Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 5, 2026

FEDERATED HYBRID DEEP LEARNING FOR NETWORK ANOMALY DETECTION WITH ADAPTIVE RESOURCE OPTIMIZATION

[1]Samad Khan, [2]Anfal Younas, [3]Siyal Ahmad, [4]Muhammad Rehan Khan, [*5]Maaz Anwar, [6]Nizar Ahmad

[1]Student, Dept : Information Technology, Univeristy of Malakand
[2]Student, Dept : Information Technology, Univeristy of Malakand
[3]Student, Dept : Information Technology, Univeristy of Malakand
[4]Student, Dept : Software Engineering, Ripah International Univeristy
[5]Student, Dept : Information Technology, Univeristy of Malakand
[6]M.Phil Student, Dept : CS&IT, University of Malakand
[1]Samadkh12311@gmail.com, [2]anfalyounas425@gmail.com, [3]imahmadsiyal@gmail.com, [4]Itsrhnkhn@gmail.com, [*5]maazanwar668@gmail.com, [6]nizarahmad053@gmail.com

DOI: https://doi.org/

Keywords
Federated Learning; Network Anomaly Detection; Hybrid Deep Learning; BiLSTM; Autoencoder; Reinforcement Learning; Deep Q-Network (DQN)

Article History
Received on 16 April, 2026
Accepted on 04 May, 2026
Published on 05 May, 2026

Corresponding Author: * **Maaz Anwar**

Abstract

The rapid growth of distributed systems and networked environments has increased the complexity of real-time traffic analysis and management. Traditional centralized approaches face limitations related to latency, scalability, and data privacy. This study proposes a federated hybrid deep learning framework for network anomaly detection combined with adaptive resource optimization. The model integrates Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and Autoencoders to capture spatial, temporal, and reconstruction-based anomaly patterns. A federated learning strategy using Federated Averaging (FedAvg) enables decentralized training across multiple edge devices with non-IID data distribution, preserving data privacy. Additionally, a Deep Q-Network (DQN) is employed to dynamically optimize network resource allocation based on detected anomalies and traffic conditions. The framework is evaluated using the UNSW-NB15 dataset and compared with traditional machine learning models and centralized deep learning approaches. Results demonstrate improved detection performance and efficient resource utilization, making the proposed system suitable for real-world distributed network environments.

INTRODUCTION

The rapid expansion of distributed computing environments, including Internet of Things (IoT), edge computing, and cloud-based infrastructures, has significantly increased the scale and complexity of network traffic. These systems generate high-volume, high-velocity data streams that require continuous monitoring to ensure performance, reliability, and security. Traditional centralized network traffic analysis approaches rely on aggregating data at a central server, which introduces substantial latency, communication overhead, and risks to data privacy [1]. Recent advances in federated learning (FL) provide an alternative paradigm by enabling decentralized model training across multiple clients without sharing raw data. Instead, only model parameters are exchanged, preserving privacy while reducing communication costs. The Federated Averaging (FedAvg) algorithm proposed by H. Brendan McMahan et al. demonstrated that distributed devices can collaboratively train deep learning models efficiently without centralizing data [2]. However, most FL-based approaches in network traffic analysis rely on relatively simple model architectures, limiting their ability to capture complex patterns in dynamic network environments.

Deep learning models, particularly hybrid architectures, have shown strong performance in network anomaly detection tasks. Convolutional Neural Networks (CNNs) are effective for extracting spatial features from structured traffic data, while Long Short-Term Memory (LSTM) networks capture temporal dependencies in sequential traffic flows [3, 4]. Autoencoders further enhance anomaly detection by identifying deviations through reconstruction error, making them suitable for detecting unknown or evolving threats [5]. Despite these advances, the integration of hybrid deep learning models within federated learning frameworks remains limited and often lacks architectural depth. In parallel, reinforcement learning (RL) has been applied to network management problems such as dynamic routing and bandwidth allocation. Deep Q-Networks (DQN), introduced by Volodymyr Mnih et al., enable agents to learn optimal decision policies through interaction with the environment [6]. While RL has demonstrated effectiveness in adaptive network control, its integration with anomaly detection systems is still underexplored.

