International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(10s) 2026 pp. 547-567
https://doi.org/10.70917/ijcisim-2026-3637
Article

LF-TSE-IDS: An Imbalance-Aware Leakage-Free Temporal Stacking Ensemble for Multiclass Network Intrusion Detection

Anadi Shivam Joshi[1], Gaurav Shrivastava[2], Ritu Patidar[3] , Upendra Singh[4]

[1,2]Department of Information Security and Cloud Computing, Shri Vaishnav Institute of Information Technology, Indore, Madhya Pradesh, India
[3]Faculty of Engineering Sciences and Technology (FEST), Adani University, Ahmedabad, Gujarat, India
[4]Department of Information Technology, Shri G. S. Institute of Technology and Science, Indore, Madhya Pradesh, India
joshianadi@svvv.edu.in, gaurav2086@gmail.com , ritupatidar89@gmail.com , upendrasingh49@gmail.com

Abstract: Network intrusion detection remains challenging because of highly imbalanced traffic classes, evolving attack behaviors, temporal dependencies, and data leakage during model development. This study proposes LF-TSE-IDS, an imbalance-aware leakage-free temporal stacking ensemble for multiclass network intrusion detection. The framework preprocesses CICIDS2017 traffic performs five-tuple-based grouping and timestamp ordering, and constructs overlapping temporal windows with a sequence length of 20 and stride of 3. Class imbalance is addressed using class-weighted focal loss for deep models and training-only SMOTE for static learners. BiGRU-Attention and Dilated 1D-CNN capture sequential traffic characteristics, whereas Balanced Random Forest and XGBoost model static nonlinear relationships. Their class-probability outputs are integrated through five-fold out-of-fold stacking using a Logistic Regression meta-learner. Experimental results show that LF-TSE-IDS achieves 97.20% test accuracy, 97.18% weighted-F1, 0.9392 MCC, 0.9952 micro-ROC-AUC, and 0.9356 macro-PR-AUC. The framework therefore provides reliable generalization, reduced stacking bias, and robust multiclass intrusion detection.
Keywords: Intrusion detection system, temporal learning, stacking ensemble, data leakage, class imbalance, BiGRU-Attention, Dilated 1D-CNN, XGBoost, CICIDS2017.

1. Introduction

The rapid expansion of cloud computing, Internet of Things environments, software-defined networks, connected vehicles, cyber-physical systems, and intelligent communication infrastructures has significantly increased the scale and complexity of modern network traffic. Although these technologies improve connectivity and service availability, they also expose digital systems to heterogeneous cyberattacks, including denial-of-service, distributed denial-of-service, brute-force, botnet, infiltration, port-scanning, web-based attacks, and previously unseen malicious activities. Conventional security mechanisms based on fixed signatures and manually defined rules remain effective against known threats but often fail to recognize evolving, obfuscated, or zero-day attacks. Consequently, intelligent intrusion-detection systems have become an essential component of modern cybersecurity architecture.

Machine-learning-based intrusion-detection approaches provide automated classification by learning statistical relationships from network-flow features. Methods based on Random Forest, Decision Forest, Naïve Bayes, and related classifiers offer comparatively low computational cost and effective nonlinear decision boundaries [2], [8], [18]. However, most traditional models process network flows as independent records and therefore overlook the temporal dependencies, behavioural transitions, and sequential attack patterns that develop across related communications. Dimensionality-reduction methods such as PCA can reduce redundant information, but they may discard nonlinear and temporally discriminative features required to distinguish closely related attack classes [18].

Deep-learning methods have been introduced to overcome the limited representation capability of conventional classifiers. Recurrent architectures such as LSTM and GRU can model sequential traffic behaviour, while convolutional networks can identify local and multi-scale patterns within network-flow sequences [11], [13], [14]. Attention mechanisms further improve sequence modelling by assigning greater importance to informative traffic

observations. Nevertheless, deep models commonly face class imbalance, high computational complexity, overfitting, sensitivity to hyperparameter selection, and reduced performance for rare attack classes. Lightweight and adaptive intrusion-detection methods have also been proposed for resource-constrained, IoT, wireless, consumer-electronic, and Open RAN environments [4], [7], [12], [15]. Despite their practical value, these approaches generally optimize either detection effectiveness, adaptability, or computational efficiency rather than addressing all three requirements simultaneously.

Class imbalance represents a particularly important challenge in multiclass intrusion detection because benign and common attack categories frequently dominate the dataset, while critical attacks may contain only a small number of observations. A model can therefore achieve high overall accuracy while performing poorly on minority classes. Class weighting, oversampling, focal loss, and balanced tree learning have been employed to reduce this bias. However, oversampling performed before dataset partitioning can unintentionally transfer information from validation or test observations into the training process. Similarly, temporal windows generated from overlapping flows may be distributed across different subsets, creating nearly identical training and testing samples. Such data leakage produces optimistic performance estimates and weakens the credibility of experimental findings.

Ensemble learning offers a promising solution because models with different learning mechanisms can capture complementary aspects of network traffic. Tree-based learners are effective for static nonlinear feature relationships, whereas recurrent and convolutional networks are more suitable for temporal representation learning. However, simple voting and direct probability averaging cannot optimally learn the reliability of individual models across different attack classes. Moreover, training a meta-learner on predictions generated from the same samples used to fit the base models introduces stacking leakage and overconfident meta-features. Out-of-fold probability generation is therefore necessary to ensure that every meta-training prediction is produced by a model that has not observed the corresponding sample during training.

To address these limitations, this study proposes LF-TSE-IDS, an imbalance-aware leakage-free temporal stacking ensemble for multiclass network intrusion detection. The framework first performs duplicate removal, missing-value treatment, categorical encoding, and feature scaling. Network records are then grouped using source IP, destination IP, source port, destination port, and protocol, followed by timestamp-based ordering. Temporal windows of length 20 with a stride of 3 are constructed to capture overlapping behavioural patterns. Dataset partitioning is completed before balancing, and imbalance handling is restricted to the training data through class-weighted focal loss for deep models and SMOTE for static models. The framework combines BiGRU-Attention, Dilated 1D-CNN, Balanced Random Forest, and XGBoost as heterogeneous base learners. Their probability outputs are transformed into leakage-free meta-features through five-fold out-of-fold prediction and integrated using a Logistic Regression meta-learner.

The principal contributions of this study are as follows:

1. A leakage-free preprocessing and temporal-window construction strategy based on five-tuple grouping and chronological ordering is developed to reduce flow overlap and future-information leakage.
2. An imbalance-aware hybrid learning mechanism is introduced by applying class-weighted focal loss to sequential deep models and training-only SMOTE to static classifiers.
3. Complementary temporal and static representations are jointly learned through BiGRU-Attention, Dilated 1D-CNN, Balanced Random Forest, and XGBoost.
4. A five-fold out-of-fold probability-stacking method is employed to prevent meta-learning bias and improve generalization.
5. The framework is comprehensively evaluated using accuracy, balanced accuracy, macro precision, macro recall, macro-F1, weighted-F1, MCC, ROC-AUC, PR-AUC, log loss, latency, and training time.

The experimental results on the CICIDS2017 dataset demonstrate that LF-TSE-IDS achieves a test accuracy of **97.20%**, weighted-F1 of **97.18%**, MCC of **0.9392**, micro ROC-AUC of **0.9952**, and macro PR-AUC of **0.9356**. These findings indicate that the proposed framework provides reliable overall classification, strong prediction consistency, and effective discrimination across multiple traffic classes. Although individual models remain superior for certain class-balanced metrics, the stacking framework offers the most consistent balance between predictive accuracy, robustness, and computational performance.

he remainder of this paper is organized as follows. **Section 2** reviews existing machine-learning, deep-learning, optimization, lightweight, adaptive, decentralized, and continual-learning intrusion-detection approaches and identifies the principal research gaps. **Section 3** presents the proposed LF-TSE-IDS methodology, system architecture, temporal data preparation, base learners, out-of-fold stacking algorithm, and hyperparameter configuration. **Section 4** describes the experimental environment, CICIDS2017 dataset, implementation details, evaluation measures, comparative results, confusion matrices, ROC analysis, and performance discussion. Finally, **Section 5** concludes the study and outlines future research directions, including cross-dataset validation, explainable detection, federated learning, concept-drift adaptation, and real-time edge deployment.

2. Literature Review

K. J. et al. (2026) addressed the limited optimization capability of conventional intrusion-detection models by proposing a hybrid genetic algorithm–grey wolf optimization framework. The method combines evolutionary search with intelligent feature and parameter optimization to improve attack classification. The study reported enhanced detection effectiveness; however, broader dataset validation and computational-complexity analysis remain necessary. Future work should examine real-time deployment and adaptive optimization [1].

Mahamud et al. (2025) investigated automated anomaly-based intrusion detection for IoT environments, where heterogeneous devices and dynamic traffic make rule-based security ineffective. The authors employed machine-learning classifiers to distinguish benign and malicious network behaviour. Their approach demonstrated the potential of automated anomaly identification, but its performance under unseen attacks and highly imbalanced traffic was insufficiently explored. Future research should incorporate continual and federated learning [2].

Ather et al. (2025) focused on the inability of static intrusion-detection systems to respond effectively to rapidly changing cyber threats. They proposed an adaptive machine-learning-based IDS for real-time threat identification and mitigation. The framework improved responsiveness to evolving traffic patterns; nevertheless, limited evaluation of concept drift, adversarial robustness, and deployment latency restricts generalization. Future work should develop self-updating models for large-scale networks [3].

S. K. et al. (2026) addressed real-time attack detection in resource-constrained IoT networks using machine-learning techniques. The proposed system analysed network traffic and classified suspicious activities with low operational overhead. The results indicated improved feasibility for IoT security; however, the study requires validation across diverse devices, protocols, and zero-day attacks. Future work should emphasize lightweight online learning and edge-based implementation [4].

Venugopal et al. (2025) examined the centralization and single-point-of-failure limitations of conventional IoT intrusion-detection systems. They implemented a decentralized IDS that distributes monitoring and decision-making across network nodes. The architecture improved resilience and scalability, but synchronization overhead, communication cost, and trust management were not fully resolved. Future studies should integrate blockchain, federated learning, and privacy-preserving collaborative detection [5].

