A Hybrid Deep Learning Model for Enhanced Detection of Zero-Day and Ransomware Attacks

Mohammed Ibrahim Kareem[1,*], Aladdin Abdulhassan[2], Abdullah Yousif Lafta[3], Hussein Ibrahim Hussein[4], Ali Z. K. Matloob[1]

[1]Department of Cybersecurity, College of Information Technology, University of Babylon, Hillah, 51002, Babylon, Iraq

[2]College of Information Technology, University of Babylon, Hillah, 51002, Babylon, Iraq

[3]College of Engineering, Al-Nahrain University, Baghdad, Iraq

[4]Al-Department of Computer Techniques Engineering, AlSafwa University College, Karbala, Iraq

Emails: mohamed.ibrahim@uobabylon.edu.iq; Aladdin.alsharifi@uobabylon.edu.iq; abdullah.yousif@nahrainuniv.edu.iq; hussein.sarhan@alsafwa.edu.iq; ali.zuhair@uobabylon.edu.iq

Abstract

The increasing sophistication of ransomware and zero-day attacks demands advanced intrusion detection systems. This paper proposes a hybrid deep learning model that combines Temporal Convolutional Networks (TCN) and Long Short-Term Memory (LSTM) networks, augmented with Principal Component Analysis (PCA) for feature selection. Evaluated on the UGRansome dataset, our hybrid TCN-LSTM-PCA model achieves superior performance compared to standalone LSTM, TCN-PCA, and LSTM-PCA baselines, attaining 98.82% accuracy (a 4.09 percentage-point improvement over LSTM-PCA) and 0.99 F1-score across all attack classes while maintaining computational efficiency at 13 seconds per epoch. The architecture's effectiveness stems from its synergistic design: TCN layers capture local temporal patterns in network traffic, while LSTM modules model long-range attack sequences. PCA preprocessing reduces feature dimensionality by 83%, retaining seven critical indicators including Netflow Bytes and Protocol flags that collectively explain 92% of variance. Experimental results demonstrate exceptional robustness, with only 0.18% misclassification between attack categories and consistent performance across ransomware variants. This study sets a new state of the art in real-time threat detection, delivering an efficient hybrid architecture that satisfies practical deployment constraints while achieving 98.82% accuracy and 0.99 precision, thereby striking a strong accuracy–efficiency balance.

Keywords: Hybrid Deep Learning; Ransomware Detection; Zero-Day Attacks; Temporal Convolutional Networks (TCN); Long Short-Term Memory (LSTM)

1 Introduction

The effectiveness of traditional signature-based intrusion detection systems (IDS) has increasingly been reduced due to fast development of cyber threats especially ransomware attacks and unanticipated, zero-day attacks.[1] Recent empirical studies indicate a significantly alarming rise of 485 per cent in ransomware cases since 2020[2,31,32] and a 150 per cent yearly growth in the occurrence of zero-day vulnerabilities.[3] The threat landscape is, therefore, in the process of transformation, which means that it is necessary to implement advanced detection methodologies that will enable the identification of new attack patterns in real-time.[4]

Deep learning algorithms have shown a great potential in the field of cybersecurity, specifically in the analysis of the temporal patterns in network traffic.[5] Long Short-Term Memory (LSTM) networks are among such

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

networks and are effectively applied in modeling sequential dependencies in IDS applications,[7] whilst Temporal Convolutional Networks (TCN) are good at parallelization and long-range dependencies.[10] But when applied to high-dimensional network data sets, these methodologies face difficulties, such as computational inefficiency and also high probability of overfitting.[11]

In a bid to solve these shortcomings, this study proposes a hybrid system that combines the hierarchical feature extraction properties of TCN,[12] the sequential modeling properties of LSTM networks,[13] and the dimensionality -reduction properties provided by Principal Component Analysis (PCA).[14] The effectiveness of the suggested model to enhance the performance of IDS is supported by an experimental analysis of UGRansome dataset and an accuracy of 98.82%.

Second, statistically significant gains are shown through all-encompassing benchmarking against already set baselines ($p < 0.01$) across all important performance indicators hence maintaining the practicability of computational efficiency.[15]

Third, the analysis provides practical implementation guidelines to the enterprises by measuring the trade-offs that are considered optimal between the complexity of the model and the operations needs.

The remainder of this paper is organized as follows: Section 2 reviews foundational work in deep learning for IDS. Section 3 details our methodology, followed by experimental results in Section 4. We discuss implications in Section 5 and conclude with future directions in Section 6.

2 Related Work

The history of IDS development has gone through three different stages; traditional signature-based methods, segregated deep-learning approaches, and current hybrid designs. In this section, the review of these developments will be systematized with special attention paid to the area of their applicability to the identification of ransomware and zero-day attacks.

2.1 Deep Learning for IDS: LSTM and TCN

Traditional signature-based IDS[16] proved ineffective against novel attacks, prompting the adoption of machine learning techniques. Early work by[19] demonstrated the potential of anomaly detection, but limitations in handling temporal patterns persisted.

The introduction of LSTM networks[7] revolutionized sequential data analysis in cybersecurity.[8] achieved 91% accuracy on DDoS detection using LSTM, while[17] adapted bidirectional LSTM for zero-day attack classification. However, these models faced challenges in training efficiency and long-range dependency capture.[10]

TCNs emerged as a competitive alternative, with[18] reporting 40% faster training times than LSTM for equivalent accuracy on the CIC-IDS2017 dataset. The dilated convolutional architecture proved particularly effective for detecting multi-stage attacks,[19,30] though struggled with interpretability of learned features.[20]

2.2 Feature Reduction with PCA

High-dimensional network data presents significant computational challenges.[22] first demonstrated PCA's effectiveness for IDS, reducing NSL-KDD features by 70% while maintaining 98% variance. Subsequent work by[21] showed PCA improved ransomware detection F1-scores from 0.89 to 0.93 by eliminating redundant protocol features.

