

A Deep Learning-Based Intelligent Framework for Real-Time Network Traffic Classification and Anomaly Detection in Distributed Computing Systems

Dipannita Mondal*

Professor, Department of Artificial Intelligence- Data Science and Machine Learning, Dr. D. Y. Patil College of Engineering and Innovation Talegaon, Varale, Pune, Maharashtra, India

*Corresponding Author: ddmondal2684@gmail.com

Peer Review Information	Abstract
Type: Article Received: 25 July 2026 Revised: 22 Aug 2026 Accepted: 18 Sep 2026 Published: 14 Oct 2026	The increasing deployment of distributed computing systems, cloud platforms, edge networks, and interconnected digital services has significantly increased the complexity of network traffic monitoring and cybersecurity management. Conventional network traffic classification and anomaly detection techniques frequently encounter limitations associated with high-dimensional traffic data, evolving attack patterns, heterogeneous communication protocols, and strict real-time processing requirements. This study proposes a deep learning-based intelligent framework integrating a one-dimensional Convolutional Neural Network (1D-CNN), Bidirectional Gated Recurrent Unit (BiGRU), and temporal attention mechanism for network traffic classification and anomaly detection in distributed computing environments. The proposed architecture combines local feature extraction, sequential dependency modeling, and attention-based representation learning to distinguish legitimate network activity from suspicious traffic patterns. A distributed monitoring architecture is introduced in which network flow features are collected from geographically or logically separated computing nodes and processed using a centralized or edge-assisted inference service. The framework incorporates preprocessing, normalization, class imbalance management, temporal window construction, supervised classification, and anomaly score estimation. A reproducible experimental methodology is proposed using the CICIDS2017 and UNSW-NB15 datasets, with comparative evaluation against traditional machine learning and deep learning baselines. Performance measures include accuracy, precision, recall, macro-F1-score, false-positive rate, inference latency, and processing throughput. The proposed methodology additionally examines the effect of feature extraction, temporal modeling, and attention through ablation experiments. The study establishes an architectural and experimental foundation for investigating scalable, intelligent network traffic monitoring, while empirical validation is required to determine its detection effectiveness and real-time suitability. Keywords: Deep Learning, Network Traffic Classification, Anomaly Detection, Distributed Computing, CNN-BiGRU, Real-Time Intrusion Detection.

How to Cite This Article

Mondal, D. (2026). A Deep Learning-Based Intelligent Framework for Real-Time Network Traffic Classification and Anomaly Detection in Distributed Computing Systems. *International Journal on Advanced Computer Theory and Engineering* 15(2), 157-163

Introduction

Distributed computing systems have become essential to modern digital infrastructure by enabling computational processes to operate across multiple interconnected devices, servers, cloud platforms, and geographically separated environments. Distributed architectures support scalable processing, improved service availability, resource sharing, and flexible application deployment. These capabilities have encouraged the adoption of distributed systems in enterprise networks, cloud computing, industrial automation, Internet of Things applications, and edge computing infrastructures. Despite these advantages, the expanding connectivity of distributed systems introduces significant challenges related to network monitoring, traffic classification, and cybersecurity. Communication between distributed nodes produces large volumes of heterogeneous traffic with varying protocols, transmission patterns, packet sizes, and communication frequencies. Identifying legitimate activities and distinguishing them from malicious or abnormal behavior becomes particularly difficult when traffic conditions change dynamically. Network traffic classification involves identifying or categorizing traffic according to characteristics such as application type, communication behavior, or security status. Anomaly detection focuses on recognizing patterns that deviate from expected network behavior. Both functions are important for maintaining system reliability, identifying potential attacks, and supporting network management decisions.

Conventional traffic monitoring approaches frequently depend on packet signatures, protocol information, predefined rules, or manually configured thresholds. Although these approaches can provide efficient detection of known patterns, their effectiveness may decline when attacks evolve, traffic is encrypted, or legitimate workloads generate unusual communication behavior. In distributed systems, these challenges are intensified by dynamic node participation, workload migration, heterogeneous network conditions, and the possibility of attacks originating from multiple locations. Machine learning methods have been investigated as alternatives to purely rule-based traffic analysis. Algorithms such as Support Vector Machines, Decision Trees, Random Forest, and Gradient Boosting can learn statistical relationships between traffic features and observed network conditions. However, the effectiveness of conventional classifiers depends on the quality of feature engineering and their ability to represent complex dependencies within network traffic. Deep learning provides additional representation-learning capabilities through multiple computational layers. Convolutional Neural Networks can extract local patterns from structured feature sequences, while recurrent architecture such as Long Short-Term Memory and Gated Recurrent Unit networks can capture sequential dependencies. Attention mechanisms further support the selective weighting of relevant representations during classification.