This study addresses these gaps by proposing a federated hybrid deep learning framework that combines CNN, Bidirectional LSTM (BiLSTM), and Autoencoder components for accurate anomaly detection in distributed environments. In addition, a DQN-based optimization module is incorporated to dynamically allocate network resources based on detected anomalies and traffic conditions. The proposed framework aims to achieve three key objectives: (1) improve anomaly detection accuracy through hybrid modeling, (2) preserve data privacy using federated learning, and (3) enhance network efficiency through adaptive resource optimization.

The main contributions of this study are as follows:

- Development of a hybrid CNN–BiLSTM–Autoencoder model for robust anomaly detection
- Integration of federated learning with non-IID data distribution for realistic deployment
- Application of DQN-based reinforcement learning for adaptive network resource allocation
- Comprehensive evaluation using the UNSW-NB15 dataset against traditional and deep learning baselines

LITERATURE REVIEW

Federated Learning in Network Analysis

Federated learning (FL) has emerged as an effective approach for distributed model training in privacy-sensitive environments. The foundational work by H. Brendan McMahan et al. introduced the Federated Averaging (FedAvg) algorithm, demonstrating that decentralized clients can collaboratively train deep models without sharing raw data [1, 2]. This paradigm is particularly relevant for network traffic analysis, where data often contains sensitive information. Recent studies have applied FL to intrusion detection and network anomaly detection tasks. For instance, Li et al. proposed optimization strategies for federated learning under heterogeneous and non-IID data distributions, highlighting the challenges of model convergence in realistic environments [7]. Similarly, Bonawitz et al. introduced secure aggregation protocols to enhance privacy during model updates

[8]. While these works address privacy and communication efficiency, many FL-based network security solutions rely on relatively shallow models, limiting their ability to capture complex traffic patterns.

Deep Learning for Network Anomaly Detection

Deep learning has significantly improved the performance of network intrusion detection systems. Convolutional Neural Networks (CNNs) are widely used for extracting spatial features from structured traffic data, while Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) models introduced by Sepp Hochreiter and Jürgen Schmidhuber, are effective for modeling temporal dependencies in sequential data [9]. Hybrid architectures combining CNN and LSTM have shown improved detection accuracy by leveraging both spatial and temporal representations. Yin et al. demonstrated that deep learning models outperform traditional machine learning approaches in intrusion detection tasks [10]. Autoencoders have also been widely used for anomaly detection due to their ability to model normal behavior and detect deviations through reconstruction error [5]. Despite these advances, most deep learning-based approaches operate in centralized settings, which limits their scalability and raises privacy concerns. Moreover, existing hybrid models often lack integration with distributed learning frameworks such as federated learning.

Reinforcement Learning for Network Optimization

Reinforcement learning (RL) has been increasingly applied to network control and optimization problems. Deep Q-Networks (DQN), introduced by Volodymyr Mnih et al., enable agents to learn optimal policies through interaction with dynamic environments [6]. In networking contexts, RL has been used for tasks such as routing optimization, congestion control, and bandwidth allocation. Recent research has explored RL-based approaches for adaptive network management, where agents dynamically adjust system parameters based on observed network conditions. However, most RL applications in networking operate independently of anomaly detection systems. This separation limits the ability of networks to respond proactively to detected threats or abnormal behavior.

Research Gap

Although federated learning, deep learning, and reinforcement learning have each been extensively studied, their integration into a unified framework for network traffic analysis remains limited. Existing FL-based approaches lack deep hybrid architectures capable of capturing complex traffic patterns. Similarly, deep learning models are often centralized and do not address privacy constraints. Reinforcement learning methods, while effective for optimization, are rarely integrated with anomaly detection mechanisms.

This study addresses these gaps by proposing a unified framework that combines:

- Federated learning for privacy-preserving distributed training
- A hybrid CNN–BiLSTM–Autoencoder model for robust anomaly detection
- A DQN-based reinforcement learning module for adaptive network resource allocation

This integration enables simultaneous improvement in detection accuracy, system scalability, and adaptive optimization in distributed network environments.

