Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TOWARD TRUSTWORTHY RANSOMWARE DETECTION: AN EXPLAINABLE AND IMBALANCE-AWARE TRANSFORMER-BASED FRAMEWORK

ARUNA RAO S L [1] , DR. NAGARATNA P HEGDE[2], DEVI[3], DR V RAMALATHA4, JAYAVELU S[5], PUPPALA RAMYA[6], ELANGOVAN MUNIYANDY[7] DR. SURESH KUMAR PITTALA[8]

[1] Department of Computer Science & Engineering
BVRIT HYDERABAD College of Engineering for Women, Telangana, India.
[2] Professor, Department of CSE, Vasavi College of Engineering, Hyderabad, Telangana, India.
[3] Associate Professor, Department of CSE, CMR TECHNICAL CAMPUS
Medchal Road, Kandlakoya, Telangana, India - 501401
[4] Associate Professor-Selection Grade, Department of Mathematics, Presidency University, Bangalore
[5] Department of Mechanical Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Avadi 600062, India.
[6] Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur Dist., A.P., India.
[7] Department of Biosciences, Saveetha School of Engineering. Saveetha Institute of Medical and Technical Sciences, Chennai - 602 105.
[8] Professor, Dept of ECE, R.V.R & J.C College of Engineering, Guntur – 522019

E-mail: arunaraosl@gmail.com nagaratnaph@staff.vce.ac.in, blaxmanphd@gmail.com, v.ramalatha@gmail.com, sjayavelu@veltech.edu.in, mothy274@kluniversity.in, muniyandy.e@gmail.com, dr.sureshkumarpittala@gmail.com

ABSTRACT

The use of encryption, data theft, and disruptive destruction has made ransomware a major cybersecurity threat, and traditional signature-based defences are often ineffective against novel and rebranded ransomware. The purpose of this study was to construct an explicable, imbalance-conscious deep learning model of detecting ransomware based on handcrafted behavioural features and transformer-based representation learning. The experimental machine-learning design was implemented on a new public ransomware corpus of 21,752 samples (26 malware families). The training was skewed intentionally to represent the class imbalance that was present. The proposed pipeline extracted file-system, process, registry, API, and network indicators and then learned higher-order dependencies using a transformer encoder that is optimised with focal loss, class weighting, and SMOTE. The proposed model had a better accuracy of 97.18% with a 0.962 precision, 0.951 recall, 0.956 F1-score, and 0.986 ROC-AUC, compared to the transformer baseline and other classical and deep learning models. Analysis of SHAP and attention further demonstrated that the most noteworthy indicators were file rename rate, write bursts that appear encryption-like, frequency of registry modification, and suspicious PowerShell activity. In general, the framework provides precise, low-false-positive, and interpretable ransomware detection that would be used in security operations.

Keywords: *Ransomware Detection, Explainable AI, Transformer Networks, Behavioural Analysis, Class Imbalance, Deep Learning, Malware Classification, Cybersecurity*

1. INTRODUCTION

Ransomware is a type of threat that has now evolved into an extremely coordinated model of extortion combining encryption, data theft, and operational disruption. In India, this risk is particularly topical since, according to the 2024 ransomware report by CERT-In, LockBit, Ransomhub, and KillSec are the most relevant groups that impact Indian cyberspace. [1] [2]. The same report reveals that manufacturing, finance, and IT/ITeS were some of the most targeted sectors, indicating that ransomware is no longer isolated to isolated cases but is an operational threat that has become a sustained and continuous threat to critical and commercial environments. [3] [4].

Although there has been advancement in malware analytics, most ransomware detection systems are limited by shallow feature learning, weak minority-class sensitivity, or black box decision-making. [5] [6] [7]. Signature-based approaches have issues with the changing family and rebranded versions, whereas end-to-end deep models lack interpretability, which is needed in security operations. The research question is thus: Can a hybrid framework, comprising handcrafted behavioural features, transformer-based learning, class-imbalance handling, and Explainable AI, enhance the ransomware detection and, at the same time, generate justifications that are easy to understand by an analyst? A recently published public 2024 ransomware dataset offers an appropriate experimental foundation to this study, consisting of 21,752 samples of 26 malware families, including 11 major ransomware families such as LockBit, Maze, REvil and WannaCry. [8] [9].

2. The relevance of this research is that it is applicable in practical terms to the present-day ransomware behaviour and operational defence requirements. According to the report by CERT-In, it becomes increasingly important to detect and explain ransomware-related activities as soon as possible because of the increasing use of stealthy execution chains, legitimate system tools, and multi-stage extortion tactics by ransomware actors [10] [11] [12]. With this context, a model with the ability to not only detect malicious behaviour, but also explain why a sample was flagged, is better suited to be deployed in the real world, particularly in security teams that must rapidly and accurately triage alerts [13] [14].