Imrana et al. (2024) examined CNN-GRU architecture with feature fusion for network intrusion detection. Their research demonstrated the relevance of combining convolutional and recurrent structures for learning complementary network traffic characteristics. Nikitenko and Bashkov (2024) investigated CNN-BiGRU-Attention architecture, providing particularly close prior work for the proposed model. Recent surveys have also emphasized the importance of evaluating intrusion detection systems under realistic network conditions. Ruffo et al. (2024) reviewed deep learning-based anomaly and intrusion detection in software-defined networks, identifying challenges involving datasets, preprocessing, algorithm configuration, and experimental evaluation. Kohli and Chhabra (2025) examined deep learning approaches for anomaly detection and highlighted the continued need for hybrid architectures, interpretability, and suitable evaluation measures.

Literature Review

Imrana et al. (2024) proposed a CNN-GRU feature fusion intrusion detection model. Their research combined convolutional and recurrent learning to examine how complementary network traffic representations could improve classification. The study provides methodological support for hybrid architecture and highlights the importance of feature processing and class imbalance management. Nikitenko and Bashkov (2024) developed a network intrusion detection system combining CNN, BiGRU, and attention. Their work demonstrates that this specific architectural combination has already been explored. Accordingly, the present research must differentiate itself through its distributed monitoring design, latency-sensitive evaluation, and experimental protocol rather than claim novelty in the neural-network combination alone.

Yang et al. (2024) investigated an intrusion detection architecture for the Industrial Internet of Things using attention mechanisms, BiGRU, and an Inception-CNN structure. Their research illustrates the applicability of hybrid sequential and convolutional learning in connected industrial environments. It also demonstrates the importance of evaluating model generalization across heterogeneous communication conditions. Ruffo et al. (2024) presented a survey of anomaly and intrusion detection using deep learning in software-defined networks. The study reviewed commonly used datasets, preprocessing strategies, architectures, hyperparameter selection, and evaluation practices. It identified methodological challenges that remain relevant to distributed network monitoring systems.

Kohli and Chhabra (2025) surveyed deep learning models for anomaly detection, examining existing techniques, performance measures, and research limitations. Their work reinforces the importance of balancing model complexity with application requirements and identifies interpretability and hybrid learning as significant directions for further investigation. Although hybrid deep learning methods have demonstrated promising benchmark performance, several unresolved deployment considerations remain. Distributed computing environments produce network traffic at different locations and rates, making centralized collection and analysis potentially expensive. Real-time monitoring must also account for data transfer overhead, buffering delays, memory consumption, and inference latency. Furthermore, classification models trained using historical attack labels may experience reduced effectiveness when encountering previously unseen or significantly modified attack patterns.

Methodology

Research Design

This research adopts a quantitative, design-oriented experimental methodology to develop and evaluate an intelligent network traffic monitoring framework. The system is designed around distributed traffic acquisition, preprocessing, temporal representation learning, supervised classification, and anomaly alert generation.

Let a network traffic dataset be represented as:

$$D = \{(X_i, y_i)\}_{i=1}^N \quad \text{-----}(1)$$

where (Xi) denotes the feature representation of the (i)-th observation, (yi) represents its corresponding traffic label, and (N) denotes the total number of observations. For binary classification, labels represent benign and malicious traffic. For multiclass classification, the labels correspond to the attack categories available in the dataset.

Proposed Methodology Architecture

The proposed framework begins by collecting network flow information from distributed monitoring agents. Each agent extracts relevant metadata and transmits structured records to an inference service or executes a lightweight version of the classification model locally. The preprocessing module validates the extracted traffic features and constructs temporally ordered input sequences. The CNN component extracts local representations from these sequences. The BiGRU module captures forward and backward temporal relationships within completed observation windows, while the attention layer aggregates information from relevant time steps. The final classification module generates traffic probabilities and supports alert generation using a configurable risk threshold.