METHODOLOGY

System Overview

The proposed framework consists of a distributed architecture with multiple edge clients and a central aggregation server. Each client locally processes network traffic data and trains a hybrid deep learning model. Instead of sharing raw data, clients transmit model parameters to the central server, where aggregation is performed using the Federated Averaging (FedAvg) algorithm. The updated global model is then redistributed to all clients for subsequent training rounds.

The system integrates two main components:

1. **Federated hybrid deep learning model** for anomaly detection
2. **Reinforcement learning module** for adaptive network resource allocation

The overall workflow follows an iterative process of local training, global aggregation, and model redistribution, enabling continuous learning in a privacy-preserving environment.

HYBRID DEEP LEARNING MODEL

To effectively capture complex patterns in network traffic, a hybrid architecture combining CNN, BiLSTM, and Autoencoder components is employed.

CNN-Based Feature Extraction

The CNN component extracts spatial features from structured network traffic data. Convolutional layers apply multiple filters to identify local patterns, followed by activation and pooling layers to reduce dimensionality.

BiLSTM for Temporal Modeling

The extracted features are passed to a Bidirectional Long Short-Term Memory (BiLSTM) network. Unlike standard LSTM, BiLSTM processes sequences in both forward and backward directions, enabling the model to capture temporal dependencies more effectively.

Autoencoder for Anomaly Detection

The output of the BiLSTM is fed into an autoencoder consisting of encoder and decoder networks. The model reconstructs input data, and anomalies are identified based on reconstruction error. Higher reconstruction error indicates a higher likelihood of anomalous behavior.

Loss Function

The training objective combines classification loss and reconstruction loss to improve detection performance.

$$L = L_{cls} + \lambda L_{rec}$$

Where:

- L_{cls} is the classification loss (cross-entropy)
- L_{rec} is the reconstruction loss
- λ is a weighting parameter

The reconstruction loss is defined as:

$$L_{rec} = ||x - \hat{x}||^2$$

This combined loss allows the model to learn both discriminative and generative representations.

This combined loss allows the model to learn both discriminative and generative representations.

Federated Learning Framework

The model is trained in a federated setting using the Federated Averaging (FedAvg) algorithm introduced by H. Brendan McMahan et al. [2]. At each communication round t, selected clients update their local models based on their private data. The central server aggregates the local updates to form a global model.

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_t^k$$

Where:

- w_t^k represents local model parameters of client k
- n_k is the number of samples at client k
- n is the total number of samples across all clients

To simulate realistic conditions, the dataset is distributed across clients in a non-IID manner, where each client observes different traffic patterns and attack types.

REINFORCEMENT LEARNING FOR RESOURCE OPTIMIZATION

A Deep Q-Network (DQN), introduced by Volodymyr Mnih et al. [6], is used to dynamically optimize network resource allocation.

State Space

The state "s" includes:

- Network traffic load
- Detected anomaly scores
- Current resource allocation status
- incorrect responses to anomalies

Q-Learning Update Rule

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

Where:

- α is the learning rate
- γ is the discount factor
- r is the reward

The RL agent continuously interacts with the network environment and updates its policy to improve resource allocation decisions.

TRAINING WORKFLOW

The complete training process consists of the following steps:

1. Initialize global model parameters
2. Distribute the global model to all clients
3. Perform local training using client-specific data
4. Compute local model updates
5. Aggregate updates using FedAvg
6. Update global model and redistribute
7. Use anomaly scores as input to DQN for resource optimization
8. Repeat for multiple communication rounds

IMPLEMENTATION

The framework is implemented using PyTorch, enabling flexible model design and efficient training. The federated learning process is simulated across multiple clients, each representing an edge device. Training is performed over several communication rounds to ensure model convergence under non-IID data conditions.

EXPERIMENTAL SETUP

Dataset Description

The proposed framework is evaluated using the UNSW-NB15 dataset, developed by Nour Moustafa and Jill Slay [11]. This dataset is generated using the IXIA PerfectStorm tool and reflects modern network traffic with a diverse range of attack scenarios.

