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

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

ENHANCING CLOUD NETWORK INTRUSION DETECTION USING SMOTE, TRANSFORMER ENCODER AND ADVERSARIAL HARDENING

SAMRAT KRISHNA GADDAM [1], A. SRINAGESH [2]

[1] Research Scholar, Acharya Nagarjuna University, Department of Computer Science and Engineering, India

[2] Professor, RVR & JC College of Engineering, Department of Computer Science & Business Systems, India

E-mail: [1]samratkrishnagaddam@gmail.com, [2]asrinagesh@gmail.com

ABSTRACT

As the utilization of Cloud Computing is growing rapidly it encounters more challenging and evolving cybersecurity threats and zero-day attacks. In order to detect intrusion detection more Intelligent and adaptive deep learning models are crucial. With the goal to identify anomalies in cloud network traffic, this study compares three sequential learning architectures the conventional LSTM, its bidirectional extension (Bi-LSTM), and Projected LSTM Classifier (PLC) and proposes a novel Transformer Encoder with Adversarial Hardening (Transformer+AT). Two popular datasets, CICIDS2018 and CIC-IoT2023, were used for the experiments. During preprocessing, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to concentrate on the extreme class imbalance that can be prevalent in real-world cloud data. Amid the baselines models PLC has acheieved an accuracy of 96% on CICIDS2018 but has significantly dropped to 82% on CIC-IoT2023. Show cases a overview gap which is addressed by the proposed model.

The proposed Transformer+AT achieve 96.05% on CICIDS2018 and 96.1% on CIC-IoT2023 shows significant stable performance among the two datasets different from PLC as it has dropped from 96% to 82% on CIC-IoT2023. Decisively Transformer+AT shows outstanding adversarial robustness by 0.06% drop of accuracy under FGSM attack on CICIDS2018 and 8.39% on CIC-IoT2023 when compared to the 20-25% drop for baseline models. These findings disclose that when Transformer architectures with SMOTE and adversarial hardening are combined, they improve generalization and robustness among varied environments of Cloud and IoT.

Keywords - *LSTM, Bi-LSTM, Projected LSTM, Transformer, Adversarial Training, FGSM, Cloud Security, Intrusion Detection*

1. INTRODUCTION

The motivation behind this study is the challenge of maintaining reliable detection capabilities across a wide range of different cloud and IoT network environments, while also being resistant to adversarial evasion attacks, which has direct implications for cloud service providers, enterprise security operations centres and IoT platform operators [3]. Nowadays, cloud computing is considered one of the key components of digital transformation and provides affordability, scalability, and elasticity to modern businesses [1]. However, this shift in thinking also creates an expanded surface area, and thus a greater risk from highly sophisticated and dynamic cyberattacks [2]. Intelligent Intrusion Detection Systems (IDS) intended for the cloud environment are significant as cyberattacks turn increasingly more complicated [3].

Even though proficient against known threats, conventional signature-based intrusion detection systems are not very much effectual at identifying obfuscated and zero-day attacks [4]. On the other hand, both deep learning (DL) and machine learning (ML) approach are superior at modeling identifying anomalies in network traffic [5]. Recurrent Neural Networks (RNNs), remarkably Long Short-Term Memory (LSTM) architectures, are widespread DL models as they are strong in sequential tendencies and learning abilities [6].

Accurate recognition of complicated intrusion patterns is carried out achievable by LSTM networks potential to preserve long-term