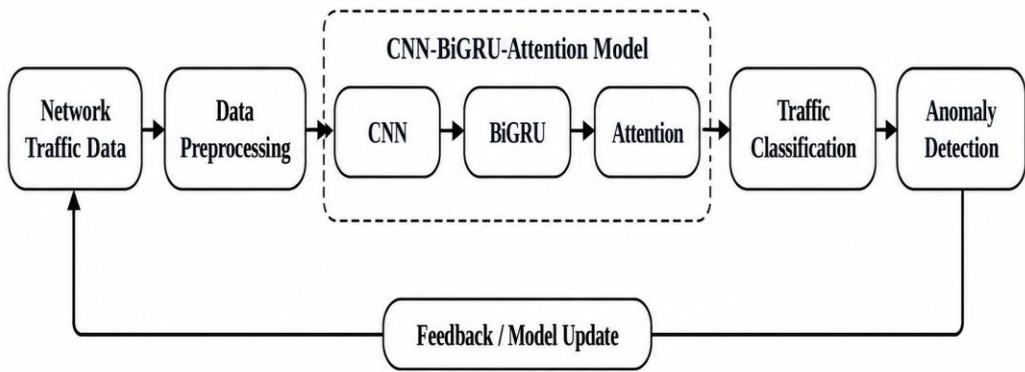


Figure 1. Proposed Distributed Network Traffic Classification and Anomaly Detection Architecture Using CNN-BiGRU-Attention

Figure 1 illustrates the proposed distributed network traffic classification and anomaly detection architecture integrating Convolutional Neural Networks (CNN), Bidirectional Gated Recurrent Units (BiGRU), and an attention mechanism. The framework begins with network traffic data collection and preprocessing, followed by CNN-based feature extraction, BiGRU-based temporal dependency learning, and attention-based feature refinement. The extracted features are used for traffic classification and anomaly detection to distinguish normal network behavior from potential security threats. A continuous feedback mechanism supports model updates and adaptive learning, improving detection accuracy, classification efficiency, and responsiveness to evolving network traffic patterns in distributed environments.

Data Preprocessing

The preprocessing stage is designed to improve data quality and ensure compatibility with deep learning architectures. Network flow records are examined for missing observations, duplicated entries, invalid values, inconsistent feature types, and incomplete labels.

Numerical features are standardized using:

$$X' = \frac{X - \mu}{\sigma} \quad \text{-----}(2)$$

where (X) represents the original feature value, (μ) denotes the training-set mean, and (σ) represents the corresponding standard deviation. Constant features are handled separately to prevent invalid transformations. Categorical attributes are encoded using appropriate representations. Class weighting or training-only resampling may be considered for imbalanced attack categories. Preprocessing parameters must be estimated from training data without using the final test observations.

Temporal Sequence Construction

Network traffic records are organized into chronologically ordered sequences based on capture time, monitoring node, or communication context. For each sequence, the model receives a tensor:

$$X \in \mathbb{R}^{T \times F} \quad \text{-----}(3)$$

where (T) represents the number of observations in the temporal window and (F) denotes the number of traffic features.

Sequence construction is necessary because the BiGRU component is intended to model genuine temporal dependencies. Arbitrarily reshaping independent tabular records into sequences would not establish meaningful chronological structure.

For real-time monitoring, a sliding-window mechanism is proposed in which new observations update the window and initiate classification when sufficient data are available. Window duration and stride influence both detection capability and alert latency.

Convolutional Neural Network Layer

The proposed model uses a one-dimensional CNN to learn local patterns from network traffic representations. A convolutional operation is formulated as:

$$C_t^{(k)} = \sigma\left(\sum_{j=0}^{q-1} W_j^{(k)} X_{t+j} + b^{(k)}\right) \quad \text{-----}(4)$$

where ($C_t^{(k)}$) denotes the output of filter (k), ($W_j^{(k)}$) represents its convolutional weights, (q) indicates the kernel size, and (σ) denotes the activation function.

The convolutional component is intended to extract short-range temporal patterns associated with traffic bursts, changes in connection characteristics, and other localized behavior. Its output is supplied to the recurrent modeling layer.

Network Traffic Classification

The final classifier employs a dense neural layer followed by a softmax function:

$$P(y = c \mid X) = \frac{\exp(s_c)}{\sum_{j=1}^K \exp(s_j)} \quad \text{-----}(5)$$

where (K) represents the number of traffic classes and (s_c) denotes the corresponding output score.

For multiclass classification, the model predicts the likelihood of each known traffic class. The predicted label is selected from the class with the highest probability, subject to an optional rejection or review policy for uncertain predictions.