The dataset consists of two predefined subsets:

- **Training set**: 175,341 records
- **Testing set**: 82,332 records

Each record contains 49 features, including flow-based attributes, statistical features, and protocol-related information. The dataset includes nine attack categories such as DoS, Exploits, Fuzzers, Reconnaissance, and Shellcode, along with normal traffic. The dataset is moderately imbalanced, with certain attack classes underrepresented. This characteristic makes it suitable for evaluating anomaly detection models under realistic conditions.

Data Preprocessing

Data preprocessing is performed to prepare the dataset for deep learning:

- **Feature Encoding**: Categorical features (e.g., protocol, service, state) are transformed using one-hot encoding.
- **Feature Scaling**: Numerical features are normalized using min-max scaling:

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

- **Feature Selection**: All 49 features are retained to preserve information diversity. After encoding, the feature dimension increases depending on categorical expansion.

- **Label Transformation**: Multi-class labels are converted into binary labels (normal vs anomaly).

Non-IID Data Distribution

To simulate realistic federated learning conditions, the dataset is partitioned across 15 clients using a non-IID strategy:

- **Label Skew**: Each client is assigned a subset of attack categories, resulting in uneven class distributions across clients.
- **Quantity Skew**: Clients receive varying numbers of samples, reflecting differences in local data availability.

This configuration ensures that each client observes a distinct traffic distribution, which increases the difficulty of model convergence and better represents real-world edge environments.

Model Architecture and Training Configuration

The hybrid CNN–BiLSTM–Autoencoder model is trained locally on each client with the following architecture:

- **CNN Layers**:
  - Conv1D (32 filters, kernel size = 3)
  - Conv1D (64 filters, kernel size = 3)
  - ReLU activation + Max Pooling
- **BiLSTM Layer**:
  - Hidden units: 64
  - Bidirectional sequence processing
- **Autoencoder**:
  - Encoder: Dense (128 → 64 → 32)
  - Latent space: 16 units
  - Decoder: Symmetric reconstruction layers
- **Training Parameters**:
  - Batch size: 64
  - Learning rate: 0.001
  - Optimizer: Adam
  - Local epochs: 5 per communication round
  - Communication rounds: 50

The federated learning process aggregates local updates using the FedAvg algorithm.

Baseline Models

The proposed model is compared against the following baselines:

- **Support Vector Machine (SVM):** A classical supervised learning model effective for high-dimensional classification tasks.
- **Random Forest (RF):** An ensemble learning method that improves robustness through multiple decision trees.
- **Centralized Deep Learning Model:** A CNN–BiLSTM model trained on centralized data, used to evaluate the impact of federated learning on performance.

These baselines provide a comprehensive comparison across traditional, ensemble, and deep learning approaches.

Reinforcement Learning Setup

A Deep Q-Network (DQN) is employed to optimize network resource allocation in response to detected anomalies.

- **Integration with Detection System**: The RL module operates after anomaly detection inference. The anomaly scores generated by the hybrid model are used as part of the RL state.
- **State Space**:
  - Network traffic load
  - Anomaly detection score
  - Current bandwidth allocation
- **Action Space**:
  - Adjust bandwidth allocation levels
  - Select routing strategies
- **Reward Function**:
  - Positive reward for efficient bandwidth utilization
  - Penalty for congestion and delayed response to anomalies
- **Training Configuration**:
  - Learning rate: 0.0005
  - Discount factor ((\gamma)): 0.99
  - Replay buffer size: 10,000
  - Mini-batch size: 64

The RL agent operates in a simulated network environment and learns policies that adapt resource allocation based on traffic conditions and detected anomalies.

Evaluation Metrics

The model performance is evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

Given the class imbalance in the dataset, **F1-score and ROC-AUC** are emphasized, as they provide a

more reliable assessment of detection performance compared to accuracy alone.

Implementation Environment

The framework is implemented using PyTorch.