Mihaylova (2025) analysed adversarial machine-learning attacks against network intrusion-detection systems, highlighting how carefully manipulated inputs can evade trained classifiers. The work classified major adversarial attack types and examined their effect on IDS reliability. Although it strengthened understanding of model vulnerabilities, experimental defence mechanisms were limited. Future research should evaluate adversarial training, certified robustness, and explainable detection under realistic attack conditions [6].

S. M. et al. (2026) addressed the high computational cost of wireless intrusion detection in resource-constrained networks by proposing a lightweight frame-based IDS. The system analyses wireless-frame characteristics rather than relying on complex packet-level processing. It demonstrated efficient detection with reduced resource demand; however, cross-protocol applicability and resilience to sophisticated attacks require further investigation. Future work should target embedded and energy-aware deployment [7].

Durole et al. (2025) explored the use of Naïve Bayes and Decision Forest techniques to improve intrusion detection while maintaining manageable computational complexity. Their comparative machine-learning approach identified malicious traffic using probabilistic and ensemble decision mechanisms. The reported findings showed useful classification capability, but limited model diversity and inadequate evaluation of minority attacks remain concerns. Future work should employ hybrid ensembles and imbalance-aware learning [8].

Tran et al. (2025) addressed catastrophic forgetting in graph-neural-network-based intrusion detection by introducing a memory-replay mechanism. The method preserves representative historical traffic samples while adapting the GNN to new network behaviours. The approach improved continual detection capability; however, memory selection, storage overhead, and long-term scalability remain unresolved. Future work should investigate adaptive replay, graph compression, and privacy-preserving continual learning [9].

Zhang et al. (2025) reviewed automotive intrusion-detection and prevention technologies for protecting connected and autonomous vehicles from cyberattacks. The study examined in-vehicle networks, attack surfaces, monitoring mechanisms, and prevention strategies. It highlighted the importance of integrated security architectures, but lacked extensive experimental comparison across vehicle platforms. Future research should address vehicle-to-everything security, real-time adaptation, and standardized automotive datasets [10].

Zeeshan et al. (2025) addressed cyber-physical security by combining a spider-monkey-optimized GRU–LSTM model with blockchain-based protection. The optimization method tunes the recurrent architecture, while blockchain supports trusted and tamper-resistant security records. The framework improved intelligent attack detection; however, blockchain latency, energy consumption, and scalability may hinder deployment. Future work should examine lightweight consensus and edge-assisted implementation [11].

Gaurav et al. (2026) investigated zero-day threat detection in consumer electronic devices, where limited processing capacity restricts complex security models. The authors proposed a lightweight AI-enhanced intrusion-detection system capable of identifying previously unseen malicious behaviour. The study demonstrated promising detection capability, but generalization across device types and evolving attacks requires further validation. Future work should explore continual, explainable, and on-device learning [12].

Kadri et al. (2025) examined real-time threat classification by integrating deep-learning and machine-learning algorithms within an intrusion-detection framework. The study compared complementary models for extracting complex traffic patterns and classifying attacks. The results indicated improved threat-recognition capability; however, computational cost, dataset imbalance, and deployment latency remained important limitations. Future research should develop optimized hybrid ensembles with real-time interpretability [13].

Zhang (2025) proposed a deep-learning-based network intrusion-detection and defence system to overcome the limited feature-learning capability of conventional security approaches. The method automatically extracted discriminative representations from network traffic and supported attack classification. Although effective detection was reported, model complexity and limited explainability may restrict practical adoption. Future work should include lightweight architectures, adversarial robustness, and adaptive defence mechanisms [14].

Niknami et al. (2026) addressed changing attack patterns in Open Radio Access Networks through RAN-IDS, an adaptive continual-learning intrusion-detection framework. The method incrementally updates the detection model as new traffic and threats emerge. It improved adaptability while reducing model obsolescence; however, catastrophic forgetting, resource overhead, and distributed RAN deployment remain challenges. Future work should investigate federated continual learning and drift-aware adaptation [15].

James et al. (2025) studied the role of artificial intelligence in real-time intrusion and anomaly detection for IoT-enabled smart networks. The work discussed how AI models can identify complex and evolving malicious behaviour more effectively than static rules. However, privacy concerns, computational limitations, and insufficient explainability restrict adoption. Future research should focus on trustworthy, lightweight, and privacy-preserving AI security frameworks [16].

Zhang et al. (2025) investigated laser-based intrusion detection using three-dimensional point-cloud discretization and data-processing techniques. The method converts spatial sensing data into structured representations for identifying unauthorized physical entry. The approach improved environmental perception, but sensitivity to weather, occlusion, sensor noise, and computational burden remains problematic. Future work should combine multimodal sensing, deep learning, and real-time edge processing [17].

Waskle et al. (2020) addressed high-dimensional network data by combining principal component analysis with a Random Forest classifier. PCA reduced redundant features, while Random Forest performed intrusion classification using the transformed data. The method achieved effective detection with lower feature complexity; however, linear dimensionality reduction may discard nonlinear relationships. Future work should examine advanced feature selection and hybrid temporal models [18].

K. J. et al. (2026) addressed the limited exploration capability of conventional optimization-based intrusion-detection systems by integrating Genetic Algorithm and Grey Wolf Optimization. The hybrid method improves feature selection and classifier-parameter tuning by balancing global and local search. Although detection performance improved, computational overhead and convergence stability remain concerns. Future work should investigate adaptive parameter control, parallel execution, and validation on large, imbalanced network datasets [19].

Mahamud et al. (2025) addressed the difficulty of identifying anomalous activities in heterogeneous IoT environments using static signature-based mechanisms. The authors applied machine-learning classifiers to automate traffic analysis and malicious-behaviour recognition. Their findings demonstrated improved anomaly-detection capability and reduced dependence on predefined signatures. However, limited zero-day evaluation and cross-dataset testing restrict generalizability. Future research should integrate temporal learning, explainable AI, and federated model training [20].

Ather et al. (2025) examined the inability of conventional intrusion-detection systems to adapt to rapidly changing attack behaviours. Their adaptive machine-learning framework continuously analyses network traffic for real-time threat identification and mitigation. The approach improved detection responsiveness, but insufficient consideration of concept drift, encrypted traffic, and adversarial manipulation limits operational reliability. Future work should develop drift-aware, self-learning models with automated retraining and low-latency edge deployment [21].

S. K. et al. (2026) focused on real-time intrusion detection in IoT networks with limited computational and memory resources. The authors employed machine-learning methods to classify network traffic while maintaining acceptable processing efficiency. The system demonstrated practical potential for connected-device security; however, evaluation across diverse IoT protocols and attack families was limited. Future work should investigate lightweight ensembles, online feature extraction, and hardware-assisted implementation on edge devices [22].

Venugopal et al. (2025) addressed the single-point-of-failure problem associated with centralized IoT intrusion-detection architectures. Their decentralized framework distributes traffic monitoring and detection responsibilities among multiple participating nodes. This design improves availability and resilience; however, communication overhead, node trust, synchronization, and privacy remain unresolved. Future research should combine decentralized detection with blockchain-based integrity, federated learning, secure aggregation, and reputation-aware collaborative decision mechanisms [23].

Mihaylova (2025) investigated how adversarially manipulated network samples can mislead machine-learning-based intrusion-detection systems. The study classified major evasion and poisoning attacks and analysed their effects on classification reliability. It highlighted critical weaknesses in existing models, but did not provide comprehensive experimental defences. Future work should integrate adversarial training, input purification, uncertainty estimation, robust feature selection, and certified defence mechanisms under realistic network conditions [24].

S. M. et al. (2026) addressed excessive computation in wireless intrusion detection by developing a lightweight frame-based security mechanism for resource-constrained networks. The method examines wireless-frame characteristics and avoids expensive payload analysis. It achieved efficient detection with reduced processing requirements; nevertheless, its resistance to encrypted, coordinated, and cross-layer attacks remains unclear. Future work should evaluate energy consumption, embedded implementation, protocol diversity, and adaptive online classification [25].

Durole et al. (2025) examined intrusion detection using Naïve Bayes and Decision Forest classifiers to balance predictive performance and computational simplicity. Naïve Bayes provides efficient probabilistic classification, whereas Decision Forest captures nonlinear feature relationships. Their comparative implementation improved malicious-traffic identification; however, limited handling of temporal dependencies and class imbalance restricts performance. Future studies should employ stacking, boosting, temporal feature learning, and cost-sensitive classification for rare attacks [26].

Tran et al. (2025) addressed catastrophic forgetting in graph-neural-network-based intrusion detection through a memory-replay strategy. Historical graph samples are retained and reused during incremental updates, enabling the model to preserve previously learned attack knowledge. The approach improved continual-learning performance, but storage requirements and replay-sample selection remain limitations. Future work should investigate compressed memory, uncertainty-guided replay, privacy protection, and adaptive graph learning under evolving network topology [27].

Zhang et al. (2025) investigated intrusion-detection and prevention technologies for connected and autonomous vehicles. The study reviewed in-vehicle networks, attack surfaces, detection architectures, and defensive applications. It emphasized the need for timely and reliable automotive cybersecurity; however, limited benchmark standardization and real-world experimentation restrict comparative assessment. Future work should develop vehicle-specific datasets, cross-domain detection, over-the-air adaptation, and lightweight protection for vehicle-to-everything communications [28].

Zeeshan et al. (2025) addressed cyber-physical intrusion detection by integrating Spider Monkey Optimization, GRU–LSTM learning, and blockchain security. Optimization improves recurrent-model configuration, while blockchain maintains trustworthy and tamper-resistant security records. The framework enhanced attack identification and data integrity; however, high computational cost, blockchain delay, and scalability remain limitations. Future work should explore lightweight distributed ledgers, edge-assisted processing, and energy-efficient recurrent architectures [29].

Gaurav et al. (2026) addressed zero-day attack detection in consumer electronics, where limited hardware capacity makes conventional deep models unsuitable. The authors developed a lightweight AI-enhanced IDS for recognizing previously unseen malicious behaviour. The proposed solution demonstrated promising detection with reduced resource demand; however, transferability across devices and attack domains requires further validation. Future research should include continual learning, explainable predictions, model compression, and secure on-device adaptation [30].

Kadri et al. (2025) investigated real-time threat classification by applying deep-learning and machine-learning algorithms to network-traffic analysis. The study demonstrated that automated feature learning can improve recognition of complex attack behaviours. Nevertheless, model comparison was limited by computational cost, class imbalance, and insufficient external validation. Future work should integrate complementary learners through leakage-free stacking, optimize inference latency, and evaluate models across multiple contemporary intrusion datasets [31].