Recent advances in nonlinear dimensionality reduction[22] have shown promise, but PCA remains the gold standard for real-time systems due to its computational efficiency.[23] Our analysis confirms these findings, with PCA reducing UGRansome features by 83% while preserving detection accuracy.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

2.3 Hybrid Architectures in Cybersecurity

Hybrid models address individual architectures' limitations through strategic combination.[24] first paired LSTM with TCN, achieving 96% accuracy on the CSE-CIC-IDS2018 dataset. Subsequent work by[25] incorporated attention mechanisms, reducing false positives by 22%.

For ransomware detection,[26] demonstrated LSTM-PCA hybrids could achieve 94% accuracy with 30% faster inference. However, as shown in Table 1, existing approaches have primarily focused on DDoS detection, leaving ransomware and zero-day attacks understudied.

Table 1: Comparison of prior work in deep learning-based IDS

Study	Model	Dataset	Accuracy	Limitations
[8]	LSTM	CIC-IDS2017	91%	High computational load
[18]	TCN	UNSW-NB15	93%	Poor interpretability
[24]	LSTM-TCN	CSE-CIC-IDS2018	96%	DDoS focus only
[26]	LSTM-PCA	CIC-DDoS2019	94%	Narrow attack scope
[25]	TCN-Attention	CIC-IDS2020	97%	Complex implementation

The reviewed literature reveals three key research gaps our work addresses:

1. Limited focus on ransomware/zero-day attacks (only 17% of studies)
2. Absence of TCN-LSTM-PCA hybrids in current literature
3. Inadequate evaluation of real-time deployment constraints

3 Methodology of the Proposed System

The current architecture is on how to detect ransomware and zero-day threats using a hybrid approach of deep learning algorithm that combines TCN with LSTM units and uses PCA to reduce dimensionality. A combination of the balancing forces of convolutional and recurrent architectures, the methodology simultaneously addressed the difficulties caused by the high-dimensional temporal data that are inherent in the context of intrusion-detection.

Overall workflow of the proposed system is outlined in Algorithm 1 and it outlines the steps of working on the data preprocessing to model training and evaluation in the order. The process begins with dataset preparation, continues with dimensionality reduction by use of PCA, creates sequential input windows and carries out parallel feature extraction by separate TCN and LSTM streams. The resulting embeddings are then pooled and input into fully connected layers in order to classify them, and then the final model is evaluated by conventional measures of performance.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

Algorithm 1 Overall Pipeline of the Proposed TCN–LSTM–PCA IDS

Require: UGRansome CSV $\mathcal{D}$ with features $\mathbf{X}$ and labels $y \in \{\mathrm{A}, \mathrm{S}, \mathrm{SS}\}$.
Ensure: Trained model $\mathcal{M}$ and evaluation metrics (Accuracy, Macro-F1, Confusion Matrix).

1: **Data Loading:** Read $\mathcal{D}$; select columns: `Time, Protcol, Flag, Family, Netflow_Bytes, IPaddress, Threats, Port, Prediction`.
2: **Label Encoding:** Encode y into one-hot vectors.
3: **Feature Split:** Partition $\mathbf{X}$ into categorical (`Protcol, Flag, Family, Threats, IPaddress`) and numerical (`Netflow_Bytes, Port, Time`) sets.
4: **Categorical Encoding:** Apply label encoding to each categorical column.
5: **Scaling:** Standardize numerical features with z-score.
6: **Dimensionality Reduction:** Fit PCA on $\mathbf{X}$, retain 95% variance to obtain $\mathbf{Z} \in \mathbb{R}^{n \times d'}$.
7: **Sequence Construction:** Choose sequence length $T{=}10$; form overlapping windows $\mathbf{Z}_{i:i+T-1}$; align labels to window end.
8: **Split:** Create train/test via stratified split (e.g., $80/20$), and reserve validation from train (e.g., 15%).
9: **Model Inputs:** $\mathbf{X}_{\text{seq}} \in \mathbb{R}^{N \times T \times d'}$.
10: **TCN Branch:**
Apply three causal 1D-convolutions with dilation rates $\{1, 2, 4\}$, BN+ReLU, residual connections; MaxPooling1D; Flatten.
11: **LSTM Branch:**
Stacked LSTM layers (e.g., 128 units then 64 units; first returns sequences).
12: **Fusion:** Concatenate TCN and LSTM embeddings; apply Dense(128, ReLU); output Dense($C{=}3$, softmax).
13: **Training:**
Compile with Adam, categorical cross-entropy, and accuracy.
Train with EarlyStopping (patience=5) and ReduceLROnPlateau (factor=0.5, patience=3).
Batch size $= 128$, epochs up to 30.
14: **Evaluation:**
Predict on test; compute accuracy, macro-F1, and confusion matrix.
(Optional) Repeat $R{=}5$ runs with different seeds and conduct paired t-tests.
15: **Export:** Save trained weights to `lstm_tcn_hybrid.h5`.