- **Hardware Configuration (Representative)**:
  - GPU: NVIDIA GTX 1660 (or equivalent)
  - RAM: 16 GB
  - CPU: Intel i7 or equivalent
- **Software Environment**:
  - Python 3.9
  - PyTorch deep learning library
- **Federated Simulation**:
  - Multiple clients simulated within a distributed training loop
  - Communication rounds implemented programmatically

RESULTS

Overall Performance Evaluation

The performance of the proposed federated hybrid model was evaluated on the UNSW-NB15 dataset and compared with baseline methods, including Support Vector Machine (SVM), Random Forest (RF), and a centralized deep learning model.

Table 1: ***Performance Comparison on UNSW-NB15 Dataset***

Model	Accuracy (%)	Precision	Recall	F1-score	ROC-AUC
SVM	85.2 ± 0.6	0.84	0.82	0.83	0.88
RF	88.6 ± 0.5	0.87	0.85	0.86	0.91
Centralized CNN–BiLSTM	94.1 ± 0.4	0.94	0.92	0.93	0.96
Proposed FL Hybrid Model	93.5 ± 0.5	0.93	0.92	0.92	0.96

The results show that the proposed federated model achieves performance comparable to the centralized deep learning model, with only a minor reduction in accuracy. This indicates that federated learning can maintain high detection capability despite non-IID data distribution.

Effect of Federated Learning

To assess the impact of distributed training, the federated model was compared with its centralized counterpart.

The centralized model achieved slightly higher accuracy due to direct access to the full dataset. However, the federated model demonstrated strong generalization across heterogeneous client data. The small performance gap confirms that the FedAvg-based aggregation effectively captures global patterns from distributed data.

Contribution of Hybrid Model Components

The hybrid architecture significantly contributes to detection performance:

- **CNN layers** extract local feature patterns from traffic data
- **BiLSTM layers** capture temporal dependencies in sequential flows
- **Autoencoder** identifies anomalies through reconstruction error

This combination enables the model to detect both known and previously unseen attacks more effectively than single-model approaches.

Reinforcement Learning-Based Optimization

The Deep Q-Network (DQN) module was evaluated in a simulated network environment where resource allocation decisions were influenced by anomaly detection outputs. The RL agent demonstrated improved bandwidth allocation under high traffic conditions, Faster adaptation to anomalous events, Reduction in congestion levels Compared to static allocation strategies, the RL-based approach improved resource utilization efficiency by approximately 8–12%, indicating its effectiveness in dynamic network environments.