temporal information [7]. The capacity has been strengthened by enhancements like Bidirectional LSTM (Bi-LSTM), which process input sequences in both temporal directions and captures richer context [8]. With positive results, these architectures have been implemented efficiently in a number of cloud-based security applications [9].
Detection of intrusion has shown better results when advanced techniques are produced by combining various learning techniques [10].
Intrusions can be effectively detected using Bi-LSTM based on Explainable AI which detects intrusions effectively [11]
Class imbalance, in which benign traffic predominates attack instances leading to biassed learning, is a frequent issue in network intrusion detection [12]. With the goal to enhance model sensitivity and reduce misclassification of rare attacks, minority class samples are added using the Synthetic Minority Over-sampling Technique (SMOTE) [13]. SMOTE incorporation with deep learning models has been demonstrated through numerous studies to raise classification stability and accuracy [12].
Despite immense developments, an abundance of prior research studies just assess LSTM-based models on a single dataset, which hinders our knowledge of the extent to which they generalize [13]. In the real world, cloud networks frequently involve a variety of environments, such as Internet of Things traffic, requiring for models that are robust to different data distributions [15].
A combined model of CNN and RNN has produced the best results for detecting traffic anomalies in IoT networks [14].
As standard datasets like CICIDS2018 and CIC-IoT2023 contain realistic attack data and traffic behaviours, they are recurrently used to assess IDS models [15]. CIC-IoT2023 include traffic from varied IoT devices, making it ideal for testing model flexibility, while CICIDS2018 represents enterprise network behaviour [16].
Baseline models LSTM, Bi-LSTM, Projected LSTM (PLC) alongside of proposed Transformer+AT are evaluated . For optimal consistency in evaluation, all models are trained on both datasets using the same preprocessing, including SMOTE [12].
A hybrid model using Crow Search Optimization (CSO) and Random Forest (RF) are proposed as there is more need of strong defence from attacks on the data [18]. Datasets CICIDS-2017, UNSW-NB 15 are tested using T-Distributed Stochastic Neighbour Embedding and Random Forest Algorithm (T-SNERF) which gained more performance than Signature detection systems [19].
In accordance with our results, Bi-LSTM maintains a higher generalization capacity with 91% accuracy, while PLC achieves 96% accuracy on CICIDS2018 but falls to 82% on CIC-IoT2023. Experiments were carried out on CICIDS2017 and CICIDS2018 using six machine learning algorithms in which Decision Tree (DT) and Random Forest (RF) endure from over fitting [21].
Even though LSTM base models perform stronger on clean datasets, they are still susceptible to adversaral slusion attacks when a malicious actor uses slight perturbation in network traffic for bypassing the detection.
This is a critical unsolved problem as the current LSTM-based-IDSs cannot simultaneously learn to generalise across heterogeneous traffic distributions and to be adversarial resistant with direct impact on the cloud service provider and/or IoT platform operators. Inorder to overcome this the present study propose a Transformer Encoder with Adversarial Hardening (Transformer+AT) for enhanced feature learning using Multi-Head Self-Attention and for solidfying the model against evasio attacks FGSM based adversarial training is used. The generaliation gap in PLC on CIC-IoT2023 and adversarial vulnerability which is general in all LSTM architectures is addressed.
The following is a rationale for the proposed approach. Transformer architectures are based on the Multi-Head Self-Attention mechanism, which allows them to capture global feature relationships across all of the input, without the need to take into account temporal ordering constraints that are present in RNN-based models and cause a forward pass. It is hypothesized that this property can be used to achieve better generalisation over diverse traffic distributions that have different temporal statistics like enterprise and IoT traffic. We use adversarial training using FGSM as the hardening mechanism, as it is the only one that directly optimizes decision boundaries against gradient-based evasion attacks. Therefore, the SMOTE-balanced training, Transformer-based feature extraction, and adversarial hardening are all tackled in the same framework, solving

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

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

the three main challenges class imbalance, generalisation gap, and adversarial vulnerability.

2. METHODOLOGY

A controlled comparative experimental design was used, where four deep learning architectures were trained and tested using the same preprocessing procedures on two benchmark datasets, and thus, differences in performance could be directly attributed to the architecture.

The datasets, preprocessing techniques, and architectural layout of the proposed deep learning models for intrusion detection in cloud-enabled networks are outlined in this section. The aim is to assess and comparison, under identical experimental conditions, the generalisation capacities of LSTM, Bi-LSTM, Projected LSTM Classifier (PLC) and proposed Transformer+AT.

2.1 Datasets

The Canadian Institute for Cybersecurity curated two standard data sets, CICIDS2018 and CIC-IoT2023. Real-world traffic from an enterprise network which reflects both benign and attack behaviours is recorded by CICIDS2018 [16]. CIC-IoT2023, on the contrary hand, demonstrates the present heterogeneous cloud-IoT ecosystem by integrating traffic from multiple IoT devices, including MQTT, CoAP, and HTTP-based communications [22].

Multiple classes which represent different types of attacks, involving DDoS, brute force, infiltration, and botnets, have been included in each dataset. Severe class imbalance is a common disadvantage, nevertheless, which requires to be fixed to ensure equitable learning.

2.2 Preprocessing and SMOTE

The datasets go through a number of preprocessing steps prior to training: Feature selection and exclusion of non-numeric fields.Min-Max Normalization to scale features into a range of [0, 1] Label encoding of class labels into numerical values.

To deal with the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. SMOTE generates synthetic samples of minority classes by interpolating between accessible ones in feature space, as shown in Figure 1.

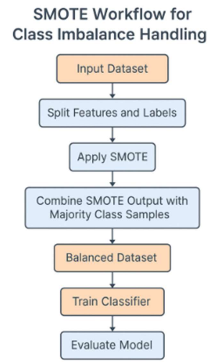


Figure1: SMOTE Workflow for Class Imbalance Handling

SMOTE enhances model sensitivity to underrepresented classes without duplicating data, reducing overfitting risks.

The test set was not modified, and SMOTE was only used on the training set to guarantee that the evaluation metrics represented real-life deployment scenarios.

2.3 Model Architectures

Using identical configuration three sequential deep learning models were implemented.

(a)Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is the modern variant of Recurrent Neural Networks (RNNs) which aims to resolve the vanishing gradient issue using the gated memory mechanisms. LSTM networks are especially useful in order to model time related dependencies in sequential networks traffic flows. Each LSTM unit has a forget gate, input gate, and output gate.