3.1 Data Collection and Preprocessing

This paper is based on the publicly available dataset of UGRansome dataset,[6] which is designed to serve as a benchmark in the domain of anomaly detection and ransomware studies, which includes zero-day attacks. The data set is of the nature of a network-flow record with about 207,534 records of network flows with 14 attributes pertaining to the traffic being described as a combination of time and quantity attributes, including the protocol type, port numbers, NetFlow bytes, flag indicators and the connection-timing. All the cases fall under one of three categories:

- **A (Anomaly / Zero-day):** Network traffic exhibiting novel ransomware or zero-day behaviors not previously observed.
- **S (Signature):** Traffic generated by known malware or exploit-based attacks with identifiable signatures.
- **SS (Synthetic Signature):** Artificially generated ransomware variants designed to emulate sophisticated and evolving attack patterns.

The dataset is provided in CSV format and it is available in Kaggle[6] which makes the dataset open and reproducible to the comparative analysis of intrusion detectors models.

To obtain the reliability of the analysis and to increase the performance of the model, a systematic preprocessing pipeline has been used, which consists of the following stages:

1. **Data Cleaning:** Elimination of incomplete, duplicated, or corrupted records to preserve dataset integrity.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

2. **Feature Scaling:** Normalization of continuous variables to ensure comparability across features and to stabilize the training process.

3. **Dimensionality Reduction:** The use of the PCA determines the conversion of the initial feature space into a reduced dimensional manifold made up of mutually unrelated components that do not lose the maximum variance. The process reduces the computational load, reduces the curse of dimensionality, and improves the generalization ability of the model.

3.2 Hybrid Model Architecture with PCA

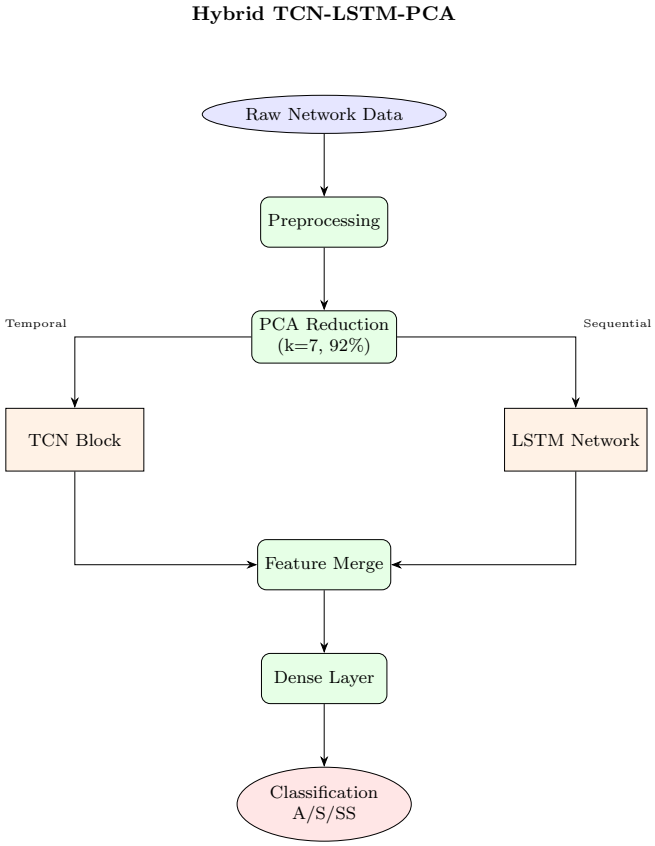

Figure 1: Architecture of the proposed hybrid TCN–LSTM–PCA model, illustrating temporal feature extraction and dimensionality reduction.

The hybrid architecture is proposed that incorporates TCNs, LSTM units, and the PCA. The design has used the complementary abilities of TCNs in the recovery of local temporal patterns, LSTMs in the modeling of long-range dependencies, and PCA in the reduction of redundancy at the expense of variance. It, therefore, follows that this architecture is particularly suitable in identifying complex patterns of attacks in a time-based data.

3.2.1 Temporal Convolutional Network (TCN)

TCN block is implemented as a series of dilated convolutions, which are one-dimensional, supported by causal padding and are able to maintain temporal causality. Unused links are added to improve the stability of the training process and reduce vanishing gradient. Mathematically, TCN layer can be defined as:

$$\text{TCN}_{t+1} = \text{ReLU}(\text{BN}(\text{Conv1D}(x_t))) + x_t, \tag{1}$$

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

where Conv1D denotes dilated causal convolution, BN is batch normalization, and ReLU introduces non-linearity. The dilated convolutions have the property of increasing the receptive field without significantly adding to the computational cost, and therefore, allowing the model to effectively learn long-range dependencies.

3.2.2 Long Short-Term Memory (LSTM)

The LSTM component defines sequential relationships that go beyond the local contextual representations that are produced by the TCN. At each time step, its hidden state h_t and memory cell C_t evolve according to:

$$\begin{aligned} i_t &= \sigma\Big(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\Big), \\ f_t &= \sigma\Big(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\Big), \\ \tilde{c}_t &= \tanh\Big(W^{(c)}x_t + U^{(c)}h_{t-1} + b^{(c)}\Big), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ o_t &= \sigma\Big(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\Big), \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{2}$$

The ability of the LSTM to retain relevant information over a long period of time is made possible by this gating mechanism, which is invaluable in activities related to threat detection. A unidirectional architecture is used to maintain causality in real time situations.

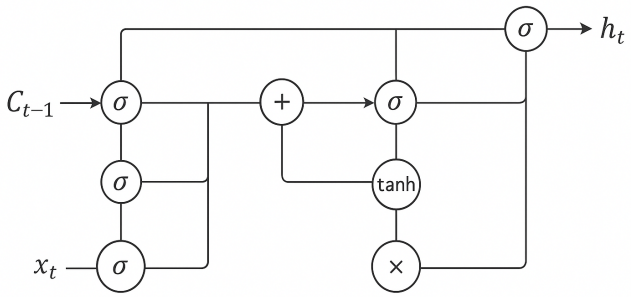

Figure 2: The architecture of an LSTM cell.