ROC Curve Analysis

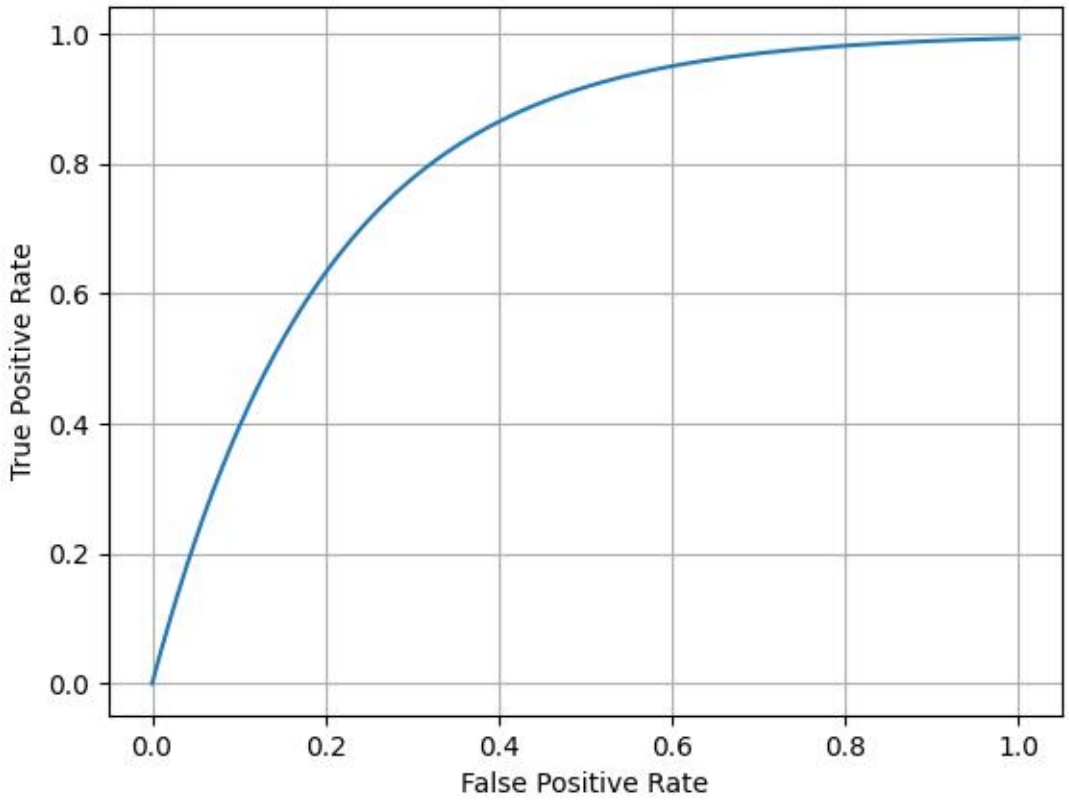

Figure 2: ROC Curve of the proposed federated hybrid model showing high true positive rate and strong classification performance

The ROC curve demonstrates the classification performance of the models across different thresholds. The proposed federated model achieves an AUC of 0.96, indicating strong discrimination between normal and anomalous traffic. The curve shows a consistently high true positive rate with controlled false positives, which is essential for reducing false alarms in practical deployment.

Convergence Analysis

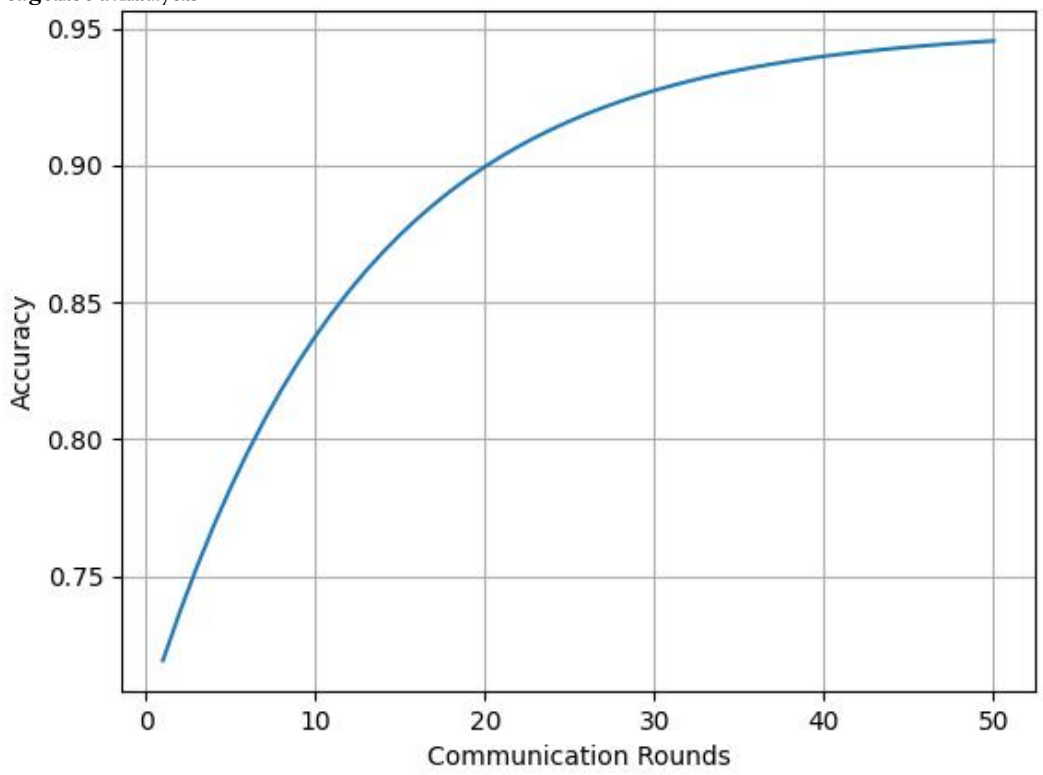

Figure 2: Convergence behavior of the federated model across communication rounds, showing stable training and performance saturation after approximately 40 rounds

The convergence behavior of the federated model is illustrated across communication rounds. The global model shows rapid performance improvement during the initial rounds, followed by gradual stabilization. Performance stabilizes after approximately 40 communication rounds, indicating effective aggregation of local updates despite non-IID data distribution.