Proposed Experimental Setup

Experimental Configuration

Table 1. Proposed implementation parameters

Parameter	Proposed configuration
Programming language	Python
Deep learning framework	PyTorch or TensorFlow
Data processing	pandas, NumPy
CNN filters	64 and 128
Convolution kernel	3
BiGRU hidden units	64
Attention	Additive temporal attention
Dropout	0.30
Optimizer	Adam
Learning rate	0.001
Batch size	64

Training epochs	Up to 100
Early stopping	Validation loss
Loss function	Weighted cross-entropy
Main evaluation	Macro-F1, recall, false-positive rate
Operational evaluation	Alert latency and throughput

These values are initial experimental settings and may be adjusted through development-set validation.

Results and Discussion

Comparative Evaluation Framework

The proposed experimental analysis compares CNN-BiGRU-Attention architecture with conventional machine learning models and individual deep learning approaches.

Table 2. Proposed classification performance comparison

Algorithm	Accuracy (%)	Macro-F1	FPR (%)
Random Forest	Pending	Pending	Pending
XGBoost	Pending	Pending	Pending
1D-CNN	Pending	Pending	Pending
GRU	Pending	Pending	Pending
BiGRU	Pending	Pending	Pending
CNN-GRU	Pending	Pending	Pending
CNN-BiGRU	Pending	Pending	Pending
CNN-BiGRU-Attention	Pending	Pending	Pending

All entries are marked pending because numerical experiments have not been conducted for this manuscript. Actual measurements are required before claiming that the proposed architecture outperforms the alternatives.

Analysis of Hybrid Deep Learning

The proposed model integrates convolutional feature extraction with bidirectional sequential modeling. The CNN component is intended to recognize short-range traffic patterns, while BiGRU processes dependencies within chronological observation windows. Attention-based aggregation provides an additional mechanism for weighting learned temporal representations. These characteristics may be valuable for detecting traffic activities involving changes over time. However, the actual benefit depends on the information contained within the selected input windows. If individual records have limited temporal dependence, recurrent processing may introduce computational overhead without improving classification performance. The 2024 study by Imrana et al. and the 2025 study by Elshewey et al. provide relevant evidence that CNN-GRU combinations can perform effectively on intrusion detection benchmarks. Their findings motivate the proposed comparison but do not establish the performance of this framework.

Anomaly Detection Performance

The effectiveness of an anomaly alert mechanism depends on both its sensitivity to malicious traffic and its tendency to generate false alarms. A highly sensitive model may detect more attacks but also classify unusual legitimate behavior as suspicious. Conversely, a conservative threshold may reduce false alerts while overlooking lower-confidence attacks. The proposed framework addresses this trade-off through validation-based threshold selection and explicit false-positive reporting. Precision–recall curves and threshold-dependent detection measurements should be used to characterize the available operating points. Another important consideration is class imbalance. If the majority of observations are benign, a model may achieve high overall accuracy while performing poorly on rare attacks. Macro-F1, class-specific recall, and precision–recall analysis are therefore necessary to evaluate minority attack detection.

Real-Time Distributed Performance

The proposed architecture is designed to support distributed network monitoring, but real-time capability must be demonstrated through end-to-end measurements. The computational time required for neural inference represents only one component of total detection latency. Traffic collection, feature extraction, temporal window construction, communication between nodes, and alert delivery can collectively introduce substantial delays. A model with low inference time may therefore fail to satisfy an application-specific alert deadline when deployed across distributed infrastructure.

Table 3. Proposed operational evaluation

Configuration	Alert latency (ms)	Throughput (flows/s)
---------------	--------------------	----------------------

Centralized CNN	Pending	Pending
Centralized CNN-GRU	Pending	Pending
Centralized proposed model	Pending	Pending
Edge-assisted proposed model	Pending	Pending

The comparison should use consistent hardware, identical traffic workloads, and explicitly defined collection and transport conditions.

Ablation Analysis

Ablation experiments are required to determine the contribution of individual components. The proposed evaluation compares CNN alone, BiGRU alone, CNN-BiGRU without attention, and the complete CNN-BiGRU-Attention model. If the full architecture improves predictive metrics, the result must be considered alongside additional parameters, inference latency, and memory requirements. An improvement in detection accuracy may not justify significantly greater processing overhead in latency-sensitive networks. The distributed preprocessing component should also be evaluated independently to determine whether local feature extraction reduces communication volume and how it affects overall monitoring performance.