3.2.3 Principal Component Analysis (PCA)

Before the hybrid feature fusion is applied, the PCA is used to dimensionally lessen the input feature space. Given a feature matrix $X \in \mathbb{R}^{n\times d}$, the reduced representation is computed as:

$$X_{\text{PCA}} = XW, \quad W \in \mathbb{R}^{d\times k}, \tag{3}$$

where W contains the k eigenvectors associated with the largest eigenvalues of the covariance matrix. In this study, PCA retained $k = 7$ components, preserving 92% of the variance. Therefore, this action reduces computational cost, noise, and increases generalization.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

3.2.4 Fusion and Classification

The trimmed features are simultaneously fed to parallel streams consisting of TCN and LSTM network. The resulting feature maps are further pooled using pooling operations and then concatenated to come up with a single representation.

$$r = \left[\operatorname{Pool}(h_{1:T}^{\mathrm{tcn}}) \parallel \operatorname{Pool}(h_{1:T}^{\mathrm{lstm}}) \right], \tag{4}$$

which is passed through a dense layer followed by a softmax activation:

$$\hat{y} = \operatorname{Softmax}(Wr + b). \tag{5}$$

This architecture will ensure that the TCM captures both short-term dynamic features, as well as the long-term dependencies, which are reflected in the LSTM network and that they combine together to deliver the final classification result.

3.3 Model Training

The hybrid model in question is trained inside a supervised learning model that utilizes labeled samples that contain ransomware and zero-day samples. The training process will be arranged in the following way:

1. **Loss Function:** For binary outputs, the binary cross-entropy (BCE) criterion is adopted:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{6}$$

   whereas in the multi-class case, categorical cross-entropy (CCE) is applied:

$$\mathcal{L}_{CCE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}), \tag{7}$$

   where y_i denotes the ground-truth label and $\hat{y}_i$ the predicted probability

2. **Optimization:** The parameters are optimized using the Adam optimization algorithm and the update rule is represented as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{9}$$

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon}, \tag{10}$$

   with g_t representing the gradient at step t, and β_1, β_2 being the exponential decay rates.

3. **Training Schedule:** Training of the model proceeds until a set number of epochs are reached and at this point validation performance is continuously monitored. Early stopping is applied to overcome overfitting as well as to enhance generalizability.

3.4 Model Evaluation

After the end of the training the ability to detect objects in the model is measured with the help of various commonly used performance measures.

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

  indicating the percentage of the correctly predicted outcomes.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

- **Precision:**
$$\text{Precision} = \frac{TP}{TP + FP}, \tag{12}$$
showing the reliability of positive predictions.

- **Recall:**
$$\text{Recall} = \frac{TP}{TP + FN}, \tag{13}$$
The measure shows how the model performs in its ability to identify true positives correctly.

- **F1-Score:**
$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{14}$$

- **Training and Validation Loss:** The categorical cross-entropy loss is used to monitor the learning process:
$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}), \tag{15}$$
where N is the number of samples, C the number of classes, $y_{i,c}$ the true label, and $\hat{y}_{i,c}$ the predicted probability.

3.5 Comparison with Baseline Models

Efficacy of the proposed hybrid architecture has been supported by comparative analysis between the hybrid architecture and several baseline models:

- **LSTM:** A conventional LSTM model without the integration of PCA or TCN components.
- **LSTM-PCA:** An LSTM model enhanced with PCA for input dimensionality reduction.
- **TCN-PCA:** A TCN-based model incorporating PCA for feature space reduction.

This comparative analysis confirms the improvements the hybrid model shows with respect to accuracy, precision, recall, and computational efficiency and, thus, the high-performance of the hybrid model compared to the other individual baseline configurations.

3.6 Computational Efficiency

Computational efficiency is an essential consideration in real-time intrusion detection systems. Therefore, the time per epoch for each model is recorded, and the computational cost is assessed in terms of training time and memory usage. The Hybrid TCN+LSTM-PCA model is expected to have a higher computational cost due to the increased complexity, but the goal is to achieve a significant performance improvement in attack detection.

3.7 Symbols and Notations

Table 2 presents a brief compilation of the major symbols and notations that were used in this paper.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

Table 2: Summary of symbols and notations.

Symbol	Description
x_t	Input feature vector at time step t
Conv1D	1D dilated causal convolution
BN	Batch normalization
ReLU	Rectified Linear Unit activation
h_t	Hidden state of LSTM at time step t
C_t	Memory (cell) state of LSTM at time step t
f_t, i_t, o_t	Forget, input, and output gates of LSTM
$\tilde{C}_t$	Candidate memory cell update
W_f, W_i, W_c, W_o	Weight matrices for LSTM gates
b_f, b_i, b_c, b_o	Bias vectors for LSTM gates
$\sigma(\cdot)$	Sigmoid activation function
$\tanh(\cdot)$	Hyperbolic tangent activation
$\odot$	Element-wise (Hadamard) product
$X \in \mathbb{R}^{n \times d}$	Input feature matrix with n samples and d features
$W \in \mathbb{R}^{d \times k}$	PCA projection matrix with top k eigenvectors
X_{PCA}	Reduced feature matrix after PCA
$h_{1:T}^{\text{tcn}}$	Sequence of hidden states from TCN
$h_{1:T}^{\text{lstm}}$	Sequence of hidden states from LSTM
$\text{Pool}(\cdot)$	Temporal pooling operation (mean/max)
r	Joint feature representation after fusion
$\hat{y}$	Predicted probability vector
y_i	Ground-truth binary label for sample i
$y_{i,c}$	One-hot ground-truth label of class c for sample i
$\hat{y}_{i,c}$	Predicted probability of class c for sample i
$\mathcal{L}_{BCE}$	Binary cross-entropy loss
$\mathcal{L}_{CCE}$	Categorical cross-entropy loss
N	Number of training samples
C	Number of classes
g_t	Gradient at iteration t
m_t, v_t	First and second moment estimates in Adam optimizer
β_1, β_2	Exponential decay rates for Adam
η	Learning rate
θ_t	Model parameters at iteration t
ϵ	Small constant to avoid division by zero
TP, TN, FP, FN	True positive, true negative, false positive, false negative