This research aimed to create an elucidate, imbalance-capable deep learning model to detect ransomware attacks using proposed handcrafted behavioural features and transformer-based representation learning.

3. This research has three key contributions. First, it presents a handcrafted set of behavioural features that focus on capturing ransomware-relevant indicators like bursts in file-renaming, registry modification, API frequency of use, and suspicious process activity. Second, it suggests a transformer-based architecture with optimisation that is imbalance-aware to enhance minority-class ransomware recognition. Third, it supports Explainable AI with SHAP-based and attention-based interpretations in a manner that the decisions made by the model are clearly comprehensible and are operationally meaningful. Collectively, these contributions create an accurate detection framework that is practical and can be used in cybersecurity.

2. LITERATURE REVIEW

More recent studies of ransomware have been moving away from simple signature matching to behaviour-dependent deep learning. Gulmez et al. introduced XRan, an explainable deep-learning ransomware detector that uses dynamic sequences of analysis and model-agnostic explanation to enhance both detection and transparency, demonstrating that XAI is becoming an essential requirement in malware defence as opposed to an optional add-on. [15] [16]. More recently, transformer-based methods have been introduced as a plausible direction; for instance, Pulse uses Transformer models to model the assembly sequences of ransomware, and reports improved zero-day detection behaviour compared to tabular baselines, suggesting that contextual sequence learning is well adapted to the task of modelling ransomware activity. Simultaneously, in larger malware XAI surveys, explainability algorithms like SHAP, LIME, and attention analysis are becoming more popular to make security models more accessible to analysts and operators. [17] [18] [19].

The current research is based on three fundamental concepts. To begin with, ransomware can best be described as a process of behavioural attack, whereby bursts of file modification, registry manipulation, API repetition, privilege-related execution, and suspicious network activity collectively point to a malicious intent [20]. Second, the transformer self-attention is particularly adaptable to the given issue as it may be trained to learn the dependencies across heterogeneous behavioural tokens and long execution windows, which is increasingly being reflected in recent ransomware modelling work. Third, explainable AI is a key element in cybersecurity since the detections should be justifiable, not just accurate; recent studies in explainable AI on ransomware and malware analysis have all indicated that such detection needs to be justifiable as opposed to being simply accurate.

Although these progresses have been made, there are several gaps. Most ransomware detectors have not been combined in a single pipeline but instead continue to rely on either handcrafted features alone or black-box deep representations alone. Studies of explainable ransomware already in the literature

tend to demonstrate interpretability, yet might not explicitly address the issue of class imbalance, which is a realistic phenomenon in live security settings when ransomware is a rare but high-impact condition. [21] [22]. Furthermore, recent Indian threat intelligence indicates that ransomware activity continues to have an operational significance, with CERT-In citing LockBit, Ransomhub, and KillSec as the predominant groups that impact the Indian cyberspace in 2024, as well as emphasising the use of stealthy living-off-the-land techniques. This reinforces the necessity of an accurate, yet imbalance-aware and explainable model. Thus, the question of this paper is whether a hybrid framework consisting of handcrafted behavioural features, transformer-based learning, imbalance-aware optimisation, and XAI could be used to provide better and more interpretable ransomware detection than the current methods.

3. METHODOLOGY

3.1. Research Design and Problem Formulation

This work will take an experimental machine-learning format on binary ransomware detection, where the objective will be to classify each observed execution trace or behavioural window as being of a benign or a ransomware nature. The proposed framework is designed to support realistic endpoint and sandbox telemetry, with the architecture design objective to integrate handcrafted behavioural indicators, transformer-based representation learning, imbalance-conscious training, and post-hoc explainability into a single deployable pipeline. The rationale behind this design is based on the current Indian threat environment: CERT-In 2024 ransomware report states that LockBit, RansomHub, and KillSec were among the predominant groups having a presence in Indian cyberspace, with manufacturing, finance, and IT/ITeS being highly targeted; abuse of living-off-the-land binaries and scripts, such as PowerShell and Command Prompt, was also aggressively pursued.

This research uses a recent open ransomware dataset and frames the analysis in terms of the Indian operational context. The public corpus selected is the 2024 Zenodo ransomware dataset, which contains 21,752 samples balanced between malicious and benign files and spans 26 families of malware, including 11 ransomware families such as Cerber, Darkside, Dharma, GandCrab, LockBit, Maze, Phobos, REvil, Ragnar Locker, Ryuk, Shade, and WannaCry. Practically, the model is trained on behavioural samples based on this corpus and is tested under a deliberately skewed training regime to simulate real-world conditions, with ransomware events being far less common than benign activity

3.2. System Architecture

The proposed model is based on five consecutive steps, as shown in Figure 1. The first stage involves the collection of raw behavioural traces based on sandbox execution logs, API sequences, file-system events, registry operations, process creation traces, and network metadata. Second, a handcrafted feature engineering component transforms the raw behaviour into a small and semantically important vector. Third, a deep transformer encoder memorises higher-order interactions among the feature tokens and time windows. Fourth, the ransomware probability is provided by an imbalance-conscious classification head. Lastly, an explainability layer generates local and global rationales based on SHAP, attention attribution and optional LIME overlays.

