkqlrmsnsntnv

**Table 2 Detection Performance Comparison**

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | FPR (%) | AUC |
|---|---|---|---|---|---|---|
| Random Forest | 91.6 | 90.4 | 92.1 | 91.2 | 6.8 | 0.931 |
| SVM | 89.8 | 88.6 | 90.2 | 89.4 | 8.3 | 0.912 |
| CNN-LSTM | 94.1 | 93.9 | 94.4 | 94.1 | 4.8 | 0.956 |
| TFT | 95.6 | 95.1 | 96.0 | 95.5 | 3.7 | 0.971 |
| GAT | 95.1 | 94.6 | 95.5 | 95.0 | 3.9 | 0.968 |
| Informer | 96.0 | 95.7 | 96.3 | 96.0 | 3.3 | 0.975 |
| HATE (Proposed) | 97.8 | 97.6 | 98.1 | 97.8 | 2.1 | 0.989 |

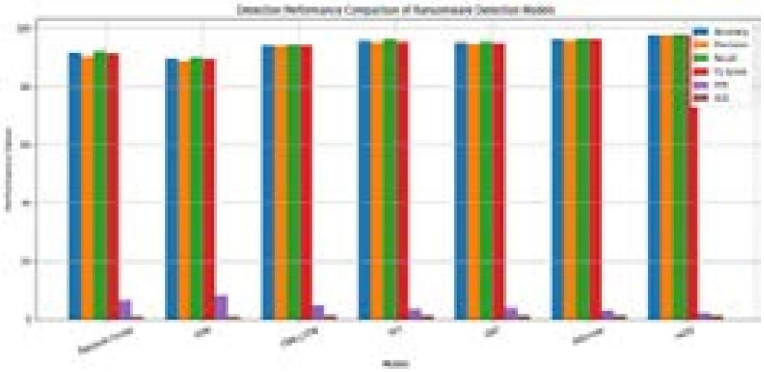

**Figure 3 Detection Performance Comparison of Ransomware Detection Model**

## Confusion Matrix for HATE

The confusion matrix of the suggested HATE framework on the test sample of 2,496 samples was shown in Figure 4. Out of 1248 benign cases, 1221 are correctly identified and 27 are wrongly labeled as ransomware which means the false positive rate is very low. Likewise, in 1,248 ransomware samples, 1,224 are accurately identified as such, and 24 are falsely detected as a benign operation, which shows that the framework has a high potential to detect malicious encryption behaviour. Having 51 total misclassifications, the findings will affirm the HATE model as having high reliability and accurate discrimination of benign and ransomware activities.

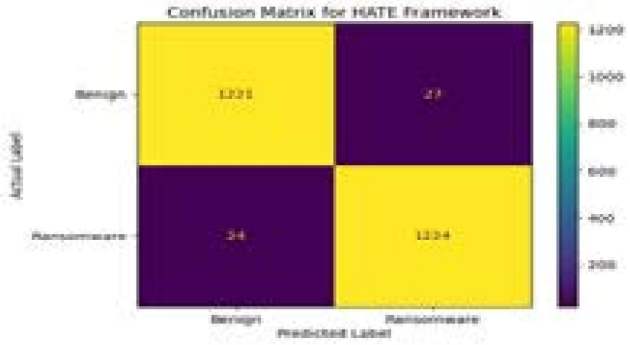


**Figure 4 Confusion Matrix**

## Early-Stage Detection Analysis

The analysis of early-stage detection that is presented in table 3 shows that the proposed framework is capable of detecting ransomware behaviour at its early stages of execution. The

model already reaches a recall of 91.0 and an accuracy of 90.2 when it is only fed with the first 10 file events and can detect attacks in an average of 0.82 seconds. The detection performance is better with increasing observation window of 20 and 30 events, as 94.5 and 96.9 are achieved in accuracy, respectively, with high recall of 95 and above. The highest accuracy of the framework with a recall of 98.1 was achieved using the complete file-I/O sequence, which is 97.8%. These findings affirm that the system is capable of reliably identifying ransomware in just a few seconds of its operation, allowin□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