Zhang (2025) addressed the restricted feature-learning capability of conventional intrusion-detection systems through a deep-learning-based detection and defence framework. The model automatically extracted complex network representations and classified suspicious traffic. Although improved detection was reported, limited interpretability and resource-intensive training may hinder real-world adoption. Future work should incorporate explainable attention mechanisms, lightweight architectures, adversarial robustness, online updating, and automated response strategies [32].

Niknami et al. (2026) addressed evolving cyber threats in Open Radio Access Networks through an adaptive continual-learning framework called RAN-IDS. The model incrementally learns new traffic patterns without complete retraining, improving adaptability to changing attacks. However, catastrophic forgetting, distributed-resource constraints, and communication overhead remain challenging. Future work should combine federated continual learning, drift detection, replay optimization, and privacy-preserving collaboration across multiple RAN components [33].

James et al. (2025) examined artificial intelligence for real-time intrusion and anomaly detection in smart IoT networks. The study highlighted AI's ability to identify complex behaviours that traditional rule-based systems may overlook. However, privacy risks, computational limitations, false alarms, and low model transparency restrict practical deployment. Future research should develop trustworthy, lightweight, explainable, and privacy-preserving AI architectures suitable for dynamic edge and cloud environments [34].

Zhang et al. (2025) addressed physical intrusion detection using three-dimensional point-cloud discretization and laser-sensing data processing. Their approach transforms spatial observations into structured representations for identifying unauthorized entry. The method improved environmental monitoring, but sensor noise, weather changes, occlusion, and processing cost remain limitations. Future work should incorporate deep point-cloud networks, multimodal sensor fusion, adaptive filtering, and real-time edge processing [35].

Waskle et al. (2020) addressed high-dimensional intrusion data using Principal Component Analysis and Random Forest classification. PCA reduces redundant and correlated features, whereas Random Forest performs robust nonlinear classification. The approach improved detection efficiency and reduced dimensionality; however, PCA may remove class-discriminative nonlinear information. Future research should investigate hybrid feature selection, temporal representation learning, imbalance-aware ensembles, and cross-dataset generalization [36].

3. Methdology

The proposed LF-TSE-IDS methodology follows a leakage-free temporal and ensemble-learning pipeline for intrusion detection using the CICIDS2017 dataset. The raw traffic records are cleaned, encoded, scaled, grouped according to five-tuple flow identifiers, and arranged in chronological order before constructing overlapping temporal windows of length 20 with a stride of 3. The processed data are divided into 70% training, 15% validation, and 15% testing subsets, with class balancing applied only to the training data through class-weighted focal loss for deep models and SMOTE for static models. BiGRU-Attention and Dilated 1D-CNN learn sequential traffic behaviour, while Balanced Random Forest and XGBoost capture static and nonlinear feature relationships. Their class-probability outputs are combined through five-fold out-of-fold stacking and a Logistic Regression meta-learner to generate the final attack predictions. Performance is assessed using accuracy, precision, recall, macro-F1, weighted-F1, MCC, ROC-AUC, PR-AUC, confusion matrix, and inference latency.

3.1 Proposed architecture

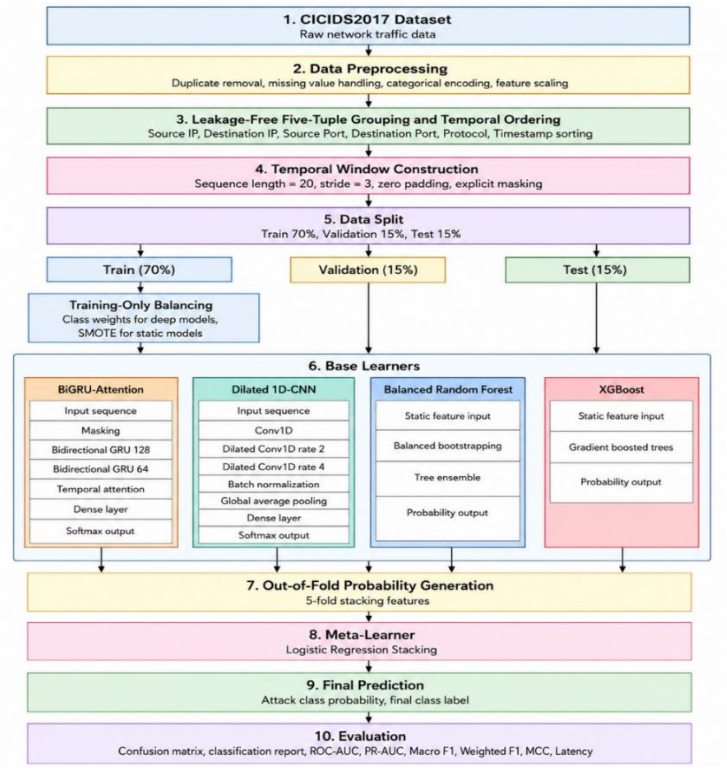

Figure 1. Proposed LF-TSE-IDS Architecture for Leakage-Free Temporal and Ensemble-Based Intrusion Detection.

The proposed in figure 1 LF-TSE-IDS architecture processes the CICIDS2017 dataset through duplicate removal, missing-value treatment, categorical encoding, and feature scaling, followed by leakage-free five-tuple grouping and timestamp-based temporal ordering. Fixed-length temporal windows are then constructed using a sequence length of 20, stride of 3, zero padding, and explicit masking. The dataset is divided into 70% training, 15% validation, and 15% testing subsets, while class balancing is applied only to the training data using class weights for deep-learning models and SMOTE for static machine-learning models. Four complementary base learners BiGRU-Attention, Dilated 1D-CNN, Balanced Random Forest, and XGBoost generate class-probability outputs from sequential and static feature representations. These outputs are transformed into leakage-free out-of-fold stacking features using five-fold cross-validation and are subsequently integrated through a logistic-regression meta-learner. The final stage produces attack-class probabilities and predicted labels, which are evaluated using the confusion matrix, classification report, ROC-AUC, PR-AUC, macro-F1, weighted-F1, Matthews correlation coefficient, and inference latency.

3.2 Proposed algorithms

Algorithm 1: Leakage-Free Temporal Data Preparation and Base-Learner Training

Input: CICIDS2017 dataset $D = \{X, y\}$, temporal window length $L = 20$, stride $S = 3$, number of classes C, and number of folds $K = 5$

Output: Processed datasets $D_{\text{tr}}, D_{\text{val}}, D_{\text{te}}$, trained base learners $\mathcal{M}$, and labels y_{tr}

1. Load the CICIDS2017 dataset D.
2. Remove duplicate records from D.
3. Replace missing and infinite values using the selected imputation method.
4. Encode categorical attributes and scale numerical features.
5. Define the network-flow identifier as

   $f = \{\text{SrcIP}, \text{DstIP}, \text{SrcPort}, \text{DstPort}, \text{Protocol}\}$
6. Group the traffic records according to f.
7. Sort the records in each group using their timestamps.
8. For each temporally ordered traffic group:
   1. Construct a sliding window of length $L$and stride S.
   2. Apply zero padding to incomplete windows.
   3. Generate an explicit masking vector for padded observations.
9. Combine the generated temporal windows into the processed dataset D'.
10. Split D'into training, validation, and testing subsets as

    $D_{\text{tr}} = 70\%,\ D_{\text{val}} = 15\%,\ D_{\text{te}} = 15\%$. Apply balancing only to D_{tr}:
    1. Compute class weights for the deep-learning models.
    2. Apply SMOTE to the static-feature training data.
11. Initialize the base-learner set as

    $\mathcal{M} = \{M_{\text{BiGRU}}, M_{\text{CNN}}, M_{\text{BRF}}, M_{\text{XGB}}\}$.
12. Configure M_{BiGRU}as

    Masking → BiGRU(128) → BiGRU(64) → Attention → Dense → Softmax.
13. Configure M_{CNN}as

    Conv1D → Dilated Conv1D(2) → Dilated Conv1D(4) → BatchNorm → Global Average Pooling → Dense → Softmax.
14. Configure M_{BRF} using balanced bootstrapping and a tree ensemble.
15. Configure M_{XGB} using gradient-boosted decision trees.
16. Train M_{BiGRU} and M_{CNN} using temporal-window inputs.
17. Train $M_{\text{BRF}}$and M_{XGB} using static-feature inputs.
18. Validate the base learners using D_{val}.
19. Return $D_{\text{tr}}, D_{\text{val}}, D_{\text{te}}, y_{\text{tr}}$, and $\mathcal{M}$.

Algorithm 1 performs leakage-free preprocessing, temporal sequence generation, data partitioning, and heterogeneous base-learner construction. Network records are grouped using five-tuple flow information and temporally ordered before sliding-window formation. Data balancing is restricted to the training subset to prevent validation and testing information from influencing model learning. The resulting sequential and static data representations are used to train the BiGRU-Attention, Dilated 1D-CNN, Balanced Random Forest, and XGBoost

classifiers. The processed datasets and base-learner configurations produced by Algorithm 1 are supplied to Algorithm 2 for out-of-fold probability generation and stack prediction.