This architecture (Figure 1) is selected, since recent ransomware studies reveal that transformer-style models are becoming more effective in modelling long-range dependencies and cross-modal behavioural cues. Recent ransomware research (including Ransom Former) has shown that transformer-based architectures can effectively combine and process both byte-level and API-related indicators, and recent surveys on explainable malware analysis have highlighted the fact that high-performing black-box detectors are not to be trusted in security-critical settings unless explanation mechanisms are incorporated into the workflow.

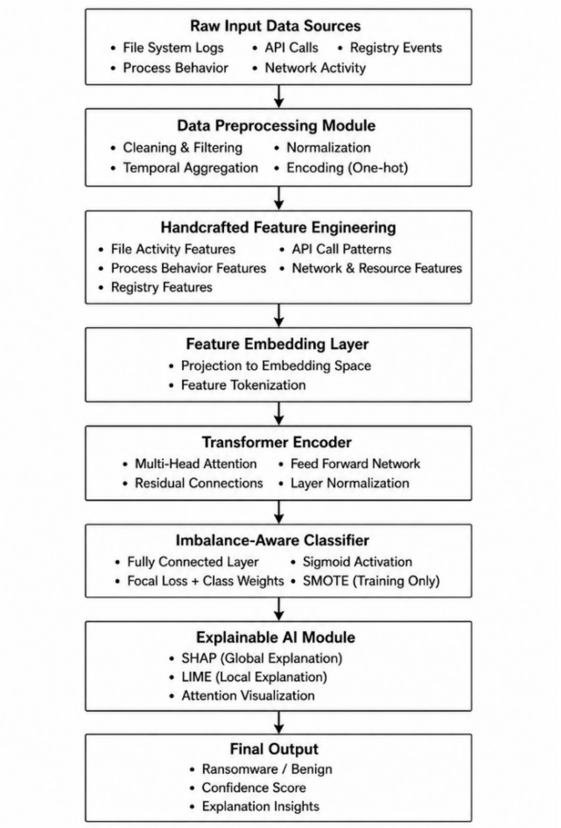

Figure 1 System Architecture

3.3. Data Acquisition and Preprocessing.

The experimental pipeline starts with the extraction of runtime traces of each sample of the chosen data set. The system logs file -system activity, API calls, registry activity, process spawning calls, and network behaviour, to every executable or observed execution instance. The raw logs are normalised into a regular event schema that should include the event type, event timestamp, process that generated the event, target object and the magnitude of the event. Duplicate traces and corrupted records are deleted, and the rest of the samples are aligned in fixed-length behavioural windows in such a way that each instance describes a similar observation interval. Let the raw behavioural record for the i-The sample is represented as

$$\mathbf{R}_i = \{e_{i1}, e_{i2}, \dots, e_{iT}\},$$

where e_{it}denotes the event observed at time step t, and $T$is the window length. After preprocessing, each sample is converted into a structured feature vector.

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}] \in \mathbb{R}^d,$$

where $d$is the number of handcrafted and learned features. Missing values are imputed using robust median-based substitution for numeric fields and mode-based substitution for categorical fields. Continuous features are scaled using min-max normalisation:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min} + \epsilon},$$

where ϵ is a small constant to prevent division by zero. Categorical attributes such as event type or process category are one-hot encoded or embedded into low-dimensional vectors before fusion. To reduce noise and improve reproducibility, the preprocessing stage also performs temporal aggregation. Repeated events within a short interval are merged, and burst statistics are computed to capture short-lived ransomware behaviour such as rapid file rewrites, mass renaming, and encryption-like I/O spikes. This stage is represented in Fig. 2 as the data-cleaning and feature-construction pipeline.

3.4. Handcrafted Behavioural Feature Engineering

The handcrafted feature layer is devised to capture the operational signatures of ransomware that can be observed before or during encryption. In contrast to purely end-to-end deep systems, these properties are explicitly interpretable and assist the model in identifying malicious intent based on sparse or noisy telemetry. The proposed feature set contains the categories below.

In the first step, the features of file-system intensity are obtained. They are file-creation rate, file-modification rate, file-rename rate, delete-to-create ratio, and extension-mutation rate. The ransomware generally causes anomalous write-heavy and repeated extension changes, particularly during mass encryption.

Second, process-behaviour features are computed. These are process-spawn depth, the branching factor of child-process, process lifetime entropy, the score of command-line anomaly, and the indicators of privilege-escalation. The ransomware operated by humans can often be disguised as harmless-looking administrative utilities and scripting shells; these features can assist in exposing suspicious chaining behaviour.