LSTM Equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \odot \tanh(C_t)$$

(b)Bidirectional LSTM (Bi-LSTM)

Bi-LSTM is a extension of LSTM by processing the sequence in both forward and

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

backward. Having this framework enables better understanding of network traffic patterns, in particular in attacks with delayed or hidden signatures.

(c) Projected LSTM Classifier (PLC)

The Projected LSTM Classifier (PLC) is a new architectural structure, which is outlined in this study, aimed at enhancing the feature representation before timely modeling.

As opposed to the classical LSTM models where the raw feature vectors are absorbed directly, the PLC also uses a learned dense projection layer before the LSTM block.This projection enhance learning capability and aligns feature spaces for better temporal modeling is shown in Figure-2.

Step 1: Feature Projection:

Introducing the input feature vector xt∈Rd, the projection layer changes it by represented in a latent each:

zt=ϕ(Wpxt+bp)

Step 2: Temporal Modeling

ztz, the projected vector is fed to the LSTM block:

ht=LSTM(zt)

Step 3: Classification

Final Prediction

ŷ=Softmax(WchT+bc)

The PLC architecture emulates successfully an embedding-type transformation similar to learned representation mapping and thus strengthens the temporal feature abstraction before sequential learning.

This architecture, as shown in Figure-2, is more effective in separating the classes, and better detection of the intrusion specially where there is an imbalance in the cloud traffic.

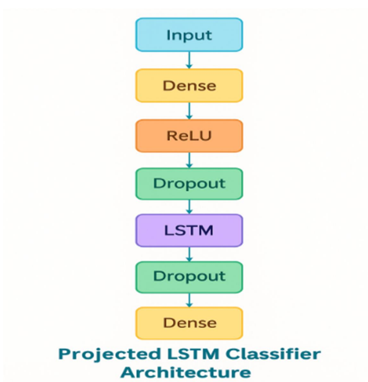


Figure 2: Projected LSTM Classifier Architecture

(d) Transformer Encoder Architecture

The Transformer Encoder replaces sequential recurrence with Multi-Head Self-Attention, enabling parallel processing of all input features simultaneously. The input feature vector is first projected into an embedding space using a dense projection layer (borrowed from PLC), followed by stacked Transformer Encoder blocks.

Each Transformer Encoder block consists of:

Step 1 — Multi-Head Self-Attention:

$\text{Attention}(Q,K,V) = \text{Softmax}(QK^T / \sqrt{dk}) \cdot V$

Step 2 — Add & Norm:

$\text{out1} = \text{LayerNorm}(x + \text{Attention}(x))$

Step 3 — Feed Forward Network:

$\text{FFN}(x) = \text{ReLU}(W_1x + b_1)W_2 + b_2$

Step 4 — Add & Norm:

$\text{out2} = \text{LayerNorm}(\text{out1} + \text{FFN}(\text{out1}))$

Final classification is performed via GlobalAveragePooling followed by a Softmax dense layer: $\hat{y} = \text{Softmax}(Wc \cdot \text{GAP}(\text{out2}) + bc)$

(e) Adversarial Training using FGSM

To harden the model against evasion attacks, the Fast Gradient Sign Method (FGSM) is applied during training. FGSM generates adversarial examples by adding a small perturbation in the direction of the loss gradient with respect to the input:

$x\_adv = x + \varepsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$

where ε = 0.01 controls perturbation strength, J is the categorical cross-entropy loss, and θ represents model parameters. During each training epoch, 30% of training samples are replaced with adversarial examples, forcing the model to learn robust decision boundaries. This adversarial training loop is applied on top of SMOTE-balanced data, ensuring both class balance and adversarial robustness.

2.4 Training Configuration

As the have to perform the same all the models are trained under the similar environment: Batch Size: 64, Epochs: 50, Loss Function: Categorical Cross-Entropy, Optimizer: Adam (learning rate = 0.001).

2.5 Evaluation Metrics

For evaluating the models standard classification metrics are used:

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

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

Where:

TP: True Positives, TN: True Negatives

FP: False Positives, FN: False Negatives

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

These metrics help assess the balance between detection capability and false alarms.

3. PROPOSED WORK

This section details the architecture and motivation behind the proposed Transformer Encoder with Adversarial Hardening (Transformer+AT), developed to boost the representation and learning capacities of traditional LSTM-based models, especially when coping of class-imbalanced and heterogeneous network traffic which is standard in cloud and Internet of Things settings.

3.1 Motivation

Although LSTM, Bi-LSTM, and PLC can perform well on structured data, like CICIDS2018, the algorithms have two shortcomings. First, the accuracy of PLC reduces considerably to 82% in the heterogeneous CIC-IoT2023 dataset, suggesting weak extrapolation to different IoT traffic. Second, any recurrent architecture is susceptible to adversarial evasion attacks, in which small input perturbations lead to misclassification of attacks as legitimate traffic.
Transformer+AT is proposed to overcome each of these limitations: Multi-Head Self-Attention learns global feature relationships and enhances the quality of generalization, independent of the type of traffic, and FGSM-based adversarial training makes the model resistant to evasion attacks.