Algorithm 2: Out-of-Fold Probability Stacking and Final Intrusion Classification

Input: Training dataset D_{tr}, validation dataset D_{val}, test dataset D_{te}, training labels y_{tr}, trained base-learner set $\mathcal{M}$, number of folds $K = 5$, and number of classes C

Output: Final probability matrix P_{te}, predicted labels $\hat{y}_{\mathrm{te}}$, trained meta-learner M_{meta}, and evaluation results R

1. Receive $D_{\mathrm{tr}}, D_{\mathrm{val}}, D_{\mathrm{te}}, y_{\mathrm{tr}}$, and $\mathcal{M}$ from Algorithm 1.
2. Initialize the out-of-fold feature matrix as

   $Z_{\mathrm{tr}} \in \mathbb{R}^{N_{\mathrm{tr}} \times (|\mathcal{M}|C)}$.
3. Initialize the test-probability matrix as

   $Z_{\mathrm{te}} \in \mathbb{R}^{N_{\mathrm{te}} \times (|\mathcal{M}|C)}$.
4. For each base learner $M_i \in \mathcal{M}$:
    1. Partition D_{tr} into K stratified folds.
    2. Initialize the test-probability accumulator $A_i = 0$.
    3. For $k = 1$ to K:
        1. Select fold k as the held-out subset D_k.
        2. Select the remaining folds as D_{-k}.
        3. Apply class weighting or SMOTE only to D_{-k}.
        4. Train a fresh instance $M_i^{(k)}$ using D_{-k}.
        5. Compute held-out class probabilities as

           $P_k = M_i^{(k)}(D_k)$. Store P_k in the corresponding rows of Z_{tr}.
        6. Compute test probabilities as

           $P_{\mathrm{te}}^{(k)} = M_i^{(k)}(D_{\mathrm{te}})$.
        7. Update the accumulator as

           $A_i \leftarrow A_i + P_{\mathrm{te}}^{(k)}$.
    4. End for.
    5. Compute the average test probabilities as

       $\bar{P}_{i,\mathrm{te}} = \frac{A_i}{K}$.
    6. Store $\bar{P}_{i,\mathrm{te}}$ in the corresponding columns of Z_{te}.
5. End for.
6. Initialize the logistic-regression meta-learner M_{meta}.
7. Train M_{meta} using Z_{tr} and y_{tr}.
8. Tune the regularization and decision settings using D_{val}.
9. Compute the final class-probability matrix as

   $P_{\mathrm{te}} = M_{\mathrm{meta}}(Z_{\mathrm{te}})$.
10. Determine the final class labels as

$\hat{y}_{\text{te}} = \arg \max_{c \in \{1,\ldots,C\}} P_{\text{te}}(:, c)$.

11. Compute the confusion matrix and class-wise classification report.
12. Compute ROC-AUC and PR-AUC.
13. Compute macro-F1, weighted-F1, and Matthew's correlation coefficient.
14. Measure the average prediction latency.
15. Store all performance measures in R.
16. Return $P_{\text{te}}, \hat{y}_{\text{te}}, M_{\text{meta}}$, and R.

Algorithm 2 uses the processed data and base-learner definitions generated by Algorithm 1 to construct leakage-free stacking features. Each learner is retrained across five stratified folds, and predictions for a training sample are produced only by a model that has not observed that sample during fitting. The out-of-fold probability vectors are concatenated and used to train a logistic-regression meta-learner. Fold-specific test probabilities are averaged and passed to the meta-learner to obtain the final attack-class probabilities and labels. The complete LF-TSE-IDS framework is then evaluated using discrimination, classification, correlation, and computational-efficiency measures.

3.3 Comparison of Hyperparameter

Table 1. Technical Comparison of Hyperparameter-Tuning Settings

No.	Hyperparameter / Feature	Existing Base Paper	Proposed LF-TSE-IDS
1	**Sequence length**	$T = 20$, with ablation at 5, 10, and 20	$T = 20$, stride $= 3$
2	**Deep-learning epochs**	Maximum 30 epochs	Maximum 100 epochs with early stopping
3	**Batch size**	1024 for LSTM, GRU, and 1D-CNN; 256 for Transformer	256 for BiGRU-Attention and Dilated 1D-CNN
4	**Learning rate**	10^{-3} for LSTM and 1D-CNN; 5×10^{-4} for GRU and Transformer	5×10^{-4} for BiGRU-Attention; 7×10^{-4} for Dilated 1D-CNN
5	**Recurrent architecture**	LSTM 64–32 and GRU 64–32	BiGRU 128–64 with temporal attention
6	**CNN configuration**	Conv1D 64–128 with pooling and global average pooling	Dilated Conv1D 128 with dilation rates 1, 2, and 4 plus residual connection
7	**Random Forest**	100 trees, maximum depth 20	500 balanced trees, unrestricted depth, balanced subsampling
8	**Boosting model**	Not included	XGBoost with 700 trees, depth 10, learning rate 0.035, subsample 0.90
9	**Class-imbalance handling**	Inverse-frequency class weights	Class-weighted focal loss for deep models and training-only SMOTE for static models
10	**Ensemble tuning**	Individual models evaluated separately	Five-fold out-of-fold stacking with Logistic Regression, $C = 1.5$, maximum 4000 iterations

Table 1 presents a technical comparison between the hyperparameter-tuning settings of the existing base paper and the proposed LF-TSE-IDS framework. Although both approaches use a sequence length of $T = 20$, the proposed method additionally employs a stride of 3 to generate overlapping temporal windows and capture fine-

grained traffic dependencies. The proposed deep-learning models are trained for a maximum of 100 epochs with early stopping, compared with only 30 epochs in the base paper, while a uniform batch size of 256 is adopted for the BiGRU-Attention and Dilated 1D-CNN models to improve training stability and memory efficiency. Model-specific learning rates of $5 \times 10^{-4}$and $7 \times 10^{-4}$are used for the BiGRU-Attention and Dilated 1D-CNN models, respectively. The conventional LSTM and GRU architectures are replaced with a deeper BiGRU $128 - 64$network integrated with temporal attention, whereas the standard CNN is enhanced using dilated convolutions with dilation rates of 1, 2, and 4 and a residual connection. The proposed framework also strengthens static classification through a Balanced Random Forest containing 500 trees and an XGBoost model with 700 trees. Class imbalance is handled using class-weighted focal loss for deep models and training-only SMOTE for static learners. Finally, unlike the base paper, which evaluates individual models separately, LF-TSE-IDS applies five-fold out-of-fold probability stacking with a Logistic Regression meta-learner, $C = 1.5$, and a maximum of 4000 iterations to improve robustness, generalization, and final classification performance.

Table 2. Technical Comparison of Hyperparameter-Tuning Settings

No.	Hyperparameter / Feature	Existing Base Paper	Proposed LF-TSE-IDS	Justification for Better Proposed Setting
1	**Sequence length**	$T = 20$, with ablation at 5, 10, and 20	$T = 20$, stride $= 3$	Retains the best-performing temporal horizon while generating more overlapping behavioural patterns through controlled stride.
2	**Training epochs**	Maximum 30 epochs	Maximum 100 epochs with early stopping	Provides sufficient learning time while early stopping prevents unnecessary overfitting.
3	**Batch size**	1024 for LSTM, GRU, and 1D-CNN; 256 for Transformer	256 for BiGRU-Attention and Dilated 1D-CNN	A moderate batch size provides more frequent gradient updates and may improve minority-class learning.
4	**Learning rate**	$10^{-3}$for LSTM and 1D-CNN; $5 \times 10^{-4}$for GRU and Transformer	$5 \times 10^{-4}$for BiGRU-Attention; $7 \times 10^{-4}$for Dilated 1D-CNN	Model-specific learning rates improve convergence stability for recurrent and convolutional branches.
5	**Recurrent architecture**	LSTM 64–32 and GRU 64–32	BiGRU 128–64 with temporal attention	Bidirectional processing captures richer sequence context, while attention highlights the most informative flows.
6	**CNN configuration**	Conv1D 64–128 with pooling and global average pooling	Dilated Conv1D 128 with dilation rates 1, 2, and 4 plus residual connection	Dilated convolutions capture short- and long-range temporal patterns without requiring a very deep network.
7	**Random Forest**	100 trees, maximum depth 20	500 balanced trees with balanced subsampling	More balanced trees improve model diversity, robustness, and minority-class representation.
8	**Boosting model**	Not included	XGBoost with 700 trees, depth 10, learning rate 0.035, and subsample 0.90	Adds a strong boosted learner capable of modelling complex nonlinear feature interactions missed by Random Forest.

9	**Class-imbalance handling**	Inverse-frequency class weights	Class-weighted focal loss for deep models and training-only SMOTE for static models	Combines hard-sample-focused learning with controlled minority oversampling, improving rare-attack detection.
10	**Ensemble tuning**	Models evaluated independently	Five-fold out-of-fold stacking with Logistic Regression, $C = 1.5$, and 4000 maximum iterations	Learns complementary strengths of all base models while out-of-fold predictions reduce stacking overfitting.

Table 2 provides a detailed technical comparison between the hyperparameter-tuning settings adopted in the existing base paper and those used in the proposed LF-TSE-IDS framework, together with the rationale for each modification. The proposed method retains the effective sequence length of T=20 but introduces a stride of 3 to generate overlapping temporal windows and capture a greater number of behavioural patterns. The training duration is increased from 30 to 100 epochs with early stopping, allowing sufficient convergence while limiting overfitting. A moderate batch size of 256 is used for both deep-learning branches to enable more frequent gradient updates and improve sensitivity to minority attack classes. Model-specific learning rates are assigned to the BiGRU-Attention and Dilated 1D-CNN models to achieve stable and efficient convergence. The conventional LSTM and GRU structures are replaced by a deeper BiGRU 128−64 architecture with temporal attention, which captures bidirectional dependencies and emphasizes informative network-flow sequences. Similarly, the standard Conv1D model is enhanced through dilation rates of 1, 2, and 4 and a residual connection to learn multi-scale temporal patterns without excessively increasing network depth. The Random Forest is expanded from 100 to 500 balanced trees with balanced subsampling, improving ensemble diversity and minority-class representation, while XGBoost is additionally incorporated to model complex nonlinear feature interactions. Class imbalance is addressed through class-weighted focal loss for deep models and training-only SMOTE for static models, thereby strengthening rare-attack detection without introducing data leakage. Finally, the independent model evaluation used in the base paper is replaced by five-fold out-of-fold stacking with a Logistic Regression meta-learner, enabling the framework to combine complementary learner strengths while reducing overfitting and improving overall generalization.

4. Implementation and Result Discussion

This section presents the experimental implementation of the proposed LF-TSE-IDS framework and discusses the dataset configuration, computational environment, model training procedure, and evaluation process. The complete pipeline was implemented to perform preprocessing, leakage-free temporal-window construction, class-imbalance handling, base-learner training, out-of-fold probability generation, and Logistic Regression-based stacking. The BiGRU-Attention and Dilated 1D-CNN models were trained using sequential traffic windows, whereas Balanced Random Forest and XGBoost were trained using static flow-level features. The final performance was evaluated on an unseen test subset using accuracy, precision, recall, macro-F1, weighted-F1, MCC, ROC-AUC, PR-AUC, confusion matrix, and inference latency.