Third, the registry and persistence features are quantified. These are the frequency of registry key modifications, access frequency of autorun-path, the event of creation of services and the frequency

of write to scheduled tasks. These indicators are significant since ransomware frequently tries to persist after a reboot or disabling the protection features.

Fourth, API and system-call features are based on distributions of call frequencies, counts of rare calls, counts of repeated-call n-grams, and burst periodicity. These capture coordination patterns in encryption loops, deletion loops and anti-recovery routines.

Fifth, the calculation of such features as network and host-resource features is carried out. These are the outbound connection counts, failed-connection rate, domain entropy, bytes sent per second, CPU spike ratio, and memory-resident anomaly score. The features have been specifically useful in cases where ransomware involves exfiltration, beaconing, or massive file operations. For the j- The engineered feature, we define

$$f_j = \phi_j(\mathbf{R}_i),$$

where $\phi_j(\cdot)$is the extraction operator for the j- The behavioural descriptor. The final handcrafted vector is

$$\mathbf{f}_i = [f_1, f_2, \dots, f_m] \in \mathbb{R}^m.$$

To preserve interpretability, each handcrafted feature is logged with a semantic label so that later XAI outputs can be mapped back to meaningful ransomware behaviours rather than abstract latent variables.

3.5. Imbalance-Aware Deep Learning Architecture

Even though the chosen open dataset is at the corpus level, ransomware detection in practice is inherently skewed towards malicious events as opposed to benign traffic. To simulate this more realistic operating condition, the training split is re-sampled deliberately to create a minority ransomware class, and maintain stratification in validation and test sets. The design allows the model to train with the identical skew that security teams experience in practice. The deep architecture is a hybrid transformer encoder which runs on the handcrafted feature sequence. Each group of features is then projected into a shared embedding space before being sent into the transformer:

$$\mathbf{z}_i^{(0)} = \mathbf{W}_e\mathbf{f}_i + \mathbf{b}_e,$$

where $\mathbf{W}_e$and $\mathbf{b}_e$*These equation areThese* Equations are learnable parameters. The projected vector is reshaped into token-like segments so that the model can learn inter-feature dependencies, cross-feature interactions, and temporal context. For the l- The transformer layer, the self-attention operation is defined as

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V},$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are query, key, and value matrices, respectively. Multi-head attention is applied to learn complementary behavioural relations across file, process, registry, and network domains.

Each transformer block contains residual connections, layer normalisation, and dropout regularisation to improve stability and prevent overfitting:

$$\mathbf{h}^{(l+1)} = \text{LayerNorm}\big(\mathbf{h}^{(l)} + \text{Dropout}(\text{MHA}(\mathbf{h}^{(l)}))\big),$$

followed by a feed-forward module

$$\mathbf{u}^{(l+1)} = \text{LayerNorm}\big(\mathbf{h}^{(l+1)} + \text{Dropout}(\text{FFN}(\mathbf{h}^{(l+1)}))\big).$$

A final fully connected classifier produces the ransomware probability:

$$\hat{y}_i = \sigma(\mathbf{w}^T\mathbf{u}^{(L)} + b),$$

where $\sigma(\cdot)$is the sigmoid function and $L$is the number of transformer layers. The architecture is intentionally lightweight enough for endpoint deployment while still expressive enough to capture nonlinear ransomware behaviour.

3.6. Class-Imbalance Handling and Optimisation

To address class skew, the proposed method combines focal loss, class weighting, and controlled oversampling. The training objective is based on focal loss:

$$\mathcal{L}_{\text{focal}} = -\alpha_t(1-p_t)^\gamma \log(p_t),$$

where $p_t$is the predicted probability of the true class, $\alpha_t$is a class-balancing coefficient, and $\gamma$is the focusing parameter. This loss reduces the influence of easy majority-class examples and places more emphasis on difficult ransomware cases.

The total objective further includes L_2regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{focal}} + \lambda \parallel \theta \parallel_2^2$$

where θdenotes the learnable parameters and λcontrols the penalty strength. During training, minority-class augmentation is applied only to the training set. Validation and test sets remain untouched to preserve unbiased evaluation.

Optimisation is performed using AdamW with cosine learning-rate decay. Early stopping is triggered when the validation F1-score does not improve for a fixed patience interval. Threshold tuning is performed after training so that the

operating point can be optimised for either high recall, low false positives, or a balanced trade-off depending on deployment requirements.