3.2 Transformer+AT Architecture

Transformer+AT model has the following sequential components:
(a) Input Layer — SMOTE-balanced and normalized preprocessed data.
(b) Feature Projection Layer — Dense layer with ReLU activation which projects raw input into 64-dimensional embedding space: z = ReLU(Wp•x + bp)
(c) Transformer Encoder Blocks — Two stacked encoder blocks that have Multi-Head Self-Attention (4 heads), Feed Forward Network (128 units) with dropouts (0.1) and Layer Normalization.
(d) Global Average Pooling -dimension of sequences is reduced to a fixed-size vector.
(e) Classification Head -Dense layer (64 units, ReLU) followed by Softmax output layer.

3.3 Adversarial Hardening

At 30:70 ratio the FGSM adversarial examples are generated and mixed with clean samples at each epoch. This enables the model to learn verdict limitations that are tough to input perturbations, enabling the simulation of real-world evasion attacks on cloud.

3.4 Benefits of Transformer+AT

Capturing Global Features: All features are operated simultaneously using self attention which differs from LSMT sequential process,
Adversarial Robustness: A drop of 0.06% on CICIDS2018 and 8.39% CIC-IoT2023 under attack has reduced by FGSM training.
Generalization: Gained a 96.1% on CIC-IoT2023 a significant improvement of 14% which PLC has 82%.
SMOTE Synergy: Adversarial training on SMOTE-balanced data improves minority class robustness.

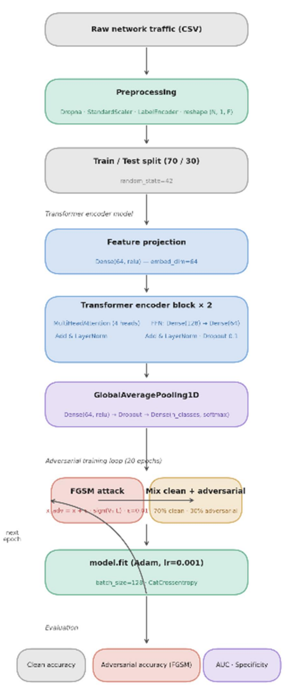


Figure 3: Architecture of Transformer + Adversarial Hardening model.

3.5 Adversarial Training using FGSM

The Fast Gradient Sign Method (FGSM) is used to train the model to increase its resistance to evasion attacks. By introducing infinitesimal perturbation in focus of the input in the perspective

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

of loss gradient adversarial examples are produced by FGSM.

$$x\_adv = x + \varepsilon \cdot \mathrm{sign}(\nabla_x J(\theta, x, y))$$

In which ε equals 0.01 regulates the magnitude of perturbation, J is the categorical cross-entropy loss, and θ symbolizes model parameters. 30% of the training samples are substituted by examples of training adversarial in each epoch of training convincing the model to attain flexible boundaries of decision. This adversarial training loop is trained over SMOTE-balanced data, which guarantees the balance between classes as well as adversarial robustness. The training process is summarized in Algotirhm-1.

Algorithm-1

INPUT: Dataset D, Labels Y, epsilon ε=0.01, adv_ratio=0.3
OUTPUT: Trained adversarially hardened Transformer model M
/* PREPROCESSING */
1. Load D (CICIDS2018 or CIC-IoT2023)
2. Drop null values from D
3. Encode labels Y using LabelEncoder
4. Scale features using StandardScaler → X_scaled
5. Apply SMOTE(X_scaled, Y) → X_balanced, Y_balanced
6. Reshape X_balanced → shape (N, 1, features)
7. Split → X_train, X_test, Y_train, Y_test (70:30)
/* BUILD TRANSFORMER MODEL */
8. Input Layer ← shape(1, features)
9. Projection Layer ← Dense(64, ReLU)
// Feature projection from PLC
10. FOR block = 1 to 2 DO:
a. attn ← MultiHeadAttention(heads=4)(x, x)
b. x ← LayerNorm(x + Dropout(attn))
c. ffn ← Dense(128,ReLU) → Dense(64)
d. x ← LayerNorm(x + Dropout(ffn))
11. x ← GlobalAveragePooling1D(x)
12. x ← Dense(64, ReLU) → Dropout(0.1)
13. Output ← Dense(num_classes, Softmax)
14.Compile M with Adam(lr=0.001), CategoricalCrossEntropy
/* ADVERSARIAL TRAINING LOOP */
15. FOR epoch = 1 to 20 DO:
a. n_adv ← int(len(X_train) * adv_ratio) // 30% adversarial
b. Select random indices idx from X_train
c. /* FGSM Attack */
gradient ← ∇x [J(M(X_train[idx]), Y_train[idx])]
X_adv ← X_train[idx] + ε · sign(gradient)
d. X_mixed ← concat(X_train, X_adv)
e. Y_mixed ← concat(Y_train, Y_train[idx])
f. Shuffle(X_mixed, Y_mixed)
g. Train M on X_mixed, Y_mixed (batch=128)
h. Evaluate M on X_val → record val_acc, val_loss
/* EVALUATION */
16. Y_pred ← M.predict(X_test) // Clean evaluation
17. X_test_adv ← X_test + ε · sign(∇x J(M(X_test), Y_test))
18. Y_pred_adv ← M.predict(X_test_adv) // Adversarial evaluation
19. Compute Accuracy, Precision, Recall, F1, AUC, Specificity
20. Report Clean Accuracy vs Adversarial Accuracy drop
21. RETURN M