4.1 Hardware and Software Environment

The proposed LF-TSE-IDS framework was implemented using Google Colab with an NVIDIA Tesla T4 GPU to accelerate the training of the BiGRU-Attention and Dilated 1D-CNN models. The experimental environment used Python with TensorFlow/Keras for deep learning, Scikit-learn for preprocessing, evaluation, Logistic Regression, and Balanced Random Forest, XGBoost for gradient-boosted classification, Imbalanced-learn for training-only SMOTE, NumPy and Pandas for numerical and tabular data processing, and Matplotlib for result visualization. GPU acceleration, early stopping, batch processing, and model checkpointing were used to reduce computation time and improve training stability.

4.2 Dataset Description

The experimental analysis was conducted using the CICIDS2017 intrusion-detection dataset obtained from Kaggle. The dataset contains realistic benign and malicious network-traffic records representing normal activities and multiple cyberattacks, including denial-of-service, distributed denial-of-service, brute-force, botnet, infiltration, port-scanning, and web-based attacks. Each record contains flow-level statistical attributes such as packet counts, flow duration, packet-length statistics, inter-arrival times, header information, flag counts, and traffic-rate features. Before model training, duplicate records, missing values, and infinite values were handled, categorical attributes were

encoded, and numerical features were scaled. The traffic records were subsequently grouped using source IP, destination IP, source port, destination port, and protocol, ordered according to timestamps, and converted into temporal windows of length 20 with a stride of 3. The processed dataset was divided into 70% training, 15% validation, and 15% testing subsets, while class balancing was restricted to the training data to avoid information leakage

4.3 Illustrative analysis

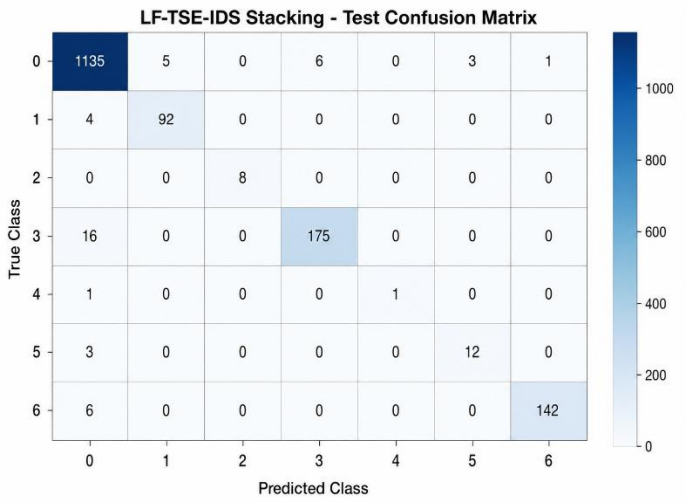

Figure 2. Test Confusion Matrix of the Proposed LF-TSE-IDS Stacking Model

Figure 2 presents the test confusion matrix of the proposed LF-TSE-IDS stacking model across seven traffic classes. The model correctly classifies most samples along the principal diagonal, including 1,135 samples from Class 0, 92 from Class 1, 8 from Class 2, 175 from Class 3, 1 from Class 4, 12 from Class 5, and 142 from Class 6. Only a limited number of samples are misclassified, with the most noticeable errors involving Classes 3, 5, and 6 being predicted as Class 0. The strong diagonal concentration demonstrates that the stacking framework effectively combines the complementary decisions of the base learners and provides robust multiclass intrusion detection performance.

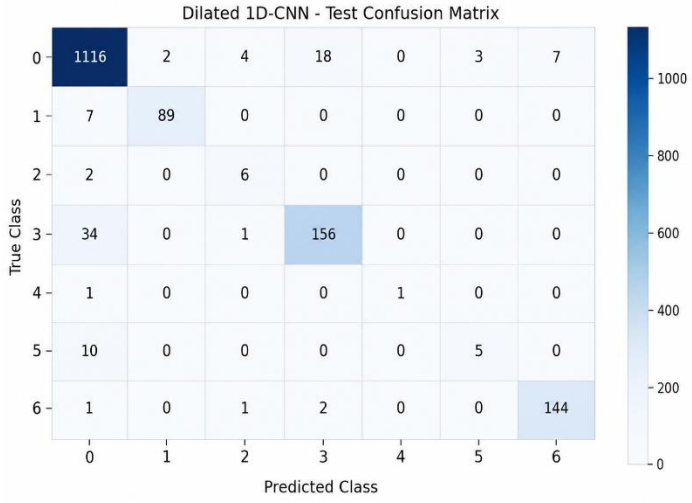

Figure 3. Test Confusion Matrix of the Dilated 1D-CNN Model

Figure 3 shows the test confusion matrix obtained from the Dilated 1D-CNN model. The model correctly identifies 1,116 Class 0 samples, 89 Class 1 samples, 6 Class 2 samples, 156 Class 3 samples, 1 Class 4 sample, 5 Class 5 samples, and 144 Class 6 samples. However, comparatively higher confusion is observed for minority classes, particularly Class 3 and Class 5, where several samples are incorrectly assigned to Class 0. The results indicate that

the dilated convolutional architecture successfully captures multi-scale temporal patterns, although its standalone performance is lower than that of the proposed stacking model for difficult and underrepresented attack categories.

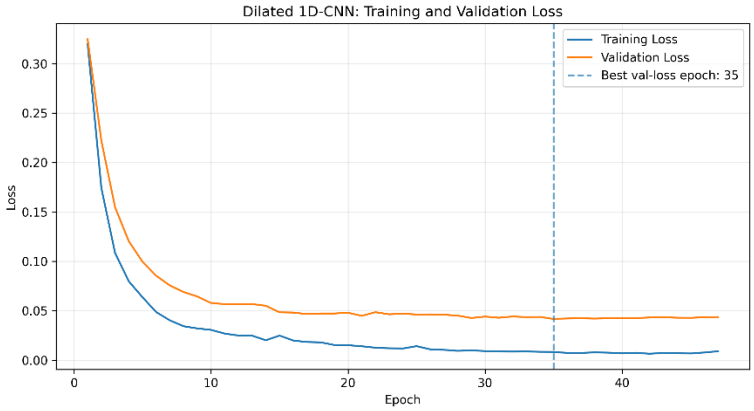

Figure 4. Training and Validation Loss of the Dilated 1D-CNN Model

Figure 4 illustrates the training and validation loss curves of the Dilated 1D-CNN model over approximately 47 epochs. Both losses decrease rapidly during the initial training stage, indicating effective learning and stable convergence. The training loss continues to decline to a very small value, whereas the validation loss stabilizes near 0.04 after approximately 20 epochs. The minimum validation loss is achieved at Epoch 35, as indicated by the dashed vertical line. The relatively small and stable difference between the two curves suggests satisfactory generalization, while early stopping prevents unnecessary training and limits overfitting.

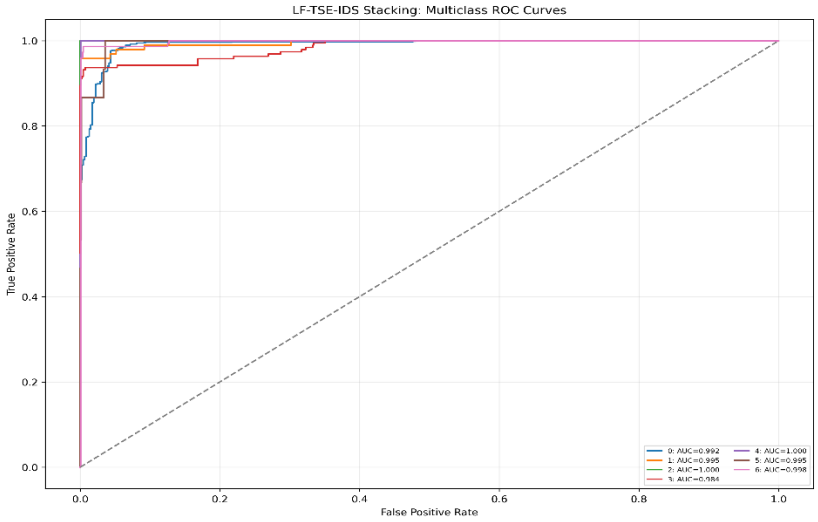

Figure 5. Multiclass ROC Curves of the Proposed LF-TSE-IDS Stacking Model

Figure 5 presents the one-vs-rest multiclass ROC curves of the proposed LF-TSE-IDS stacking model. All seven class-specific curves remain close to the upper-left corner, demonstrating high true-positive rates at very low false-positive rates. The reported AUC values are 0.992, 0.995, 1.000, 0.984, 1.000, 0.995, and 0.998 for Classes 0–6, respectively. Classes 2 and 4 achieve perfect discrimination with an AUC of 1.000, while even the lowest-performing class obtains an AUC of 0.984. These results confirm the strong discriminative capability and reliable multiclass generalization of the proposed stacking-based intrusion detection framework.

4.4 Result discussion

Figure 6. Test Weighted-F1 Comparison of the LF-TSE-IDS Models

Figure 6 presents the weighted-F1 scores of the proposed and baseline models. The LF-TSE-IDS stacking model achieves the highest weighted-F1 score, followed closely by XGBoost and Balanced Random Forest. BiGRU-Attention and Dilated 1D-CNN obtain slightly lower values. The result demonstrates that the proposed stacking framework provides a strong balance between precision and recall while accounting for the different support sizes of the traffic classes.