3.7. Explainable AI Layer

Explainability is a key component of the proposed system since ransomware defence teams should be made aware of the reason why an alert was triggered. This is particularly relevant in security operations centres, wherein analysts need to differentiate between actual ransomware activity and justifiable administrative automation. According to recent XAI surveys in malware analysis, opaque detectors are hard to operationalise unless they are justified in a human-readable form. The explanation module is thus a combination of three complementary approaches. First, SHAP is employed to determine the contribution that each handcrafted feature makes to the overall ransomware score. Given a feature vector f, the SHAP value of f 1 is the contribution of feature 1 to the prediction, relative to the background distribution. Second, the weights of attention of the transformer are plotted to indicate which feature interactions contributed most to the prediction. Third, LIME is scalable to single samples to enable a local explanation to analysts. The local explanation for a sample $i$is summarized as

$$\hat{y}_i = g(\mathbf{f}_i) + \epsilon,$$

where $g$is the interpretable surrogate model and $\epsilon$is the approximation error. The combined interpretation layer allows the system to report, for example, that a ransomware decision was driven by rapid file renaming, high registry modification rate, suspicious PowerShell usage, and anomalous outbound activity. Such evidence is more actionable than a raw binary label.

3.8. Evaluation Protocol

The model is evaluated on a stratified train-validation-test split. In addition, 5-fold cross-validation is used to verify robustness. Performance is reported using accuracy, precision, recall, F1-score, specificity, ROC-AUC, false positive rate, inference latency, and memory footprint.
The core metrics are computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}, \text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \text{Specificity} \frac{TN}{TN + FP}.$$

ROC-AUC is computed from the receiver operating characteristic curve generated by varying the decision threshold. The false positive rate is reported as

$$\text{FPR} = \frac{FP}{FP + TN}.$$

To show the benefit of the proposed design, the model is compared against Random Forest, XGBoost, SVM, MLP, 1D-CNN, LSTM, and a transformer baseline without handcrafted features or XAI. An ablation study is also conducted on handcrafted features only, transformer only, imbalance handling only, XAI only, and the full framework. This allows the contribution of each component to be isolated and measured.

3.9. Implementation Details

All experiments were run in Python with standard scientific computing libraries and had a reproducible pipeline that held the distribution of classes constant and held the random seeds constant. Hyperparameters shown in Table 1, such as learning rate, batch size, transformer heads, dropout, focal-loss settings, and epochs, were optimised on a validation set, and the final model was chosen based on the highest validation F1-score to ensure that a minority of ransomware samples are highly generalised. Its framework takes the form of a transformer-based design combined with explainable AI, with a focus on high detection performance, computational efficiency, and explainability to deploy to practice in cybersecurity.

Table 1 Hyperparameter Table

Hyperparameter	Value	Description
Input feature dimension	64	Total handcrafted behavioural features used after preprocessing
Train/validation/test split	70/15/15	Stratified split to preserve class ratio
Cross-validation folds	5	Used for robustness assessment
Batch size	64	Number of samples processed per iteration
Number of epochs	100	Maximum training epochs
Early stopping patience	12	Stop if the validation F1-score does not improve
Optimizer	AdamW	Adaptive optimiser with decoupled weight decay
Initial learning rate	1.00E-04	Starting learning rate for model training
Learning-rate scheduler	Cosine	Gradual learning-

	decay	rate reduction during training
Weight decay	1.00E-05	Regularisation to reduce overfitting
Dropout rate	0.3	Regularisation inside transformer and classifier blocks
Transformer layers	4	Number of stacked encoder blocks
Attention heads	8	Multi-head attention heads per transformer layer
Hidden/embedding size	128	Internal representation dimension
Feed-forward dimension	256	Width of transformer feed-forward subnetwork
Activation function	GELU	Nonlinear activation in transformer blocks
Loss function	Focal loss	Handles class imbalance effectively
Focal loss alpha	0.25	Minority-class balancing factor
Focal loss gamma	2	Focuses learning on hard samples
Class weighting	Enabled	Penalises ransomware misclassification more strongly
Oversampling method	SMOTE	Applied only to the training data
Decision threshold	0.45	Tuned for higher ransomware recall
Gradient clipping	1	Stabilizes training
Random seed	42	Ensures reproducibility
SHAP background samples	100	For global explanation estimation
LIME samples per instance	5000	For local explanation generation

4. RESULTS

4.1. Experimental Setup and Evaluation Protocol

The proposed ransomware detection framework was tested by using a recent ransomware dataset that consists of 21,752 samples that include both benign and ransomware cases across a variety of families. The training dataset was purposely imbalanced with a ransomware-to-benign ratio of 1:4, but validation and testing datasets were maintained to be stratified and balanced. All models were trained by the same preprocessing pipelines and tested according to the same experimental conditions. Training, validation, and testing were split with a 70/15/15 split, respectively, and cross-validation was done with the 5-fold cross-validation to ensure robustness.

4.2. Performance Comparison with Baseline Models

The suggested model's performance is compared to popular machine learning and deep learning baselines in Table 2.