**Table 3 Early-stage Detection**

| Observed Events | Accuracy (%) | Recall (%) | Avg Detection Time (s) |
|---|---|---|---|
| First 10 events | 90.2 | 91.0 | 0.82 |
| First 20 events | 94.5 | 95.2 | 1.36 |
| First 30 events | 96.9 | 97.3 | 1.92 |
| Full sequence | 97.8 | 98.1 | 2.67 |

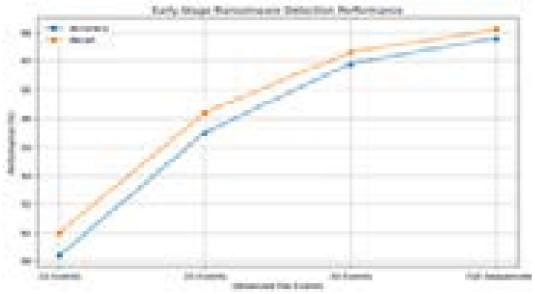

**Figure 5 Early-Stage Ransomware Detection Performance**

## Ablation Study

Figure 6 shows the ablation experiment, which was performed to estimate the individual and combined performance of the three deep learning models adopted in the proposed HATE framework. In their independent use, the accuracy of TFT, GAT, and Informer models is 95.6, 95.1, and 96.0, respectively, which implies that each architecture can learn meaningful ransomware behaviour patterns. When two models are combined, the performance is further enhanced and the highest accuracy of 97.1 was obtained with the TFT + Informer configuration of the two-model combinations. The entire ensemble incorporating TFT, GAT and Informer provide the most favourable outcome with an accuracy of 97.8, showing that the capabilities of each component are complementary to one another in terms of learning. This proves that the hybrid ensemble design is far much better in detection accuracy than any single or partial combination of models.

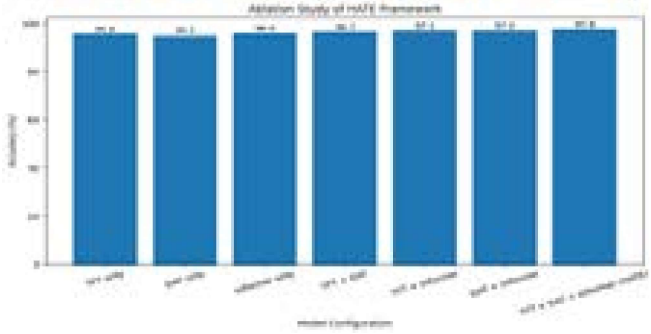

**Figure 6 Ablation Study**

**Explainability Results**

Figure 7 provides the global feature importance, as per SHAP analysis, which identifies the behavioural attributes that have the strongest impact on the ransomware detection decisions of the proposed framework. The maximum SHAP values are 0.284 and 0.271 with Read-Write Frequency and Rename-Read-Write Pattern Count respectively, which means that repetitive read-writes and renaming loops are the most significant predictors of ransomware encryption behavior. FileType Entropy also adds to the value significantly with a value of 0.198, indicating that during mass encryption, ransomware mostly targets the various file formats. Sequence Length and Delete Frequency have a moderate value, indicating that long activity bursts and atypical file deletion patterns are additional factors that evidence ransomware. These findings affirm that the framework ca'–¨

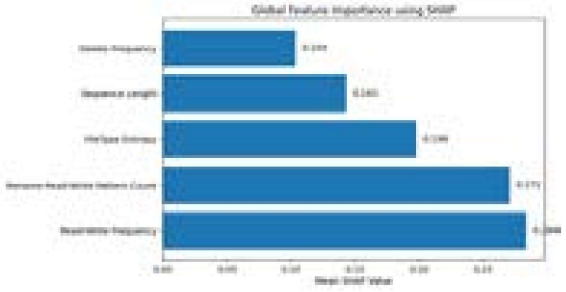

**Figure 7 Global Feature Importance using SHAP**

**Local Instance Explanation**

Figure 8 shows the local explanation of a representative ransomware prediction provided by the proposed HATE framework using the LIME. The findings indicate that Rename-Write Cycle has the best contribution score of + 0.31, which means that the repetitious renaming and read-write cycles are the most, significant, factor that motivates the classification to become ransomware. It is then succeeded by FileType Entropy with a contribution of +0.22, which indicates that the encryption of various heterogeneous file types is a great indicator of malicious behavior. High eid_count is a feature with a value of +0.19, which shows the burst-like nature of file operations that are usually witnessed during the execution of ransomware. Also, Frequent Delete Operations (+0.14) help to support the decision to detect even further because ransomware usually removes original files after encryption. Lastly, Short Idle Gaps of +0.11 represent the low number of breaks between operations, which point to the automated and constant character of ransomware operation. These findings in combination with figure 8 indicate that the model is right in its focus on realistic beha\KmHDB>=”=“=‘>