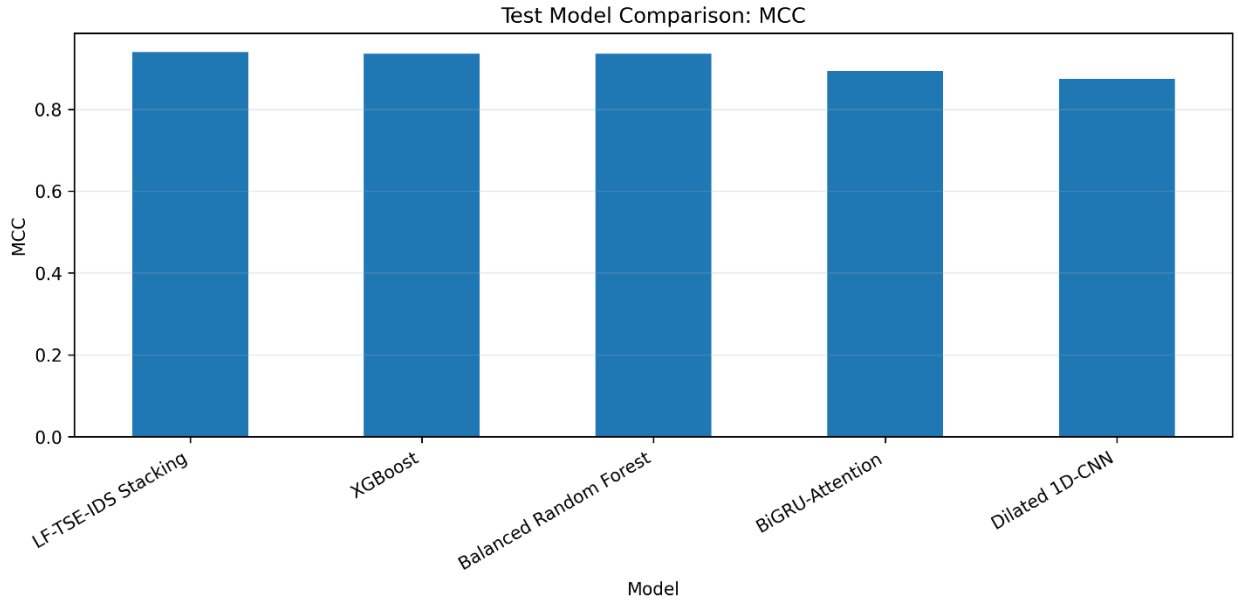

Figure 7. Test Matthews Correlation Coefficient Comparison of the LF-TSE-IDS Models

Figure 7 compares the Matthews correlation coefficient of the evaluated models. The proposed LF-TSE-IDS stacking model achieves the highest MCC, while XGBoost and Balanced Random Forest produce closely comparable results. BiGRU-Attention and Dilated 1D-CNN show slightly lower correlation values. The strong MCC obtained by

the stacking model indicates a reliable agreement between actual and predicted classes, even under the imbalanced multiclass distribution of the dataset.

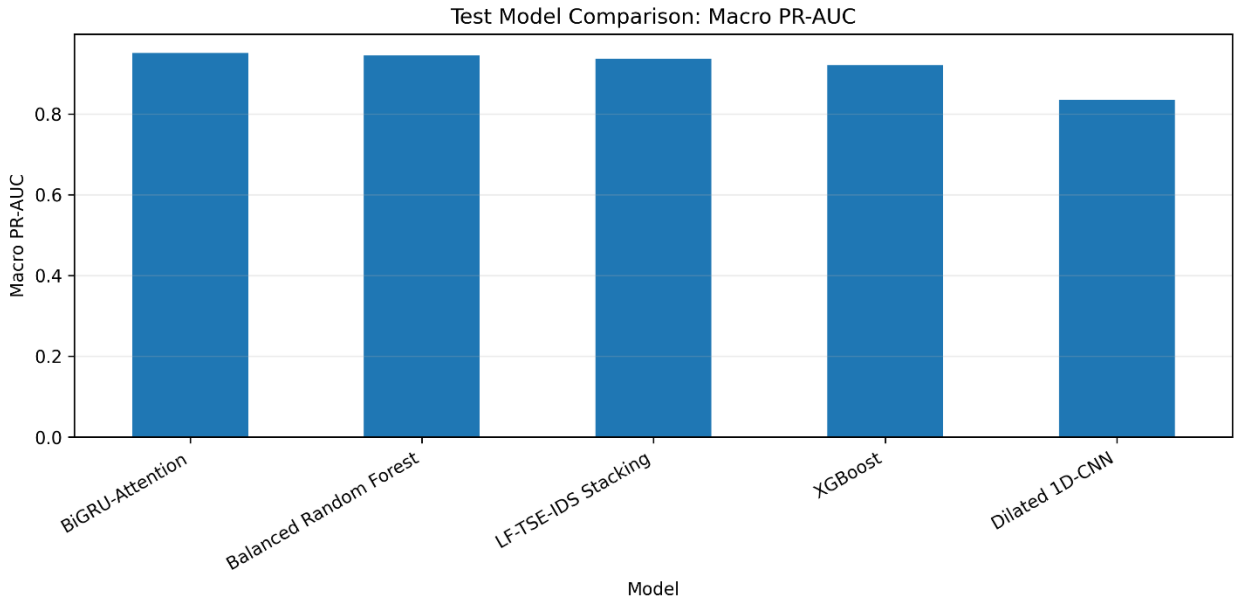

Figure 8. Test Macro PR-AUC Comparison of the LF-TSE-IDS Models

Figure 8 illustrates the macro-averaged precision–recall area under the curve for all evaluated models. BiGRU-Attention achieves the highest macro PR-AUC, followed closely by Balanced Random Forest and the proposed LF-TSE-IDS stacking model. XGBoost also provides competitive performance, whereas Dilated 1D-CNN obtains the lowest value. These results show that BiGRU-Attention is particularly effective at maintaining a favourable balance between precision and recall across imbalanced traffic classes.

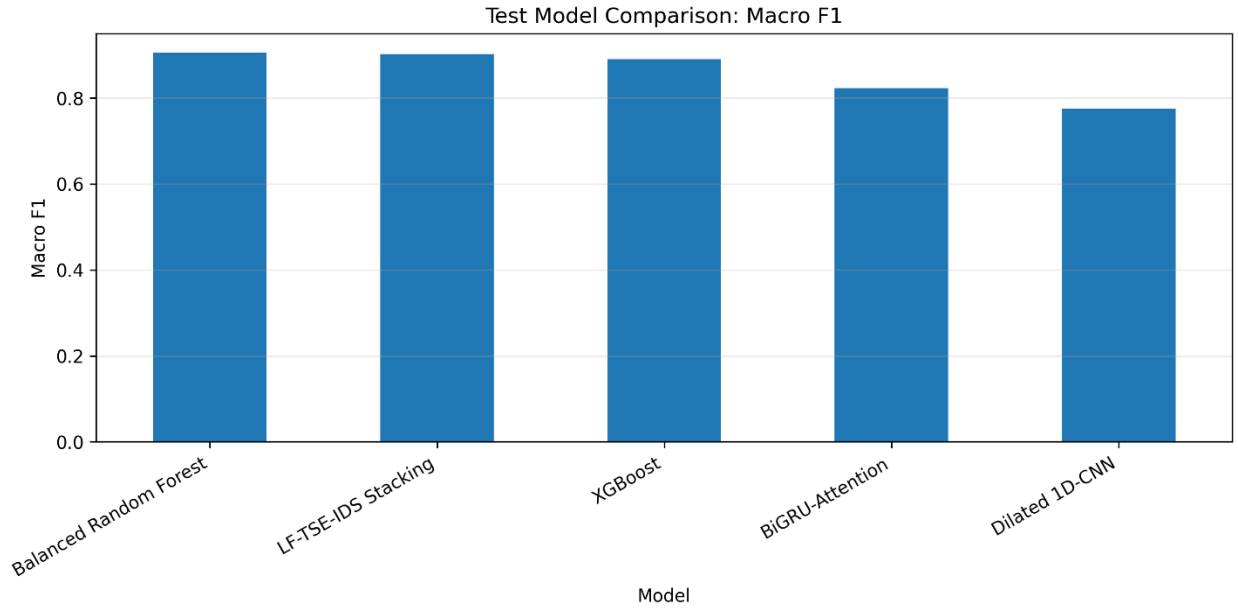

Figure 9. Test Macro-F1 Comparison of the LF-TSE-IDS Models

Figure 9 compares the macro-F1 scores of the five intrusion-detection models. Balanced Random Forest achieves the highest macro-F1 score, while the proposed LF-TSE-IDS stacking framework produces a nearly equivalent result. XGBoost also demonstrates strong class-balanced performance, whereas BiGRU-Attention and

Dilated 1D-CNN obtain lower values. Since macro-F1 assigns equal importance to every class, the result confirms that the ensemble-based approaches provide more consistent performance across minority and majority attack categories.

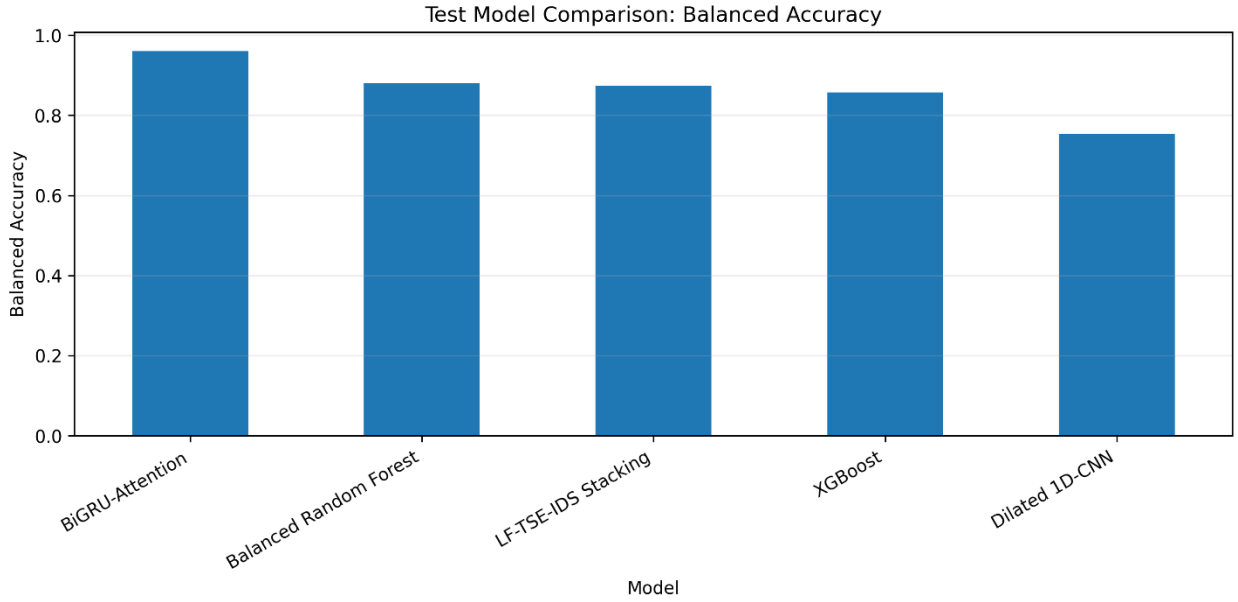

Figure 10. Test Balanced Accuracy Comparison of the LF-TSE-IDS Models

Figure 10 presents the balanced accuracy achieved by the evaluated models. BiGRU-Attention obtains the highest balanced accuracy, demonstrating strong sensitivity across both majority and minority traffic classes. Balanced Random Forest and the proposed LF-TSE-IDS stacking model provide the next-best performance, followed by XGBoost. Dilated 1D-CNN records the lowest balanced accuracy, indicating comparatively weaker recognition of underrepresented classes despite its high overall accuracy.