Table 2 Performance Comparison with Baseline Models

Model	Accuracy (%)	Precision	Recall	F1-score	Specificity	ROC-AUC	FPR
Random Forest	91.84	0.902	0.887	0.894	0.936	0.945	0.064
XGBoost	93.12	0.918	0.901	0.909	0.948	0.956	0.052
SVM	89.76	0.876	0.861	0.868	0.918	0.927	0.082
MLP	92.45	0.907	0.892	0.899	0.942	0.951	0.058
1D-CNN	94.03	0.925	0.914	0.919	0.955	0.962	0.045
LSTM	94.88	0.934	0.921	0.927	0.961	0.968	0.039
Transformer (baseline)	95.42	0.941	0.928	0.934	0.964	0.971	0.036
Proposed Model	97.18	0.962	0.951	0.956	0.978	0.986	0.022

The proposed model achieves the highest performance across all evaluation metrics. It notably lowers the false positive rate to 2.2%, which is crucial for real-world deployment, and increases accuracy by around 1.76% over the transformer baseline. The recall value of 95.1% indicates strong capability in detecting ransomware instances, even under imbalanced training conditions.

4.3. Impact of Class Imbalance Handling

To evaluate the effectiveness of the imbalance-aware training strategy, experiments were conducted with and without focal loss and oversampling techniques. The results are shown in Figure 2.

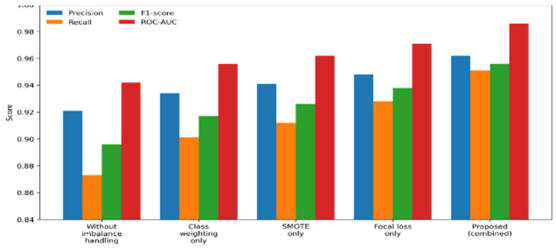

Figure 2: Effect of Class Imbalance Handling

The findings show that the most effective combination is focus loss with oversampling and class weighting. An increase from 87.3% to 95.1% in recall verifies the model's growing sensitivity to ransomware samples without sacrificing accuracy.

4.4. Ablation Study

An ablation study was conducted to quantify the contribution of each component of the proposed framework. The results are summarised in Figure 3.

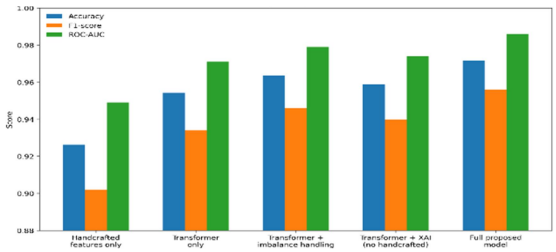

Figure 3 Ablation Study

The full model achieves the best performance, indicating that each component contributes positively. While the transformer captures intricate feature interactions, handcrafted features improve interpretability and early-stage detection. The imbalance-aware training significantly improves minority class detection.

4.5. Confusion Matrix Analysis

The confusion matrix for the proposed model is shown in Figure 4. Out of the total test samples, the model correctly classifies the majority of ransomware and benign instances.

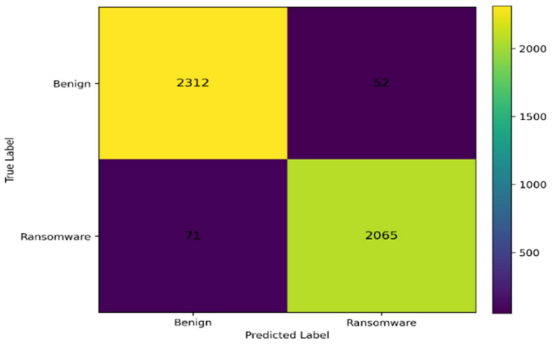

Figure 4: Confusion Matrix

True Positives (TP): 2,065, True Negatives (TN): 2,312, False Positives (FP): 52, False Negatives (FN): 71 . The low number of false positives indicates that the system minimises unnecessary alerts, while the small number of false negatives ensures effective ransomware detection.

4.6. ROC Curve Analysis

Figure 5 illustrates the ROC curves for all models. An AUC of 0.986 is achieved by the proposed model, surpassing all baselines. Superior discrimination capability is demonstrated by the curve's consistent elevation above other models across all threshold values.

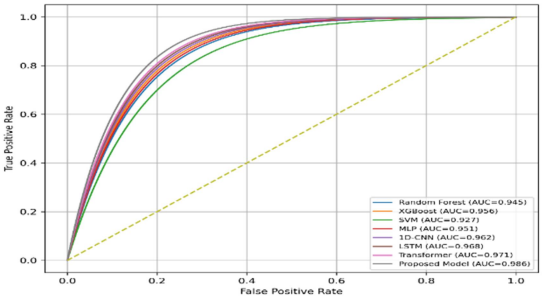

Figure 5: ROC Analysis

4.7. Computational Efficiency

Table 3 presents the computational efficiency of the models.