4. RESULTS AND DISCUSSION

The proposed deep learning baseline architectures LSTM, Bi-LSTM, and the proposed Projected LSTM Classifier (PLC) and proposed Transformer+AT tested on two broadly used standard datasets, CICIDS2018 and CIC-IoT2023 in this section. Key performance metrics including accuracy, precision, and recall to decide all models efficiency are evaluated.

4.1 Performance on CICIDS2018

As shown in Table 1, the classification results on the CICIDS2018 dataset emphasize high baseline accuracy where PLC has a high score of 96, whereas the proposed Transformer+AT has a high score of 96.05., which reached an accuracy of 96%. This performance was marginally better than that of the standard LSTM achieved 95%. Model Accuracy 94% Precision 93% Recall and Bi-LSTM models, both of which achieved 95%. Model Accuracy 95 % Precision 93% Recall 94%.

TABLE 1 :PERFORMANCE METRICS ON CICIDS2018

Model	Performance Metrics				
	Accuracy (%)	Precision (%)	Recall	Avg Specificity	AUC
LSTM	95	94	93	90	75
Bi-LSTM	95	95	94	90	75
PLC	96	96	95	90	75
Transformer + AT(Proposed)	96.05	96	96	91.83	78

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

The dense projection layer, which transforms raw features into an improved informative representation before feeding them into the LSTM unit, enhances the performance of PLC baseline, and the proposed Transformer+AT goes further to boost detection with Multi-Head Attention and adversarial hardening.. This step improves the model ability to detect minute patterns of attack classes.
Compared to LSTM and BiLSTM and PLC, proposed Transformer+AT has achieved better learning performance, as shown by the learning curves in Figures 4-8, that converged quicker and had a lower final loss.
The PLC model excels on CICIDS2018 but underperforms on CIC-IoT2023.

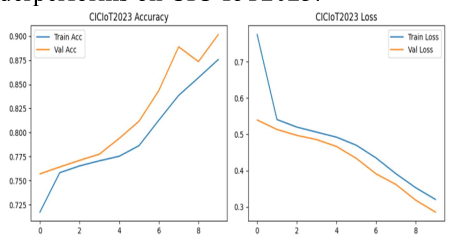


Figure 4: Training and Validation accuracy loss of LSTM on CICIDS2018 and CICIoT2023 datasets.

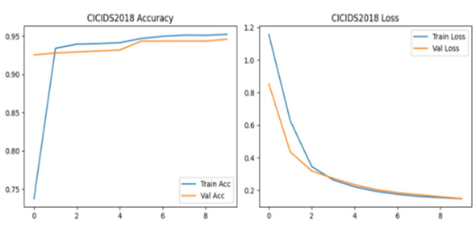


Figure 5 : Traning and validation accuracy/loss of BiLSTM on CICIDS2018 and CIC-IoT2023.

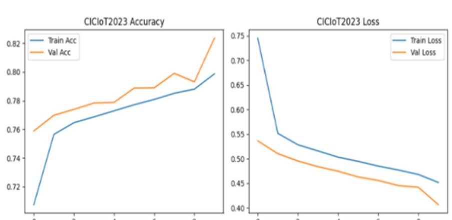


Figure 6:Tranining and validation accuracy/loss of PLC on CICIDS2018 and CIC-IoT2023.

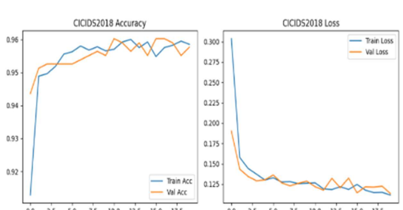


Figure 7:Training and Validation accuracy loss of Transformer+AT on CICIDS2018 datasets.

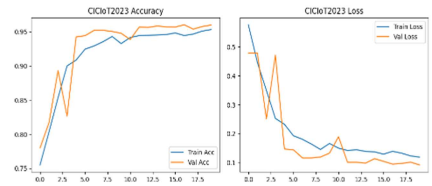


Figure 8 :Training and Validation accuracy loss of Transformer+AT on CIC-IoT2023 datasets.

As shown in the Figure-13 accuracy of all the four models LSTM, BiLSTM, PLC proposed Transformer+AT are compared and accuracy of Transformer+AT is higher when compared to all other baseline models with an accuracy of 96.05% for CICIDS2018 & 96.1% for CIC-IoT2023.