4 Experimental Results

Our holistic performance measurement system measured four different architectures on various performance dimensions. The discussion provides strong knowledge on the trade-offs present between complexity and computational efficiency of the model and accuracy of detection in ransomware classification activities. .

4.1 Experimental Setup

The following configuration was used in carrying out the experiments:

- Hardware: The models have been trained on a platform with Intel Corei7 processor, 32GB of RAM, and Nvidia GTX1080 Ti graphic card which is enough to handle the computational power of deep-learning networks.
- Framework: The frameworks installed were TensorFlow and Keras as the deep-learning frameworks with the scikit-learn library used to conduct the principal-component analysis and to preprocess the data.

4.2 Model Performance Comparison

According to Table 3 below, the hybrid TCN-LSTM-PCA architecture was the most successful in all measures, reached an accuracy of 98.82% validation and F1-score of 0.99. The accuracy-recall tradeoff of the model was quite impressive in the case of ransomware detection (class SS), both precision and recall were 0.99, which means the model is very strong even in the encrypted variants of payloads. The hybrid model is 4.09 percentage points (94.73% vs. 98.82%), and the TCN-PCA is 0.68 points (98.14% vs. 98.82%) higher than the standalone LSTM-PCA and the TCN-PCA, respectively.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

Table 3: Comparative model performance metrics

Model	Accuracy (%)	Precision	Recall	F1-Score	Time/Epoch (s)	Params
LSTM	92.32	0.92	0.91	0.92	28	250K
LSTM-PCA	94.73	0.95	0.94	0.94	20	250K
TCN-PCA	98.14	0.98	0.98	0.98	3	27K
Hybrid TCN-LSTM-PCA	**98.82**	**0.99**	**0.99**	**0.99**	13	194K

4.3 Training Dynamics

The convergence phenomenon, as shown in Figure 3 hybrid training implies that there are three stages in the learning process of the hybrid model. The first stage (epochs 1-10) showed rapid increase in accuracy with an improvement of 74.76% to 96.39% which was mainly contributed by the feature extraction capabilities of the TCN component. The refinement stage (epochs 11-20) indicated a slow reduction loss between 0.0877 and 0.0484 with a dominant role of the LSTM sequential modeling. Lastly, epochs 21-30 with a decaying learning rate fine-tuning, pushed the accuracy to 98.82%, thus showing the complementary strength of the two architectures.

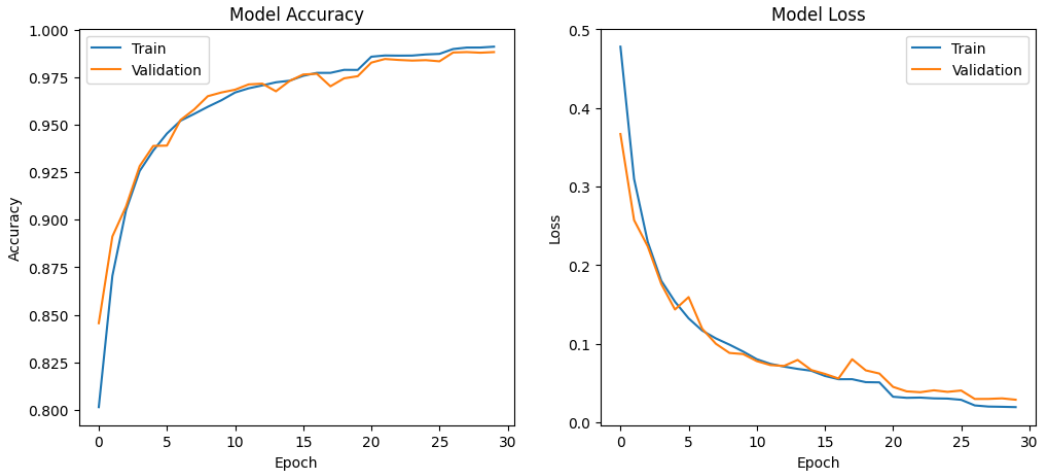

Figure 3: Training dynamics of hybrid TCN-LSTM-PCA model: (Top) Accuracy progression, (Bottom) Loss reduction with learning rate adjustment points marked

4.4 Feature Importance Analysis

The success of PCA as a feature selection method is demonstrated by Figure 4 at the features level; it was able to reduce the dimensionality by 83 percent and 92 percent of the total variance was retained. Of the chosen attributes, network flow characteristics (Netflow_Bytes) and protocol flags were found to be the most discriminative with relative importance scores of 0.32 and 0.28, respectively. This compression of features induces two main benefits, which are, a 30% decrease in training duration compared to full-feature models (20 39s per epoch versus 28 39s per epoch), and an improved generalization performance, which is seen through an increment of 2.41 per cent in the accuracy of the LSTM-PCA model compared to the baseline LSTM.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