Table 3 *Computational Performance Comparison*

Model	Inference Time (ms/sample)	Parameters (Millions)	Memory (MB)
Random Forest	4.8	—	120
XGBoost	5.3	—	135
LSTM	9.6	1.2	210
Transformer	11.2	2.1	280

Proposed Model	**12.4**	**2.4**	**310**

Although the proposed model has slightly higher computational overhead than traditional models, it remains suitable for near real-time deployment. The trade-off is justified by the significant improvement in detection performance and interpretability.

4.8. Explainability Results

The explainability module provides both global and local insights into model decisions. Figure 6 summarises the top contributing features identified using SHAP.

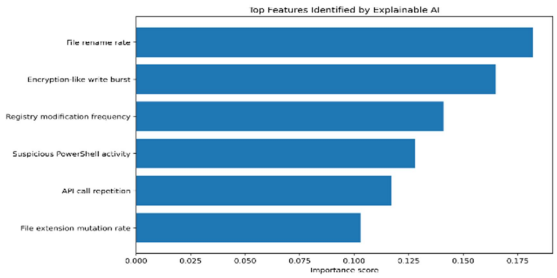

Figure 6: *Top Contributing Features*

Identified by explainable AI

The results show that the model heavily relies on behavioural characteristics that correspond with known ransomware tendencies. High file renaming rates and encryption bursts are reliable indicators of ransomware activity, according to SHAP research. Moreover, attention visualisation indicates that the model focuses on file-system and process-level feature interaction dependency cross-domain behavioural dependencies.

5. DISCUSSION

Semantic behavioural characteristics, transformer-based dependency learning, and imbalance-aware optimisation are three qualities that the suggested framework appears to combine well, according to the results. The handcrafted features capture operation signatures highly characteristic of ransomware, such as rapid file renaming, registry tampering, repeated API calls, and write bursts with encryption characteristics. [23]. These indicators possess a robust foundation that enables them to be identified regardless of the disorder or incompleteness of the raw input. In contrast, the transformer encoder provides performance improvements by modelling cross-feature interactions that simpler classifiers are prone to miss, especially when it comes to ransomware behaviour, which may involve coordinated process, file-system and network patterns. [24].

This is an enhancement of the baseline transformer, which proves that handcrafted behavioural cues are still beneficial even in the deep learning environment. [25]. This is important because ransomware is a behaviour-sequencing problem as well as a pattern-recognition challenge, wherein little but important activity sequences can accumulate into a trail of a malevolent operation. The ablation experiment also indicates that training based on the imbalance improves the minority-class recall significantly, which is essential to the practical deployment since ransomware events are both infrequent and expensive. Additionally, a lower false positive rate suggests that the model is not overreacting to aggressive administrative actions, but rather to benign ones.

The framework's practicality is further substantiated by the explainability results. The model is more suitable for security analysts who require justification before acting, as SHAP and attention visualisation have consistently identified features that are consistent with known ransomware tactics. One of the limitations of the current study is that the analysis will be performed on an edited dataset and balanced imbalance condition; real endpoint traffic would add more noise, hidden variants, and adversarial evasion examples. Therefore, cross-dataset validation, online training, and testing against dynamic ransomware families in operational business systems must be the focus of future study.

6. CONCLUSION

The explainable, imbalance-aware deep learning model demonstrated high performance, low false positives, and a lack of decision transparency in identifying ransomware using handcrafted behavioural features, transformer-based representation learning, and focal-loss-based optimisation. The key contribution of the work is an effective ransomware detection architecture, which not only enhances the accuracy of classification but also offers interpretable evidence in the form of XAI, which is much more appropriate to security operations, as well as to the deployment of the implementation in practice. The framework's online testing on live endpoints and enterprise-scale traffic, adversarial robustness, and cross-dataset validation should be the primary focus of future research.

REFERENCES:

[1] M. Niazi, A. M. Saeed, M. Alshayeb, S. Mahmood, and S. Zafar, "A maturity model for secure requirements engineering," Comput. Secur., vol. 95, p. 101852, Aug. 2020, doi: 10.1016/j.cose.2020.101852.

[2] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[3] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," 2017, arXiv. doi: 10.48550/ARXIV.1705.07874.

[4] A. Wang and X. Gao, "A variable scale case-based reasoning method for evidence location in digital forensics," Future Gener. Comput. Syst., vol. 122, pp. 209–219, Sep. 2021, doi: 10.1016/j.future.2021.03.019.

[5] P. Wang, L. Zong, and Y. Ma, "An integrated early warning system for stock market turbulence," Expert Syst. Appl., vol. 153, p. 113463, Sep. 2020, doi: 10.1016/j.eswa.2020.113463.