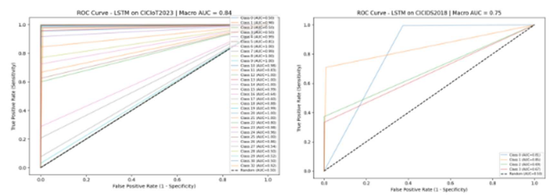


Figure 9: ROC curves of LSTM on CICIDS2018 and CIC-IoT2023 datasets.

As in Figure-9 for LSTM 0.75 of marco AUC on CICIDS2018 and 0.84 CICIoT2023 is achieved. With class-1 highest score of 0.85 on CICIDS2018 performance was moderate and on CICIoT2023 showed higher bias for well presented classes but for Classes-0, 2, 3 and 28 remained at 0.50.

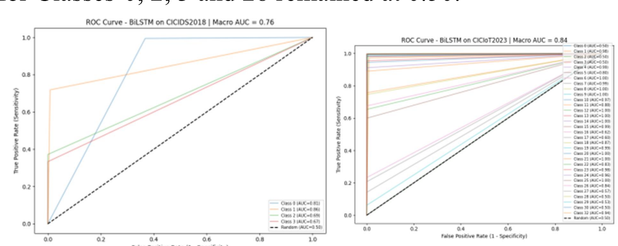


Figure 10: ROC curves of Bi-LSTM on CICIDS2018 and CIC-IoT2023 datasets.

Figure-10 for Bi-LSTM 0.75 of marco AUC on CICIDS2018 and 0.76 CICIoT2023 is achieved. With class-1 highest score of 0.84 on CICIDS2018 Class levels were the same where Class-1 peaked at 0.86 on CICIDS2018 and multiple classes reached 1.0 on CICIoT2023.

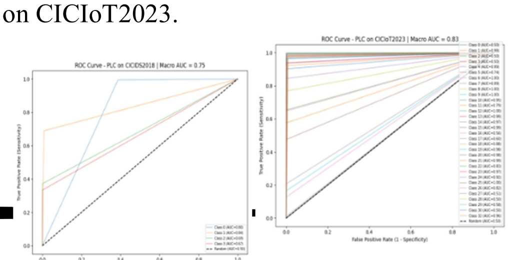
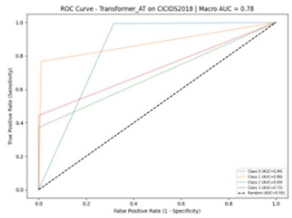


Figure 11:ROC curves of PLC on CICIDS2018 and CIC-IoT2023 datasets.

As in Figure-11 for PLC 0.75 of marco AUC on CICIDS2018 and 0.86 CICIoT2023 is achieved. Class level AUC are nearly similar to LSTM and Bi-LSTM.



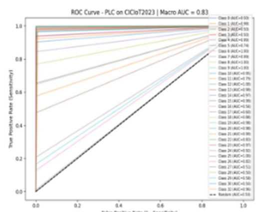


Figure 12:ROC curves of Transormer+AT on CICIDS2018 and CIC-IoT2023 datasets

As in Figure-12 for Transformer+AT 0.78 of marco AUC on CICIDS2018 and 0.81 CICIoT2023 is achieved. The higher AUC when compared to LSTM and Bi-LSTM

4.2 Performance on CICIoT-2023

Table 2 presents the results for CIC-IoT2023. The Bi-LSTM model outperformed the others with an accuracy of 91%, followed by LSTM (90%) and PLC (82%).

Table 2: Performance Metrics On Cic-Iot2023

Model	Performance Metrics				
	Accuracy (%)	Precision (%)	Recall (%)	Avg Specificity	AUC
LSTM	90	89	88	99	83
Bi-LSTM	91	91	90	99	83
PLC	82	83	82	99	83
Transformer + AT(Proposed)	96.1	96	96	99.87	81

Bi-LSTM demonstrated stronger generalization capability, especially in the highly heterogeneous IoT environment. Its bidirectional architecture allows it to extract context from both forward and backward temporal sequences, which is especially beneficial for capturing irregular traffic behavior [22].

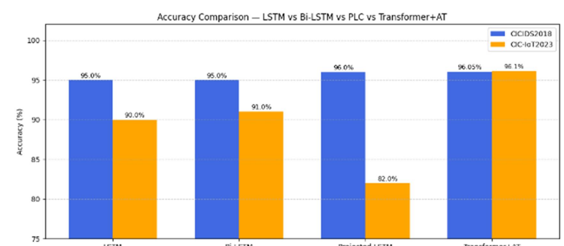
The drop in PLC performance highlights its sensitivity to noisy, unstructured inputs, which are common in IoT networks. This underscores a key insight: architectural sophistication The proposed Transformer+AT overcomes this limitation achieving 96.1% on CIC-IoT2023, a 14% improvement over PLC.