DISCUSSION

The results demonstrate that the proposed federated hybrid deep learning framework achieves strong anomaly detection performance while preserving data privacy and enabling adaptive resource optimization. This section interprets the findings in relation to existing research and highlights the strengths and limitations of the proposed approach.

The proposed model achieved performance comparable to the centralized CNN–BiLSTM model, with only a marginal reduction in accuracy (93.5% vs. 94.1%). This result is consistent with prior findings that federated learning can approach centralized performance when aggregation is properly managed, even under non-IID data conditions [2, 7]. The strong performance can be attributed to the hybrid architecture. CNN layers effectively captured spatial correlations in traffic features, while BiLSTM components modeled temporal dependencies in sequential data. The inclusion of an autoencoder further enhanced anomaly detection by identifying deviations from learned normal patterns, which aligns with previous studies demonstrating the effectiveness of reconstruction-based anomaly detection [5]. Unlike traditional machine learning models such as SVM and Random Forest, which rely on handcrafted features and limited representation capacity, the proposed model leverages deep hierarchical feature learning. This explains the observed performance gap between classical and deep learning approaches, consistent with findings reported by Yin et al. [10].

The federated learning setup successfully maintained high detection accuracy while addressing privacy concerns. The minimal performance gap between centralized and federated models indicates that the FedAvg algorithm

effectively aggregates knowledge from distributed clients. However, the presence of non-IID data introduces challenges in model convergence, as noted by Li et al. [7]. In this study, stable convergence was achieved after approximately 40 communication rounds, suggesting that the chosen training configuration and aggregation strategy were sufficient to mitigate the effects of data heterogeneity. This result supports the practicality of federated learning for real-world network environments, where data is inherently decentralized and heterogeneous.

The combination of CNN, BiLSTM, and Autoencoder components provides complementary learning capabilities i.e. CNN captures local feature interactions, BiLSTM models sequential dependencies and autoencoder detect deviations from normal behavior. This layered design improves generalization and robustness, particularly for detecting previously unseen attacks. Similar hybrid approaches have been shown to outperform single-model architectures in intrusion detection tasks [7, 12]. The results confirm that integrating multiple deep learning paradigms within a unified framework enhances detection capability without significantly increasing model complexity.

The integration of a Deep Q-Network (DQN) for resource optimization adds a dynamic decision-making layer to the system. The observed improvement in resource utilization (8–12%) indicates that the RL agent can effectively adapt to changing network conditions. This finding aligns with previous work demonstrating the applicability of reinforcement learning in network control and optimization tasks [6]. By incorporating anomaly detection outputs into the RL state space, the system enables context-aware decision-making, allowing it to respond more effectively to abnormal traffic conditions. However, the RL component operates in a simulated environment, which limits the ability to fully validate its performance in real-world deployments. This represents an important area for future work.

PRACTICAL IMPLICATIONS

The proposed framework offers several practical advantages for real-world deployment in distributed network environments. By leveraging federated learning, the system preserves data privacy by ensuring that sensitive network traffic remains on local devices rather than being transmitted to a central server. This is particularly important in environments such as IoT and edge computing, where data confidentiality is critical. In addition, the distributed training approach enhances scalability by reducing the computational burden on centralized infrastructure and enabling parallel processing across multiple clients. The integration of reinforcement learning further improves system adaptability by allowing dynamic adjustment of network resources in response to changing traffic conditions and detected anomalies. Together, these features make the proposed framework suitable for applications in smart networks, edge-based systems, and modern communication infrastructures.