Conclusion

This research proposes a deep learning-based intelligent framework for real-time network traffic classification and anomaly detection in distributed computing systems. The framework integrates 1D-CNN feature extraction, BiGRU temporal modeling, an attention mechanism, and distributed traffic monitoring to support the identification of legitimate and suspicious network activities. The proposed architecture addresses the need to connect intelligent traffic classification with operational monitoring requirements. Its distributed collection layer is intended to support traffic acquisition across multiple computing nodes, while the hybrid neural network performs classification using temporally ordered flow information. An anomaly scoring mechanism converts estimated attack probabilities into configurable security alerts. The literature published through 2025 demonstrates that CNN-GRU, CNN-BiGRU, and attention-based intrusion detection models are established areas of research. Therefore, the intended contribution of the present study lies in the proposed deployment-oriented integration and systematic evaluation of classification performance, false-positive control, throughput, and end-to-end alert latency. The experimental methodology incorporates publicly available intrusion detection datasets, comparative baseline models, ablation analysis, and operational measurements. These procedures are designed to determine whether the proposed framework provides useful improvements under realistic network monitoring constraints. Future research may investigate lightweight neural network architectures, federated learning, online anomaly detection, model compression, concept-drift adaptation, and open-set attack recognition. Further work should also examine privacy-preserving traffic analysis and secure communication between distributed monitoring agents. The proposed framework provides a foundation for evaluating intelligent traffic monitoring in distributed computing environments. However, actual performance improvements and real-time applicability must be established through reproducible experimentation and measured deployment results.

References

1. Elshewey, A. M., Abbas, S., Osman, A. M., Aldakheel, E. A., & Fouad, Y. (2025). DDoS classification of network traffic in software defined networking SDN using a hybrid convolutional and gated recurrent neural network. *Scientific Reports*, 15, Article 29122. <https://doi.org/10.1038/s41598-025-13754-1>
2. Imrana, Y., Xiang, Y., Ali, L., Noor, A., & Sarpong, K. (2024). CNN-GRU-FF: A double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units. *Complex & Intelligent Systems*, 10, 3353–3370. <https://doi.org/10.1007/s40747-023-01313-y>
3. Kohli, M., & Chhabra, I. (2025). A comprehensive survey on techniques, challenges, evaluation metrics and applications of deep learning models for anomaly detection. *Discover Applied Sciences*, 7, Article 784. <https://doi.org/10.1007/s42452-025-07312-7>
4. Nikitenko, A., & Bashkov, Y. (2024). Construction of a network intrusion detection system based on a convolutional neural network and a bidirectional gated recurrent unit with attention mechanism. *Eastern-European Journal of Enterprise Technologies*. <https://doi.org/10.15587/1729-4061.2024.305685>
5. Ruffo, V. G. da S., Lent, D. M. B., Komarchesqui, M., Schiavon, V. F., de Assis, M. V. O., Carvalho, L. F., & Proença, M. L., Jr. (2024). Anomaly and intrusion detection using deep learning for software-defined networks: A survey. *Expert Systems with Applications*, 256, Article 124982. <https://doi.org/10.1016/j.eswa.2024.124982>
6. Yang, K., Wang, J., & Li, M. (2024). An improved intrusion detection method for IIoT using attention mechanisms, BiGRU, and Inception-CNN. *Scientific Reports*, 14, Article 19339. <https://doi.org/10.1038/s41598-024-70094-2>

7. Sunkara, S. P. (2025). Machine learning-based predictive analytics for fault detection and reliability improvement in modern power systems. *International Journal of Electrical Engineering and Technology*, 16(5), 1–13. https://doi.org/10.34218/IJEET_16_05_001
8. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*.
9. Elubeyd, H., & Yiltas-Kaplan, D. (2023). Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks. *Applied Sciences*, 13(6), Article 3828. <https://doi.org/10.3390/app13063828>
10. Khan, M. A., & Kim, Y. (2021). Deep learning-based hybrid intelligent intrusion detection system. *Computers, Materials & Continua*, 68(1), 671–687. <https://doi.org/10.32604/cmc.2021.015647>
11. S. Gajula and M. Margam, "A Secure and Scalable Cloud-Based Banking Service Model Leveraging AI and Advanced Cyber Security," 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 2026, pp. 1-5, doi: 10.1109/ICAIC67076.2026.11395704.
12. Zainudin, A., et al. (2022). An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks. *IEEE Internet of Things Journal*, 10(10), 8491–8504. <https://doi.org/10.1109/JIOT.2022.3196942>
13. S. K. Ramareddy, "Integrating AI-Driven Data Visualization and Data Warehouse Analytics for Predictive Decision-Making in Financial and Stock Market Systems," 2026 International Symposium of Systems, Advanced Technologies and Knowledge (ISSATK), Hammamet, Tunisia, 2026, pp. 1-6, doi: 10.1109/ISSATK69667.2026.11566934.