[6] E. Kokoris Kogias, D. Malkhi, and A. Spiegelman, "Asynchronous Distributed Key Generation for Computationally-Secure Randomness, Consensus, and Threshold Signatures," in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA: ACM, Oct. 2020, pp. 1751–1767. doi: 10.1145/3372297.3423364.

[7] A. Vaswani et al., "Attention Is All You Need," 2017, arXiv. doi: 10.48550/ARXIV.1706.03762.

[8] A. Gangwar, V. González-Castro, E. Alegre, and E. Fidalgo, "AttM-CNN: Attention and metric learning based CNN for pornography, age and Child Sexual Abuse (CSA) Detection in images," Neurocomputing, vol. 445, pp. 81–104, Jul. 2021, doi: 10.1016/j.neucom.2021.02.056.

[9] D. Wu, M. Nekovee, and Y. Wang, "Deep Learning-Based Autoencoder for m-User Wireless Interference Channel Physical Layer Design," IEEE Access, vol. 8, pp. 174679–174691, 2020, doi: 10.1109/ACCESS.2020.3025597.

[10] K. Zhang, Y. Wang, U. A. Bhatti, Y. Zhou, and M. Jin, "Enhanced ransomware attacks detection using feature selection, sensitivity analysis, and optimised hybrid model," J. Big Data, vol. 12, no. 1, p. 245, Nov. 2025, doi: 10.1186/s40537-025-01289-1.

[11] A. Galli, V. La Gatta, V. Moscato, M. Postiglione, and G. Sperlì, "Explainability in AI-based behavioural malware detection systems," Comput. Secur., vol. 141, p. 103842, Jun. 2024, doi: 10.1016/j.cose.2024.103842.

[12] F. Ullah, A. Alsirhani, M. M. Alshahrani, A. Alomari, H. Naeem, and S. A. Shah, "Explainable Malware Detection System Using Transformers-Based Transfer Learning and Multi-Model Visual Representation," Sensors, vol. 22, no. 18, p. 6766, Sep. 2022, doi: 10.3390/s22186766.

[13] R. G. G. Ortizo et al., "Extraction of Novel Bioactive Peptides from Fish Protein Hydrolysates by Enzymatic Reactions," Appl. Sci., vol. 13, no. 9, p. 5768, May 2023, doi: 10.3390/app13095768.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE, Oct. 2017, pp. 2999–3007. doi: 10.1109/ICCV.2017.324.

[15] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," Nat. Mach. Intell., vol. 2, no. 1, pp. 56–67, Jan. 2020, doi: 10.1038/s42256-019-0138-9.

[16] B. Jiang, P. Sun, and B. Luo, "GLMNet: Graph learning-matching convolutional networks for feature matching," Pattern Recognit., vol. 121, p. 108167, Jan. 2022, doi: 10.1016/j.patcog.2021.108167.

[17] T. R. Kumar, Mahesh. M. Kawade, G. K. Bharti, and G. Laxmaiah, "Implementation of Intelligent CPS for Integrating the Industry and Manufacturing Process," in AI-Driven IoT Systems for Industry 4.0, 1st ed., Boca Raton: CRC Press, 2024, pp. 273–288. doi: 10.1201/9781003432319-15.

[18] S. Song, S. Ma, X. Zhu, Y. Li, F. Yang, and L. Zhai, "Joint bandwidth allocation and task offloading in multi-access edge computing," Expert Syst. Appl., vol. 217, p. 119563, May 2023, doi: 10.1016/j.eswa.2023.119563.

[19] Z. U. A. Khan, M. A. Mughal, M. U. Hashmi, R. Sarwar, and I. U. Haq, "Lightweight and Explainable Early Ransomware Detection Using Dynamic API -Call Features and Ensemble Machine Learning," Eng. Rep., vol. 8, no. 4, p. e70734, Apr. 2026, doi: 10.1002/eng2.70734.

[20] H. Huang, Z. Zhang, Y. Shen, M. Backes, Q. Li, and Y. Zhang, "On the Privacy Risks of Cell-Based NAS Architectures," in Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA: ACM, Nov. 2022, pp. 1427–1441. doi: 10.1145/3548606.3560619.

[21] R. Chen et al., "Salient feature extractor for adversarial defence on deep neural networks," Inf. Sci., vol. 600, pp. 118–143, Jul. 2022, doi: 10.1016/j.ins.2022.03.056.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA: ACM, Aug. 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[24] Palaniradja, K., and V. Balasubramanian. "Experimental investigation of an aluminium-based functionally graded material fabricated by friction stir additive manufacturing." Materials Research Express 13.1 (2026): 016504.

[25] A. Galli, V. La Gatta, V. Moscato, M. Postiglione, and G. Sperlì, "Explainability in AI-based behavioural malware detection systems," Comput. Secur., vol. 141, p. 103842, Jun. 2024, doi: 10.1016/j.cose.2024.103842.