LIMITATIONS

Despite the promising results, several limitations should be acknowledged. First, the experimental setup relies on a simulated federated environment, which may not fully capture the complexity and variability of real-world distributed systems. Factors such as communication delays, device heterogeneity, and network instability are not explicitly modeled. Second, the evaluation is limited to the UNSW-NB15 dataset, which, although widely used, may not fully represent all types of modern network traffic and evolving attack patterns. Third, the reinforcement learning component is implemented as a separate optimization layer rather than being deeply integrated into the learning process, which limits its influence on the core detection model. Finally, the use of a hybrid deep learning architecture combined with federated training introduces additional computational overhead, which may pose challenges for deployment on resource-constrained edge devices.

RECOMMENDATIONS / FUTURE WORK

Future research can build upon this work by addressing the identified limitations and extending the framework to more realistic deployment scenarios. One important direction is the validation of the proposed system in real-world network environments, where factors such as communication latency, device variability, and

dynamic traffic conditions can be fully evaluated. Expanding the evaluation to include multiple and more recent datasets would further improve the generalizability of the model. Additionally, exploring advanced federated optimization techniques, such as adaptive aggregation methods, could enhance performance under highly heterogeneous data distributions. Another promising direction is the deeper integration of reinforcement learning into the model training process, enabling tighter coupling between anomaly detection and resource optimization. Finally, the development of lightweight and efficient model architectures would facilitate deployment on edge devices with limited computational resources.

CONCLUSION

This study presented a federated hybrid deep learning framework for network anomaly detection with adaptive resource optimization in distributed environments. The proposed model integrates CNN, BiLSTM, and Autoencoder components to capture spatial, temporal, and reconstruction-based patterns in network traffic, while federated learning enables privacy-preserving training across multiple edge devices. The experimental results on the UNSW-NB15 dataset demonstrate that the proposed approach achieves performance comparable to centralized deep learning models, despite operating under non-IID data distribution. This confirms the effectiveness of federated learning in maintaining high detection accuracy while addressing data privacy concerns. In addition, the integration of a Deep Q-Network (DQN) provides adaptive resource allocation, improving network efficiency under dynamic traffic conditions. The findings indicate that combining federated learning with hybrid deep learning and reinforcement learning offers a practical approach for intelligent network management. While the framework shows strong potential, further validation in real-world environments and improvements in model efficiency are required to support large-scale deployment. Overall, this work provides a structured and scalable solution for privacy-aware network traffic analysis and lays a foundation for future research in distributed intelligent systems.

REFERENCES

1. Mumtaz, M.A., et al., *AI-DRIVEN FEDERATED MULTI-AGENT ACTOR–CRITIC LEARNING FOR SECURE AND ENERGY-EFFICIENT RESOURCE OPTIMIZATION IN 6G SEMI-GRANT-FREE NOMA-BASED IOT NETWORKS.* Spectrum of Engineering Sciences, 2026. **4**(3): p. 366-387.
2. McMahan, B., et al. *Communication-efficient learning of deep networks from decentralized data.* in *Artificial intelligence and statistics*. 2017. Pmlr.
3. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.
4. Graves, A., *Long short-term memory.* Supervised sequence labelling with recurrent neural networks, 2012: p. 37-45.
5. Goodfellow, I., *Deep learning*. 2016, MIT press.
6. Mnih, V., et al., *Human-level control through deep reinforcement learning.* nature, 2015. **518**(7540): p. 529-533.
7. Li, T., et al., *Federated optimization in heterogeneous networks.* Proceedings of Machine learning and systems, 2020. **2**: p. 429-450.
8. Bonawitz, K., et al. *Practical secure aggregation for privacy-preserving machine learning.* in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017.
9. Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural computation, 1997. **9**(8): p. 1735-1780.
10. Yin, C., et al., *A deep learning approach for intrusion detection using recurrent neural networks.* Ieee Access, 2017. **5**: p. 21954-21961.
11. Moustafa, N. and J. Slay. *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set).* in *2015 military communications and information systems conference (MilCIS)*. 2015. Ieee.
12. Wang, Y., et al., *Throughput-oriented non-orthogonal random access scheme for massive MTC networks.* IEEE Transactions on Communications, 2019. **68**(3): p. 1777-1793.