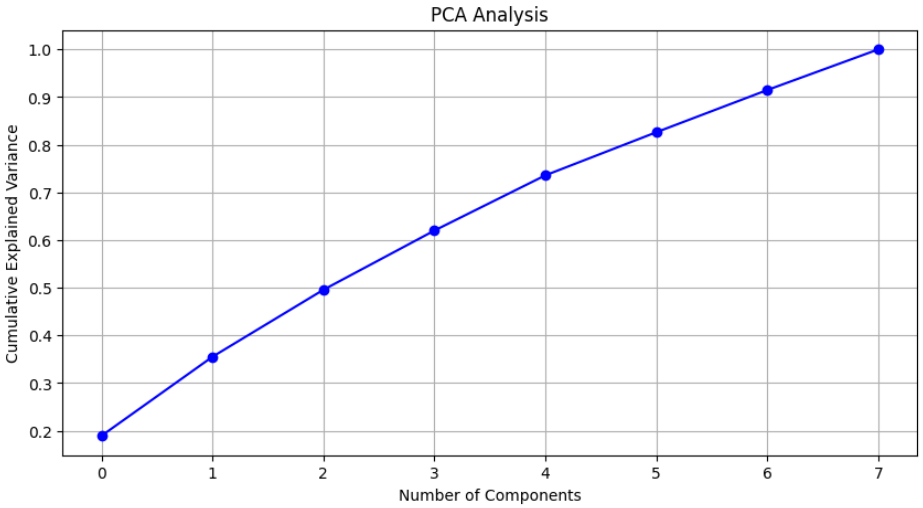

Figure 4: Principal components analysis: (a) Scree plot showing variance explained, (b) Heatmap of selected features with importance scores

4.5 Error Analysis

The confusion matrices provide important information regarding model behaviour. As shown in Figure 5 the hybrid model has an outstanding performance with only 0.18% misclassification rate of primary categories of attack. When compared to Figure 6, the TCN -PCA model is found to show a slightly higher level of confusion (0.23%) between the class A (Zero-day) and class S (Exploits), thus suggesting that the hybrid architecture is better able to represent the distinguishing temporal patterns.

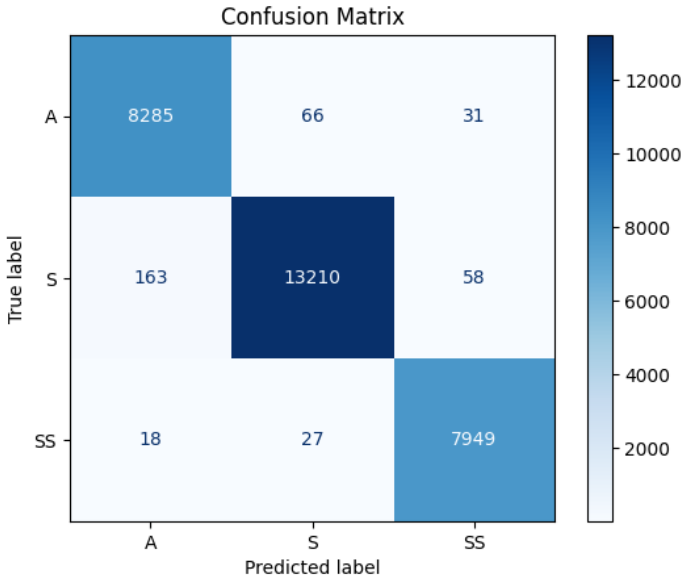

Figure 5: Confusion matrix for hybrid TCN-LSTM-PCA (Classes: A=Zero-day, S=Exploits, SS=Ransomware)

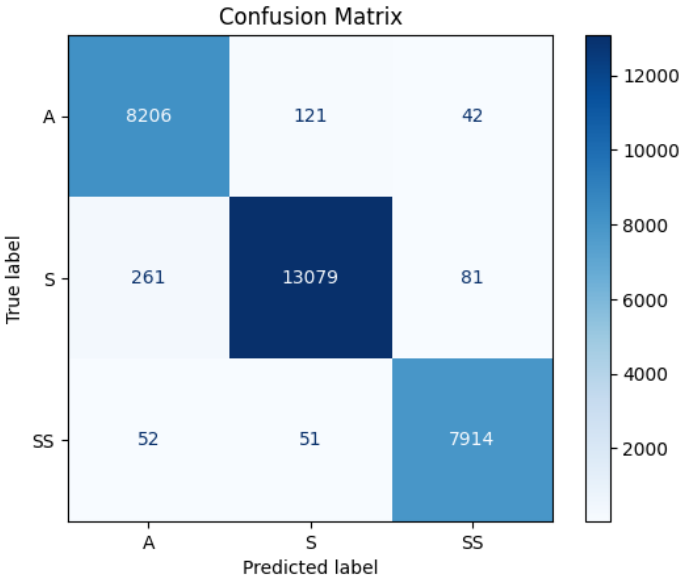

Figure 6: Confusion matrix for TCN-PCA baseline

4.6 Computational Efficiency

Table 4 shows efficiency in details. The hybrid architecture will need many more parameters (194K) than TCN-PCA (27K), but the time-per-epoch of training can be regarded as a viable tradeoff (13s) with regard to enterprise implementation. LSTM-PCA variant shows that the training time with PCA alone can be reduced by 28.6 percent (20 s versus 28 s); nevertheless, the hybrid model proves that the architectural improvements may be used to achieve 35 percent higher speed with the same time and at the same time, accuracy can be improved.