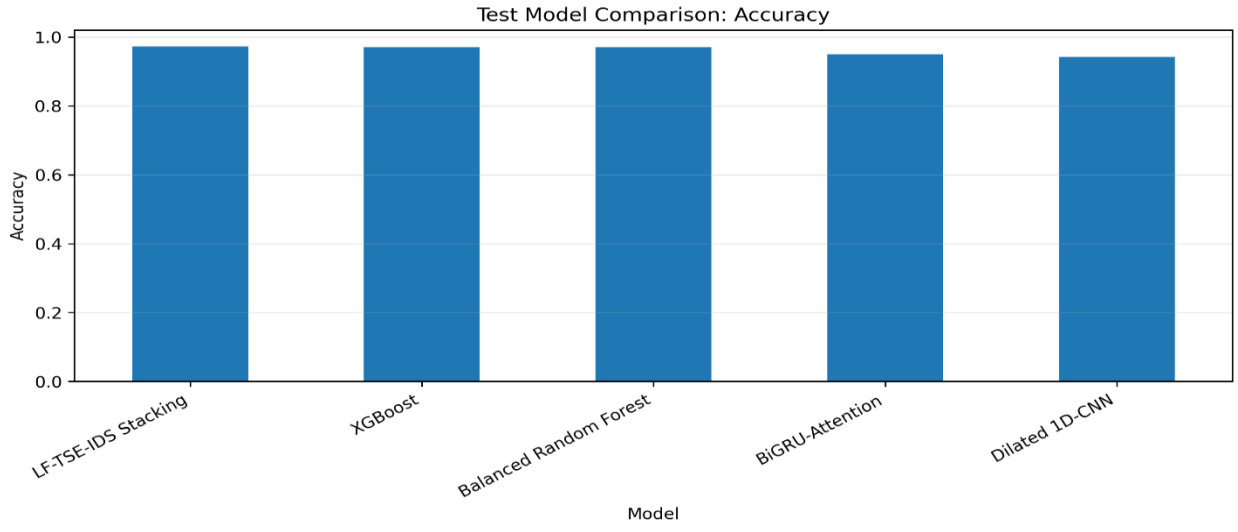

Figure 11. Test Accuracy Comparison of the LF-TSE-IDS Models

Figure 11 compares the test accuracy of the proposed LF-TSE-IDS stacking model with XGBoost, Balanced Random Forest, BiGRU-Attention, and Dilated 1D-CNN. The LF-TSE-IDS stacking model achieves the highest test accuracy, closely followed by XGBoost and Balanced Random Forest. BiGRU-Attention and Dilated 1D-CNN also obtain strong results but remain comparatively lower. The findings indicate that combining the complementary probability outputs of heterogeneous base learners through stacking improves the overall classification accuracy.

4.3 Comparative Performance Analysis

Table 3. Validation and Test Performance Comparison of the Proposed LF-TSE-IDS and Base Models

Model	Split	Accuracy	Balanced Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted F1	MCC	Micro ROC-AUC	Macro PR-AUC	Log Loss	Latency ms/sample	Training Time sec
BiGRU-Attention	Validation	0.94396	0.904527	0.740128	0.904527	0.789772	0.946742	0.879745	0.992197	0.924821	0.215921	2.301121	78.03664
Dilated 1D-CNN	Validation	0.943337	0.79756	0.861184	0.79756	0.808464	0.941446	0.874699	0.993106	0.859866	0.204221	0.316903	37.82902
Balanced Random Forest	Validation	0.976339	0.93205	0.979044	0.93205	0.953472	0.976254	0.948573	0.999102	0.967459	0.095845	0.299986	21.60957
XGBoost	Validation	0.980075	0.880598	0.980953	0.880598	0.918822	0.979898	0.956905	0.999495	0.922895	0.076225	0.305988	42.47416
LF-TSE-IDS Stacking	Validation	0.983811	0.910415	0.98544	0.910415	0.941569	0.983723	0.964923	0.998929	0.961076	0.078287	0.309844	1.864798
BiGRU-Attention	Test	0.950311	0.95992	0.764028	0.95992	0.82359	0.952583	0.893705	0.993114	0.950134	0.202723	0.734338	78.03664
Dilated 1D-CNN	Test	0.942236	0.75294	0.842294	0.75294	0.775475	0.940912	0.873596	0.989877	0.834846	0.225375	0.303244	37.82902
Balanced Random Forest	Test	0.970186	0.879972	0.954492	0.879972	0.905007	0.970008	0.935088	0.996724	0.944571	0.208011	0.29305	21.60957
XGBoost	Test	0.970186	0.856944	0.950065	0.856944	0.890776	0.970038	0.935226	0.998574	0.920509	0.129292	0.307295	42.47416

LF-TSE-IDS Stacking	Test	0.97205	0.874426	0.954651	0.874426	0.902486	0.97183	0.939177	0.995171	0.935645	0.153266	0.361496	1.864798

Table 3 presents the validation and test performance of BiGRU-Attention, Dilated 1D-CNN, Balanced Random Forest, XGBoost, and the proposed LF-TSE-IDS stacking framework using classification, discrimination, calibration, and computational-efficiency measures. On the validation set, LF-TSE-IDS Stacking achieves the highest accuracy of 98.38%, weighted-F1 of 98.37%, and MCC of 0.9649, demonstrating effective integration of the complementary base learners. XGBoost records the highest validation micro-ROC-AUC of 0.9995 and the lowest log loss of 0.0762, while Balanced Random Forest provides the highest validation macro-F1 of 0.9535. On the test set, the proposed stacking model obtains the best overall accuracy of 97.20%, weighted-F1 of 97.18%, and MCC of 0.9392, confirming strong generalization to unseen traffic. BiGRU-Attention achieves the highest balanced accuracy of 95.99%, macro recall of 95.99%, and macro-PR-AUC of 0.9501, indicating superior sensitivity across minority and majority classes. XGBoost produces the highest test micro-ROC-AUC of 0.9986 and the lowest test log loss of 0.1293, whereas Balanced Random Forest attains the highest test macro-F1 of 0.9050. In computational terms, static models provide low inference latency, while Dilated 1D-CNN is substantially faster than BiGRU-Attention. The reported stacking training time reflects only the Logistic Regression meta-learner and excludes the separate training times of the base learners. Overall, LF-TSE-IDS Stacking provides the most reliable balance of accuracy, weighted classification performance, and prediction consistency, while individual models retain advantages for specific class-balanced or probabilistic evaluation metrics.

5. Conclusion and Future Scope

This study proposed LF-TSE-IDS, a leakage-free temporal stacking ensemble framework for multiclass intrusion detection using the CICIDS2017 dataset. The framework integrates five-tuple-based traffic grouping, temporal ordering, overlapping window construction, training-only imbalance handling, and heterogeneous learning through BiGRU-Attention, Dilated 1D-CNN, Balanced Random Forest, and XGBoost. Their out-of-fold probability outputs are combined using a Logistic Regression meta-learner to reduce stacking bias and improve generalization. Experimental results demonstrate that the proposed model achieved a test accuracy of 97.20%, weighted-F1 of 97.18%, MCC of 0.9392, micro-ROC-AUC of 0.9952, and macro PR-AUC of 0.9356. The confusion matrix and class-wise ROC curves further confirm strong discrimination across most traffic categories, although some minority attacks remain challenging. Overall, LF-TSE-IDS provides a reliable balance between classification performance, prediction consistency, and computational efficiency compared with individual base learners. Future work will extend the framework to recent and cross-domain intrusion datasets, online concept-drift adaptation, explainable prediction, federated learning, lightweight edge deployment, automated hyperparameter optimization, and real-time validation under evolving zero-day and encrypted network attacks.

References

1. K. J, D. A, S. A, and E. A, “Ga-Gwo Hybrid Intelligent Intrusion Detection System,” in *Proc. 6th Int. Conf. Image Processing and Capsule Networks (ICIPCN)*, Dhulikhel, Nepal, 2026, pp. 1458–1462, doi: 10.1109/ICIPCN67432.2026.11438390.
2. N. Mahamud, M. J. Uddin, and U. Sumaiya, “Enhancing Network Security using Machine Learning for Automated Anomaly-based Intrusion Detection Systems for IoT Environment,” in *Proc. Int. Research Conf. Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka, 2025, pp. 1–6, doi: 10.1109/SCSE65633.2025.11031050.
3. D. Ather, R. Kanday, M. Rakhra, R. Kler, G. Aggarwal, and K. Jairath, “Adaptive Intrusion Detection Systems: Leveraging Machine Learning for Real-Time Threat Mitigation,” in *Proc. Int. Conf. Networks and Cryptology (NETCRYPT)*, New Delhi, India, 2025, pp. 1743–1746, doi: 10.1109/NETCRYPT65877.2025.11102773.
4. S. K, P. B. Venkata Ganesh, and K. Sai Siddhardha, “Real-Time Intrusion Detection for IoT Networks Using Machine Learning,” in *Proc. Int. Conf. Smart Futuristic Technology*, Bangalore, India, 2026, pp. 1–6, doi: 10.1109/ICSFT66733.2026.11508094.
5. B. Venugopal and A. J. M. Rani, “Implementation of a Decentralized Intrusion Detection System for IoT Networks,” in *Proc. 2nd Int. Conf. Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, Chennai, India, 2025, pp. 1–6, doi: 10.1109/RMKMATE64874.2025.11042386.
6. D. A. Mihaylova, “Adversarial Machine Learning Attacks Against Network Intrusion Detection Systems: Classification Analysis,” in *Proc. 60th Int. Scientific Conf. Information, Communication and Energy Systems and Technologies (ICEST)*, Ohrid, North Macedonia, 2025, pp. 1–4, doi: 10.1109/ICEST66328.2025.11098254.