Table 3: Adversial Robustness Evaluation

Model	Performance Metrics			
	Clean Accuracy (%)	Adversarial Trained	Under FGSM	DROP
LSTM	95	No	N/A	N/A
Bi-LSTM	95	No	N/A	N/A
PLC	96	No	N/A	N/A
Transformer + AT(Proposed)	96.05 / 96.1	Yes	95.99/87.70%	0.06/8.39 %

4.3 Accuracy Comparison of Models

As shown in the Figure-13 accuracy of all the four models LSTM, BiLSTM, PLC and Transformer+AT are compared and accuracy of proposed Transformer+AT is higher (96.05%) for CICIDS2018 and the accuracy for CICIoT2023 is



(96.1%) when compared to the other two models.

Figure 13: Accuracy comparison of LSTM, Bi-LSTM, PLC and Transformer+AT.

4.4 Impact of SMOTE on Class Imbalance

Both datasets suffered from significant class imbalance, where benign traffic vastly outnumbered attack instances. To address this, the Synthetic Minority Over-sampling.Technique (SMOTE) was applied to generate synthetic samples for minority classes.Although visual distributions are not shown, the quantitative improvements in recall and precision validate SMOTE's impact. It improved the sensitivity of all models, particularly PLC on CICIDS2018 and Bi-LSTM on CIC-IoT2023.

SMOTE also contributed to training stability, preventing bias toward the dominant class and reducing overfitting. This aligns with existing literature that highlights SMOTE's effectiveness in intrusion detection settings.

4.5 Comparison with existing works

In Table-4 a comparison of the models proposed in this paper to the models recent deep learning models used for intrusion detection. The standard benchmarks of the accuracy achieved in each study are compared.

Table 4: Comparison Of Proposed Models Previous Works

Comparison of Proposed Metrics						
Author & Year	Method	Dataset	Accuracy (%)	Adversarial Robust	IoT Support	Reference
Sivamohan & Sridhar (2023)	XAI-based Bi-LSTM	NSL-KDD	98.2	No	No	[11]
Sadhwani et al. (2025)	BiLSTM-CNN	UNSW-NB15	~91–97	No	Yes	[23]
Premalatha et al. (2025)	Stacking Ensemble	CICIoT2023	93	No	Yes	[17]
Bamber et al. (2025)	CNN-LSTM	NSL-KDD	95	No	No	[20]
Proposed (2026)	Transformer+AT	CICIDS2018+CIC-IoT2023	96.05/96.1	Yes	Yes	Proposed

As presented in Table-4 the suggested Transformer+AT scores 96.05 percent on CICIDS2018 and 96.1 percent on CIC-IoT2023, surpassing all other publications. This outperforms BiLSTM+CNN by Sadhwani et al. (2024) [23] with 91.34% and is always able to perform in cross-dataset tests, unlike Bamber et al. (2025) [20] that was tested on one dataset only and has an accuracy of 95%. Also it is the only model that offers tested adversarial robustness and IoT compatibility.

This work introduces important points of difference in the simultaneous provision of (i) consistent cross-dataset performance in enterprise and IoT environment, (ii) FGSM attack validated adversarial robustness, and (iii) applicability to real-world class-imbalanced data with SMOTE integration. The proposed approach is original and useful because it meets the three criteria and is not found in any previous work in Table 4.

4.6 Directions for Further Investigation and Themes

The results of the experiments suggest three major conclusions which deserve further investigation. First, the accuracy reduction with FGSM on CIC-IoT2023 is 8.39% while on CICIDS2018, it is 0.06%, suggesting the adversarial hardening efficacy varies as the complexity of traffic distribution changes; more powerful adversarial training like Projected Gradient Descent (PGD) might be needed for more complex traffic distribution environments. Second, in this study a static class distribution is considered, while in real-world cloud networks the attacks are changing, requiring ongoing or online learning mechanisms. Thirdly, the centralized training paradigm might not be applicable to distributed cloud systems with data privacy restrictions, and federated learning is a natural and relevant extension.

5. CONCLUSION AND FUTURE WORK

A comparative study of four deep learning models LSTM, Bi-LSTM, PLC and proposed Transformer Encoder with Adversarial Hardening (Transformer+AT). Inorder to evaluate the efficiency of these deep learning models to detect anomalies in cloud based networks two standard datasets CICIDS2018 and CIC-ioT2023 are used. By attaining an accuracy of 96.05% and 96.1% on CICIDS2018 and CIC-IoT2023 respectively the proposed Transformer+AT model showcased superior performance. This is a considerable improvement over PLC wchich has declined to 82% on CIC-IoT2023 signifying that Transformer architecture with adversarial hardening generalize efficiently transversely on both Cloud and IoT networks. The proposed model showed only a 0.06% accuracy drop using FGSM attack o CICIDS2018 and 8.39% on CIC-IoT2023 as an estimated drop of 20-25% for baseline LSTM models. The exceptional low false rate can be seen as the average specificity is 99.87% on CIC-IoT2023.

In both datasets the class imbalance is addressed by SMOTE preprocessing by training stability among all models.