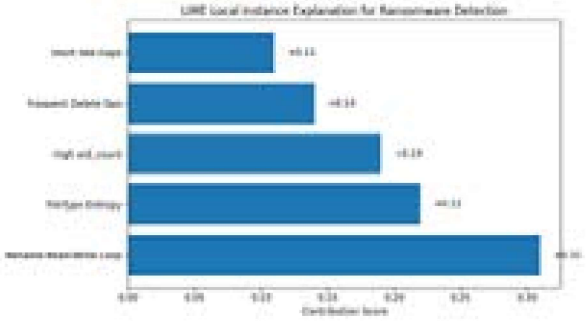

**Figure 8 LIME Local Instance Explanation for Ransomware Detection**

The time-step attribution results generated by the use of the Integrated Gradients are shown in figure 9, which determines the most significant parts of file-I/O sequences that aid in the detection of ransomware. The change in attribution scores is progressive, with the value of 0.18 at the first 10 events, 0.29 in the 11 to 20 events, and a maximum of 0.41 in the 21 to 30 events. This means that the unique behaviour of ransomware is usually manifested at a later stage, where active file encrypting activities commence following the initial setup phase. After 30 events, the contribution decreases to 0.12, indicating the existence of the critical behavioural patterns which were already taken earlier in the execution process. The results obtained can be applied in supporting the need to affirm that the framework proposed can identify ransomware at the exact point of detection when encryption is taking place to a high level to allow intervention at the right time before most files are destroyed.

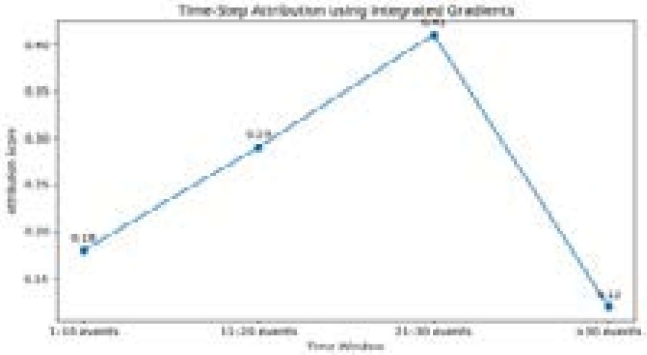

**Figure 9 Time-step Attribution using Integrated Gradients**

The effect of decision thresholds on the detection performance of the proposed HATE framework is depicted in figure 10. The system attains very high recall of 99.2 at a lower threshold of 0.4, which means that nearly all the ransomware cases are detected, but this is at the expense of increased false positive rate of 4.8. A threshold of 0.5 gives the best balance with a precision of 97.6 and a recall of 98.1 with a false positive rate of 2.1 which is significantly low. Precision is further enhanced in higher threshold of 0.6 to 98.9, but recall decreases to 95.4 which means that more instances of ransomware can be missed. These findings support the use of 0.5 as the operational threshold b

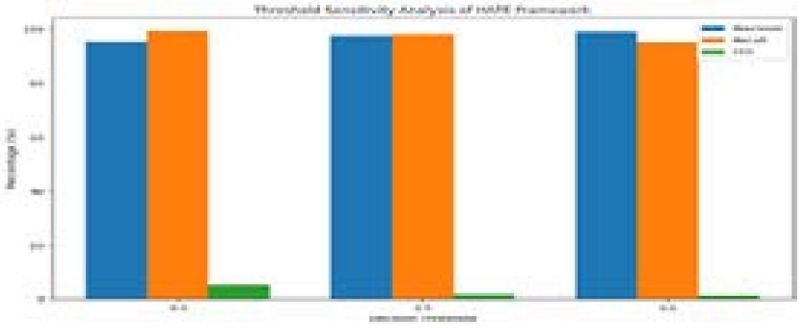

**Figure 10 Threshold Sensitivity Analysis of HATE Framework**

## Conclusion and Future Scope

As the ransomware attacks are rapidly developing and can encrypt massive amounts of data in seconds, the old signature-based and traditional behaviour-based detection methods are becoming more and more ineffective. This work introduced the HATE framework of early and understandable