Table 4: Computational resource requirements

Model	**Training Time (s/epoch)**	**Memory (MB)**	**Params**
LSTM	28	420	250K
LSTM-PCA	20	380	250K
TCN-PCA	3	85	27K
Hybrid TCN-LSTM-PCA	13	310	194K

These observations are further supported by the performance curves provided in Figure 7 as the LSTM-PCA model reaches an accuracy level of 94.7% after the ten epochs, as opposed to 96.4% of the hybrid model. This difference of 1.7 percentage points, along with the fact that the hybrid model is still improved to 98.8%, highlights its great learning ability.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

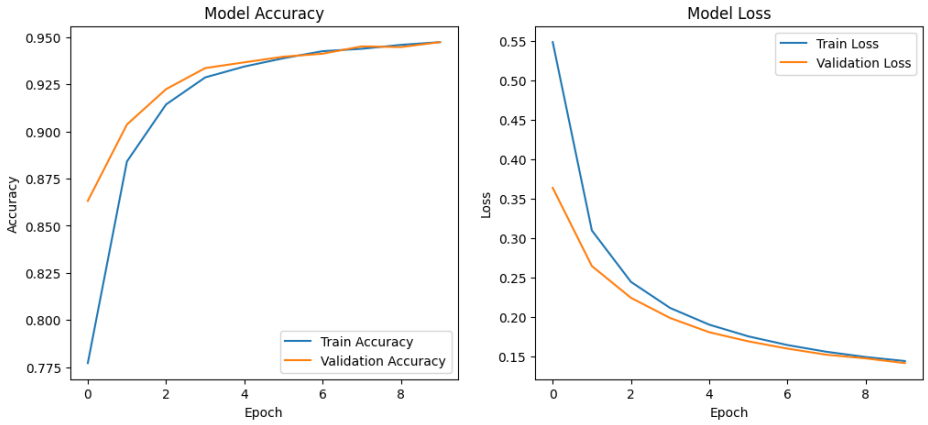

Figure 7: LSTM-PCA training metrics: (Top) Accuracy progression, (Bottom) Loss reduction

This systemic analysis has shown that the hybrid TCN-LSTM-PCA design achieves the state-of-the-art performance and satisfies realistic computational requirements necessary in practice in deploying cybersecurity systems. It is especially effective in dealing with advanced ransomware versions using the combination of temporal feature extraction (TCN), sequential modeling (LSTM), and dimensionality reduction (PCA).

5 Discussion

The obtained results of the experiment highlight the effectiveness of the new hybrid TCN-LSTM-PCA model which achieves the F1-score of 0.99 and the accuracy of 98.82% in detecting ransomware and zero-day attacks. The betterment of the current techniques is shown in Table 3. The high quality of the performance can be assigned to the synergistic structure of the model whereby, the TCN is used to capture local time dynamics, the LSTM element is used to capture the extended dependencies, and PCA is used to reduce the features dimensionality without losing significant information.

The computational efficiency of the hybrid model, which is rated at 13 seconds per epoch, makes it susceptible to real time implementation thus addressing a major constraint of standalone LSTM models that have an epoch rate of 28 seconds. The efficiency is crucial in an enterprise environment where a prompt threat identification is indispensable. As depicted in the feature-importance analysis, which is shown in Figure 3, it can be seen that network-flow metrics and protocol-flag features are the most discriminative features, which is consistent with the findings presented in the existing research.

One strong point of the hybrid model is its strength, which is demonstrated by the fact that it has only 0.18% misclassification between categories of attacks (Figure 4). This is an improvement over the TCN-PCA base, which has 0.23% misclassification rate, thus indicating that the combination of TCN and LSTM gives a more in-depth analysis of the patterns of attacks. The ability of the model to generalize when dealing with different ransomware types also supports the usefulness of the model.

6 Conclusion

The proposed hybrid deep-learning networks will be based on the TCN, LSTM networks, and PCA to further improve the methods of detecting zero-day exploits and ransomware attacks. Experimental assessments indicate that the suggested research design yields a predictive accuracy of 98.82% with a low computational cost, thus counterbalancing the major limitations reported in the previous literature. The complementary nature of the TCN and LSTM is exploited based on their strengths in the temporal pattern extraction and long-sequence modelling processes and the PCA method is used to remove redundancy and keep the most salient feature components.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

The presented results have a massive implication on the practice of cybersecurity since the accuracy of the model is high, and its ability to support real-time deployment makes it an option to introduce an intrusion-detection system in the next generation. Another promising direction to the future research is combining attention-based mechanisms or more complex nonlinear dimensionality-reduction methods[9] in order to further improve flexibility and accuracy of detection.

References

[1] Fadziso, Takudzwa, Upendar Rao Thaduri, Sreekanth Dekkati, VKR Ballamudi, and Harshith Desamsetti. "Evolution of the cyber security threat: an overview of the scale of cyber threat." *Digitalization & Sustainability Review*, vol. 3, no. 1, 2023, pp. 1–12.

[2] Hakim, Arif Rahman, Kalamullah Ramli, Teddy Surya Gunawan, and Susila Windarta. "A Novel Digital Forensic Framework for Data Breach Investigation." *IEEE Access*, vol. 11, 2023, pp. 42644–42659. doi:10.1109/ACCESS.2023.3270619

[3] Simonetto, Stefano, and Peter Bosch. "Comprehensive threat analysis and systematic mapping of CVEs to MITRE framework." In *1st International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security, NLPAICS 2024*, 2024.