7. S. M, S. SB, R. Mohandas, R. Pandimeena, V. M, and P. A, “A Lightweight Frame-Based Wireless Intrusion Detection System for Resource-Constrained Networks,” in *Proc. 8th Int. Conf. Intelligent Sustainable Systems (ICISS)*, Tirunelveli, India, 2026, pp. 1–8, doi: 10.1109/ICISS67859.2026.11453957.
8. P. Durole and M. Agarwal, “Implementing Naïve Bayes and Decision Forest Technique to Enhance Intrusion Detection Systems,” in *Proc. Int. Conf. Applications of Machine Intelligence and Data Analytics (ICAMIDA)*, Aurangabad, India, 2025, pp. 1–5, doi: 10.1109/ICAMIDA64673.2025.11209620.
9. D.-H. Tran and M. Park, “Enhancing GNN-Based Network Intrusion Detection Systems through Memory-Replay Approach,” in *Proc. Int. Conf. Information Networking (ICOIN)*, Chiang Mai, Thailand, 2025, pp. 510–512, doi: 10.1109/ICOIN63865.2025.10992956.
10. [10] W. Zhang, Z. Tian, Z. Li, T. Liu, T. Chong, and X. Lu, “Automotive Intrusion Detection and Prevention Systems: Technologies and Applications,” in *Proc. 6th Int. Conf. Computer Engineering and Intelligent Control (ICCEIC)*, Guangzhou, China, 2025, pp. 189–192, doi: 10.1109/ICCEIC67916.2025.11308861.
11. G. A. Zeeshan, K. Soma, S. A. Khan, N. Kalita, S. K. Sulaiman, and M. Karthik, “Cyber-Physical Intrusion Detection with Spider Monkey Optimized GRU-LSTM and Blockchain Security,” in *Proc. Int. Conf. Next Generation Information System Engineering (NGISE)*, Ghaziabad, Delhi (NCR), India, 2025, pp. 1–5, doi: 10.1109/NGISE64126.2025.11085348.
12. A. Gaurav, V. Arya, J. Wu, K. T. Chui, and B. B. Gupta, “Lightweight AI-Enhanced Intrusion Detection System for Zero-Day Threat Detection in Consumer Electronics Devices,” in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE)*, Dubai, United Arab Emirates, 2026, pp. 1–6, doi: 10.1109/ICCE67443.2026.11449631.
13. F. Kadri and P. Singla, “Enhancing Intrusion Detection Systems with Deep Learning and Machine Learning Algorithms for Real-Time Threat Classification,” in *Proc. Int. Conf. Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, Erode, India, 2025, pp. 957–963, doi: 10.1109/ICMSCI62561.2025.10894403.
14. Y. Zhang, “Network Intrusion Detection and Defense System Based on Deep Learning,” in *Proc. 2nd Int. Conf. Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, Hassan, India, 2025, pp. 1–6, doi: 10.1109/IACIS65746.2025.11211034.
15. N. Niknami and J. Wu, “RAN-IDS: Adaptive Continual Learning for Intrusion Detection in Open RAN,” in *Proc. Int. Conf. Computing, Networking and Communications (ICNC)*, Maui, HI, USA, 2026, pp. 1–5, doi: 10.1109/ICNC68183.2026.11416960.
16. S. K. James and S. Amutha, “Implications of Artificial Intelligence for Real-Time Intrusion Detection and Anomaly Detection in Smart Networks Concerning Cybersecurity in the Internet of Things,” in *Proc. Int. Conf. Electronics, Computing, Communication and Control Technology (ICECCC)*, Bengaluru, India, 2025, pp. 1–6, doi: 10.1109/ICECCC65144.2025.11064232.
17. J. Zhang, Q. Liu, S. Hou, C. Zong, Y. Yang, and J. Ye, “Research on 3D Point Cloud Discretization Data Processing and Laser Intrusion Detection System,” in *Proc. IEEE 8th Advanced Information Technology, Electronic and Automation Control Conf. (IAEAC)*, Guiyang, China, 2025, pp. 534–538, doi: 10.1109/IAEAC65194.2025.11166611.
18. S. Waskle, L. Parashar, and U. Singh, “Intrusion Detection System Using PCA with Random Forest Approach,” in *Proc. Int. Conf. Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 803–808, doi: 10.1109/ICESC48915.2020.9155656.
19. K. J, D. A, S. A, and E. A, “Ga-Gwo Hybrid Intelligent Intrusion Detection System,” in *Proc. 6th Int. Conf. Image Processing and Capsule Networks (ICIPCN)*, Dhulikhel, Nepal, 2026, pp. 1458–1462, doi: 10.1109/ICIPCN67432.2026.11438390.
20. N. Mahamud, M. J. Uddin, and U. Sumaiya, “Enhancing Network Security using Machine Learning for Automated Anomaly-based Intrusion Detection Systems for IoT Environment,” in *Proc. Int. Research Conf. Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka, 2025, pp. 1–6, doi: 10.1109/SCSE65633.2025.11031050.
21. D. Ather, R. Kanday, M. Rakhra, R. Kler, G. Aggarwal, and K. Jairath, “Adaptive Intrusion Detection Systems: Leveraging Machine Learning for Real-Time Threat Mitigation,” in *Proc. Int. Conf. Networks and Cryptology (NETCRYPT)*, New Delhi, India, 2025, pp. 1743–1746, doi: 10.1109/NETCRYPT65877.2025.11102773.
22. S. K, P. B. Venkata Ganesh, and K. Sai Siddhardha, “Real-Time Intrusion Detection for IoT Networks Using Machine Learning,” in *Proc. Int. Conf. Smart Futuristic Technology*, Bangalore, India, 2026, pp. 1–6, doi: 10.1109/ICSFT66733.2026.11508094.
23. B. Venugopal and A. J. M. Rani, “Implementation of a Decentralized Intrusion Detection System for IoT Networks,” in *Proc. 2nd Int. Conf. Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, Chennai, India, 2025, pp. 1–6, doi: 10.1109/RMKMATE64874.2025.11042386.
24. D. A. Mihaylova, “Adversarial Machine Learning Attacks Against Network Intrusion Detection Systems: Classification Analysis,” in *Proc. 60th Int. Scientific Conf. Information, Communication and Energy Systems and Technologies (ICEST)*, Ohrid, North Macedonia, 2025, pp. 1–4, doi: 10.1109/ICEST66328.2025.11098254.
25. S. M, S. SB, R. Mohandas, R. Pandimeena, V. M, and P. A, “A Lightweight Frame-Based Wireless Intrusion Detection System for Resource-Constrained Networks,” in *Proc. 8th Int. Conf. Intelligent Sustainable Systems (ICISS)*, Tirunelveli, India, 2026, pp. 1–8, doi: 10.1109/ICISS67859.2026.11453957.
26. P. Durole and M. Agarwal, “Implementing Naïve Bayes and Decision Forest Technique to Enhance Intrusion Detection Systems,” in *Proc. Int. Conf. Applications of Machine Intelligence and Data Analytics (ICAMIDA)*, Aurangabad, India, 2025, pp. 1–5, doi: 10.1109/ICAMIDA64673.2025.11209620.

27. D.-H. Tran and M. Park, “Enhancing GNN-Based Network Intrusion Detection Systems through Memory-Replay Approach,” in *Proc. Int. Conf. Information Networking (ICOIN)*, Chiang Mai, Thailand, 2025, pp. 510–512, doi: 10.1109/ICOIN63865.2025.10992956.
28. W. Zhang, Z. Tian, Z. Li, T. Liu, T. Chong, and X. Lu, “Automotive Intrusion Detection and Prevention Systems: Technologies and Applications,” in *Proc. 6th Int. Conf. Computer Engineering and Intelligent Control (ICCEIC)*, Guangzhou, China, 2025, pp. 189–192, doi: 10.1109/ICCEIC67916.2025.11308861.
29. G. A. Zeeshan, K. Soma, S. A. Khan, N. Kalita, S. K. Sulaiman, and M. Karthik, “Cyber-Physical Intrusion Detection with Spider Monkey Optimized GRU-LSTM and Blockchain Security,” in *Proc. Int. Conf. Next Generation Information System Engineering (NGISE)*, Ghaziabad, Delhi (NCR), India, 2025, pp. 1–5, doi: 10.1109/NGISE64126.2025.11085348.
30. A. Gaurav, V. Arya, J. Wu, K. T. Chui, and B. B. Gupta, “Lightweight AI-Enhanced Intrusion Detection System for Zero-Day Threat Detection in Consumer Electronics Devices,” in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE)*, Dubai, United Arab Emirates, 2026, pp. 1–6, doi: 10.1109/ICCE67443.2026.11449631.
31. F. Kadri and P. Singla, “Enhancing Intrusion Detection Systems with Deep Learning and Machine Learning Algorithms for Real-Time Threat Classification,” in *Proc. Int. Conf. Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, Erode, India, 2025, pp. 957–963, doi: 10.1109/ICMSCI62561.2025.10894403.
32. Y. Zhang, “Network Intrusion Detection and Defense System Based on Deep Learning,” in *Proc. 2nd Int. Conf. Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, Hassan, India, 2025, pp. 1–6, doi: 10.1109/IACIS65746.2025.11211034.
33. N. Niknami and J. Wu, “RAN-IDS: Adaptive Continual Learning for Intrusion Detection in Open RAN,” in *Proc. Int. Conf. Computing, Networking and Communications (ICNC)*, Maui, HI, USA, 2026, pp. 1–5, doi: 10.1109/ICNC68183.2026.11416960.
34. S. K. James and S. Amutha, “Implications of Artificial Intelligence for Real-Time Intrusion Detection and Anomaly Detection in Smart Networks Concerning Cybersecurity in the Internet of Things,” in *Proc. Int. Conf. Electronics, Computing, Communication and Control Technology (ICECCC)*, Bengaluru, India, 2025, pp. 1–6, doi: 10.1109/ICECCC65144.2025.11064232.
35. J. Zhang, Q. Liu, S. Hou, C. Zong, Y. Yang, and J. Ye, “Research on 3D Point Cloud Discretization Data Processing and Laser Intrusion Detection System,” in *Proc. IEEE 8th Advanced Information Technology, Electronic and Automation Control Conf. (IAEAC)*, Guiyang, China, 2025, pp. 534–538, doi: 10.1109/IAEAC65194.2025.11166611.
36. S. Waskle, L. Parashar, and U. Singh, “Intrusion Detection System Using PCA with Random Forest Approach,” in *Proc. Int. Conf. Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 803–808, doi: 10.1109/ICESC48915.2020.9155656.