ransomware detection based on file I/O behavioural analysis of the kernel level. The framework performs well in capturing the temporal evolution, relational transitions and long-sequence encryption bursts, by using Temporal Fusion Transformer, Graph Attention Network, and Informer Transformer in combination with a soft-voting ensemble strategy. The suggested model has high detection rates 97.8 and low false positive rates 2.1 and is able to identify a ransomware in the initial 30 file events in order to take preventive action. The combination of SHAP, LIME and Integrated Gradients also guarantee the transparency and confidence in the detection process. The framework may be expanded as future work to real-time deployment in operational enterprise settings, tested on larger cross-platform and multicentre datasets, and improved with adaptive thresholding and onpq~koqr

**References**

1. Starchenko, E., Bellinghamshire, H., Pickering, D., Weatherspoon, T., Berkhamstead, N., Green, E., & Rothschild, M. (2025). Decentralized Entropy-Driven Ransomware Detection Using Autonomous Neural Graph Embeddings. arXiv preprint arXiv:2502.07498.
2. Rollere, I., Hartsfield, C., Courtenay, S., Fenwick, L., & Grunwald, A. (2025). Algorithmic Segmentation and Behavioral Profiling for Ransomware Detection Using Temporal-Correlation Graphs. arXiv preprint arXiv:2501.17429.
3. Ngoie, E. M., Nkongolo, M. N. W., Azugo, P., & Tokmak, M. (2025). Interpretable Ransomware Detection Using Hybrid Large Language Models: A Comparative Analysis of BERT, RoBERTa, and DeBERTa Through LIME and SHAP. arXiv preprint arXiv:2511.13517.
4. Svet, L., Brightwell, A., Wildflower, A., & Marshwood, C. (2025). Unveiling zero-space detection: A novel framework for autonomous ransomware identification in high-velocity environments. arXiv preprint arXiv:2501.12811.
5. Hirano, M., & Kobayashi, R. (2025). RanSMAP: Open dataset of Ransomware Storage and Memory Access Patterns for creating deep learning based ransomware detectors. Computers & Security, 150, 104202.
6. Jimenez-Gutierrez, D. M., Zuazua, E., Del Rio, J., Sliusarenko, O., & Uribe-Etxebarria, X. (2025). Federated Cyber Defense: Privacy-Preserving Ransomware Detection Across Distributed Systems. arXiv preprint arXiv:2511.01583.
7. Routray, S., Prusti, D., & Rath, S. K. (2023, May). Ransomware attack detection by applying machine learning techniques. In Machine Intelligence Techniques for Data Analysis and Signal Processing: Proceedings of the 4th International Conference MISP 2022, Volume 1 (pp. 765-776). Singapore: Springer Nature Singapore.
8. Ahmad, S., Zulkifli, Z., Nasarudin, N. H., Imran, M., & Ariff, M. (2023, September). A Recent Systematic Review of Ransomware Attack detection in machine learning techniques. In 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS) (pp. 349-354). IEEE.
9. Singh, A., Mushtaq, Z., Abosaq, H. A., Mursal, S. N. F., Irfan, M., & Nowakowski, G. (2023). Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data. Electronics, 12(18), 3899.
10. Gazzan, M., & Sheldon, F. T. (2023). Opportunities for early detection and prediction of ransomware attacks against industrial control systems. Future Internet, 15(4), 144.
11. Marcinkowski, B., Goschorska, M., Wileńska, N., Siuta, J., & Kajdanowicz, T. (2024). Mirad: A method for interpretable ransomware attack detection. IEEE Access.
12. Thummapudi, K., Lama, P., & Boppana, R. V. (2023). Detection of ransomware attacks using processor and disk usage data. IEEE Access, 11, 51395-51407.

13. Zhang, X., Wang, C., Liu, R., & Yang, S. (2024). Federated rnn-based detection of ransomware attacks: A privacy-preserving approach.
14. Begovic, K., Al-Ali, A., & Malluhi, Q. (2023). Cryptographic ransomware encryption detection: Survey. Computers & Security, 132, 103349.