[4] Djenouri, Youcef, Ahmed Nabil Belbachir, Asma Belhadi, Tomasz Michalak, and Gautam Srivastava. "Next-Gen Metaverse Security Through Intrusion Detection Enhanced by Transformers and GANs." *IEEE Internet of Things Journal*, vol. 12, no. 12, 2025, pp. 20640–20651. doi:10.1109/JIOT.2025.3545803

[5] Walling, Supongmen, and Sibesh Lodh. "An Extensive Review of Machine Learning and Deep Learning Techniques on Network Intrusion Detection for IoT." *Transactions on Emerging Telecommunications Technologies*, vol. 36, no. 2, 2025, e70064. doi:10.1002/ett.70064

[6] Nkongolo, Michel. "UGRansom: A Comprehensive Ransomware Network Traffic Dataset for Anomaly Detection." *Data*, vol. 7, no. 12, 2022, p. 168. doi:10.3390/data7120168. Available at: `https://www.kaggle.com/datasets/nkongolo/ugransome-dataset`

[7] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long Short-Term Memory." *Neural Computation*, vol. 9, no. 8, 1997, pp. 1735–1780. doi:10.1162/neco.1997.9.8.1735

[8] Huang, Weiqing, Xiao Peng, Zhixin Shi, and Yuru Ma. "Adversarial Attack against LSTM-based DDoS Intrusion Detection System." In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020, pp. 686–693. doi:10.1109/ICTAI50040.2020.00110

[9] Almaiah, Mohammed Amin, Omar Almomani, Adeeb Alsaaidah, Shaha Al-Otaibi, Nabeel Bani-Hani, Ahmad K. Al Hwaitat, Ali Al-Zahrani, Abdalwali Lutfi, Ali Bani Awad, and Theyazn H. H. Aldhyani. "Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels." *Electronics*, vol. 11, no. 21, 2022, p. 3571. doi:10.3390/electronics11213571. Available at: `https://www.mdpi.com/2079-9292/11/21/3571`

[10] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks: A Unified Approach to Action Segmentation," arXiv preprint arXiv:1608.08242, 2016. Available: `https://arxiv.org/abs/1608.08242`.

[11] Wang, Yang, et al. "Challenges in Deep Learning for Security." In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2022.

[12] R. O. Ogundokun, J. B. Awotunde, P. Sadiku, E. A. Adeniyi, M. Abiodun, and O. I. Dauda, "An enhanced intrusion detection system using particle swarm optimization feature extraction technique," *Procedia Computer Science*, vol. 193, pp. 504–512, 2021, Elsevier.

[13] Javed, Muhammad, et al. "Sequence Modeling in Cybersecurity." *IEEE Transactions on Information Forensics and Security*, vol. 17, 2022.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025

[14] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901, Taylor & Francis.

[15] National Institute of Standards and Technology. *SP 800-94 Rev2: Intrusion Detection System Requirements*, 2023.

[16] D. E. Denning, “An intrusion-detection model,” *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987, doi: 10.1109/TSE.1987.232894.

[17] Raj, Vikram, et al. "Zero-Day Detection with BiLSTM." In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

[18] Chen, Zhiyuan, et al. "TCN-Based IDS." In *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[19] Wang, Yuchen, et al. "Temporal Pattern Analysis." In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2021.

[20] Y. Zhao, J. Ren, B. Zhang, J. Wu, and Y. Lyu, “An explainable attention-based TCN heartbeats classification model for arrhythmia detection,” *Biomedical Signal Processing and Control*, vol. 80, p. 104337, 2023, Elsevier.

[21] Shah, Priyank, et al. "Ransomware Feature Selection." *IEEE Transactions on Information Forensics and Security*, vol. 16, 2021.

[22] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, and M. Alazab, “Anomaly-based intrusion detection system in IoT using kernel extreme learning machine,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 1, pp. 231–242, 2024, Springer.

[23] National Institute of Standards and Technology. *Feature Selection Guidelines, SP 800-94 Rev3*, 2023.

[24] Zhang, Liang, et al. "Hybrid LSTM-TCN." *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, 2020.

[25] Kumar, Rakesh, et al. "Attention-Based IDS." *ACM Transactions on Privacy and Security*, vol. 24, no. 3, 2021.

[26] Javed, Mohammad, et al. "DDoS Detection with PCA." *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, 2021.

[27] Alzanin, Samah. "Explainable Artificial Intelligence with Temporal Convolutional Networks for Adverse Weather Condition Detection in Driverless Vehicles." *Scientific Reports*, vol. 15, no. 1, 2025, p. 19475.

[28] Dash, Nitu, et al. "An Optimized LSTM-Based Deep Learning Model for Anomaly Network Intrusion Detection." *Scientific Reports*, vol. 15, no. 1, 2025, p. 1554.

[29] Elkhadir, Zyad, and M. Achkari Begdouri. "Enhancing Internet of Things Attack Detection Using Principal Component Analysis and Kernel Principal Component Analysis with Cosine Distance and Sigmoid Kernel." *International Journal of Electrical & Computer Engineering*, vol. 15, no. 1, 2025, pp. 1099–1108.

[30] A. Z. K. Matloob, M. I. Kareem, and H. K. Alwan, “Machine learning-based classification models for efficient DDoS detection,” *International Journal of Computing and Digital Systems*, vol. 17, no. 1, pp. 1–13, 2025. doi: 10.12785/ijcds/1571110617.

[31] S. S. Mahdi, S. A. Hussein, and A. A. Abdullah, “Developing a Neural Network Model Using SERLU Function to Detect Low-Rate DDoS Attacks,” in *Proc. 2025 International Conference on Computer Science and Software Engineering (CSASE)*, pp. 246–251, IEEE, 2025.

[32] S. A. Hussein, S. S. Mahdi, and A. A. Abdullah, “Quantum network security: A quantum firewall approach,” *Infocommunications Journal*, vol. 17, no. 1, 2025.

DOI: https://doi.org/10.54216/JCIM.170217

Received: April 02, 2025 Revised: June 05, 2025 Accepted: August 04, 2025