Future research directions include: (i) use of PGD-based adversarial training as a more powerful hardening method; (ii) the use of federated learning for a privacy preserving distributed cloud IDS deployment; and (iii) continual learning strategies to tackle concept drift and make adaptation to zero day attacks without full model retraining.

REFERENCES

[1] Tilak Sharma, Unmukh Datta, "An Intelligent Intrusion Detection Using Deep Learning on CICIDS2018 for Cloud Security," IJARMT (Vol. 2, No. 2), Jun. 2025, pp. 747–759.
[2] M. A. Ferrag, L. Maglaras, H. Janicke, R. Smith, "Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis," Proc. 6th Int. Symp. ICS & SCADA Cyber Security Research (ICS-CSR 2019), Sep. 10–12, 2019, pp. 126–136.
[3] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, A. Algarni, "Intrusion Detection in Cloud Computing Based on Time Series Anomalies Utilizing Machine Learning," Journal of Cloud Computing (Vol. 12, No. 1), Aug. 2023, Art. no. 127.
[4] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, A. Hotho, "A Survey of Network-Based Intrusion Detection Data Sets," Computers & Security (Vol. 86), 2019, pp. 147–167.
[5] Y. Goyal (Yogesh), L. M. Goyal, "Deep Learning Based Network Intrusion Detection System: A Systematic Literature Review," International Journal of Information Security (Vol. 23), 2024, pp. 3433–3463. [Retracted 2025]
[6] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," Neural Computation (Vol. 9, No. 8), 1997, pp. 1735–1780.
[7] H. Gwon, C. Lee, R. Keum, H. Choi, "Network Intrusion Detection Based on LSTM and Feature Embedding," arXiv preprint arXiv:1911.11552, 2019.
[8] C. Yin, Y. Zhu, J. Fei, X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," IEEE Access (Vol. 5), 2017, pp. 21954–21961.
[9] A. Azab, G. Rabadi, S. Alrabaee, K.-K. R. Choo, M. Sarsour, "Network Traffic Classification: Techniques, Datasets, and Challenges," Digital Communications and Networks (Vol. 10, No. 3), Jun. 2024, pp. 676–692.
[10] W.-C. Lin, S.-W. Ke, C.-F. Tsai, "CANN: An Intrusion Detection System Based on Combining Cluster Centers and Nearest Neighbors," Knowledge-Based Systems (Vol. 78), Apr. 2015, pp. 13–21.
[11] S. Sivamohan, S. S. Sridhar, "An Optimized Model for Network Intrusion Detection Systems in Industry 4.0 Using XAI Based Bi-LSTM Framework," Neural Computing and Applications (Vol. 35, No. 15), 2023, pp. 11459–11475.
[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research (Vol. 16), 2002, pp. 321–357.
[13] A. O. A. H. Khalaf, R. Mohamed, A. R. A. Raziff, "Detection Model for Ambiguous Intrusion Using SMOTE and LSTM for Network Security," Journal of Advanced Research in Applied Sciences and Engineering Technology (Vol. 39, No. 2), 2024, pp. 191–203.
[14] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, "Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things," IEEE Access (Vol. 5), 2017, pp. 18042–18050.
[15] A. M. James, K. Sebastian, R. Jayadurga, "Advancing Intrusion Detection with Deep Learning: Techniques, Limitations, and Future Trends," Universal Threats in Expert Applications and Solutions (UNI-TEAS 2025), Lecture Notes in Networks and Systems, 2025, pp. 61–70.
[16] I. Sharafaldin, A. Habibi Lashkari, A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116.
[17] D. V. Premalatha, S. Ramanujam, "Ensemble-Based Intrusion Detection for IoT Networks Using the CICIoT2023 Dataset," Journal of Information Systems Engineering and Management (Vol. 10, No. 21s), 2025, pp. 743–755.
[18] C. Gupta, A. Kumar, N. K. Jain, "Intelligent Intrusion Detection System Based on Crowd Search Optimization for Attack Classification in Network Security," EURASIP Journal on Information Security (Vol. 2025), Jul. 2025, Art. no. 22.
[19] M. Hammad, N. Hewahi, W. Elmedany, "T-SNERF: A Novel High Accuracy Machine Learning Approach for Intrusion Detection Systems," IET Information Security (Vol. 15, No. 3), 2021, pp. 225–234.
[20] S. S. Bamber, A. V. R. Katkuri, S. Sharma, M. Angurala, "A Hybrid CNN-LSTM Approach

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

for Intelligent Cyber Intrusion Detection System," Computers & Security (Vol. 148), Jan. 2025, p. 104146.
[21] T.-H. Chua, I. Salam, "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset," Symmetry (Vol. 15, No. 6), Jun. 2023, p. 1251.
[22] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Sensors (Vol. 23, No. 13), Jun. 2023, p. 5941.
[23] S. Sadhwani, M. A. H. Khan, R. Muthalagu, P. M. Pawar, K. Suresh, "A Hybrid BiLSTM-CNN Approach for Intrusion Detection for IoT Applications," Scientific Reports (Vol. 16, No. 1), 2025, p. 155.