Scientific Reports

https://doi.org/10.1038/s41598-026-37289-1

Article in Press

# A federated deep learning approach for SDN security with quantum optimized feature selection and hybrid MSDC net architecture

Received: 7 August 2025

Accepted: 21 January 2026

Published online: 10 February 2026

Cite this article as: Rohith S., Logeswari G., Tamilarasi K. *et al.* A federated deep learning approach for SDN security with quantum optimized feature selection and hybrid MSDC net architecture. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-37289-1

S. Rohith, G. Logeswari, K. Tamilarasi & G. Sudhakaran

# A Federated Deep Learning Approach for SDN Security with Quantum Optimized Feature Selection and Hybrid MSDC Net Architecture

**Rohith S[1] · Logeswari G[1,*] · Tamilarasi K[1] · Sudhakaran G[2]**
[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600 127, TamilNadu, India
[2]School of Electronics Engineering, Vellore Institute of Technology, Chennai 600 127, TamilNadu, India

Corresponding Author: Logeswari G (email id: logeswari.g@vit.ac.in)

**ABSTRACT** Software-Defined Networking (SDN) is increasingly exposed to complex cyberattacks, requiring advanced, adaptive, and efficient intrusion detection mechanisms. This study presents LightIDS-SDN, a federated and explainable intrusion detection framework tailored for SDN environments. At its core, the system employs Dual Fitness Enhanced Quantum-Inspired Particle Swarm Optimization (DFE-GQPSO) for feature selection, which identifies the most informative network attributes while eliminating redundant or irrelevant features. This quantum-optimized feature selection significantly improves detection performance by reducing overfitting and enhancing generalization. The framework incorporates a hybrid deep learning architecture, MSDC-Net, combining Transformer layers, Capsule Networks, and BiLSTM units to capture contextual, spatial, and sequential dependencies in network traffic. Federated learning using FedAvg enables collaborative model training across multiple SDN controllers while preserving data privacy. Explainable AI modules, based on SHapley Additive exPlanations (SHAP) and Gradient-weighted Class Activation Mapping (Grad-CAM), provide both global and local interpretability, ensuring transparent and accountable decision-making. Experiments on the InSDN dataset demonstrate the effectiveness of the proposed system, achieving 98.73% accuracy, 98.80% precision, 98.65% recall, and 98.72% F1-score. Comparative analysis confirms that DFE-GQPSO outperforms traditional feature selection methods, enhancing model robustness and training efficiency. Overall, LightIDS-SDN effectively detects a wide range of SDN attacks while addressing limitations of conventional IDS approaches, including limited scalability, lack of interpretability, and computational inefficiency. This work lays the foundation for deploying quantum-optimized, explainable, and federated intrusion detection systems in SDN networks.

**INDEX TERMS** – Software-Defined Networking, Intrusion Detection System, Feature Selection, Particle Swarm Optimization, Transformer, Capsule Network, BiLSTM.



## I. INTRODUCTION

SDN has emerged as one of the most significant innovations in modern network architecture, fundamentally transforming the way networks are designed, managed, and optimized [1]. Unlike conventional networks, where the control plane and data plane are tightly coupled within individual devices, SDN introduces a clean separation of these planes. The control plane, responsible for making routing and forwarding decisions, is centralized in a software-defined controller, while the data plane, consisting of forwarding devices such as switches and routers, focuses solely on packet transmission [2]. This decoupling introduces programmability into networks, allowing administrators to dynamically adjust policies, automate traffic engineering, and rapidly deploy new services without manually reconfiguring each network device [5]. The result is a flexible, scalable, and agile infrastructure capable of supporting the demands of next-generation applications.

One of the key advantages of SDN is its ability to provide centralized visibility and control over the entire network [3]. By abstracting the underlying physical infrastructure, SDN simplifies management and enables advanced functions such as load balancing, quality-of-service (QoS) enforcement, and energy-aware routing. Moreover, SDN inherently supports network virtualization, where multiple independent logical networks can coexist over the same physical infrastructure. This is particularly valuable for cloud computing, data centers, and enterprise environments, where resources need to be dynamically allocated to meet diverse and fluctuating workloads [4]. As

a result, SDN has been adopted in mission-critical domains including 5G/6G telecommunication networks, Internet of Things (IoT) ecosystems, and industrial control systems [5]. Its flexibility and programmability make it indispensable for environments requiring rapid scalability, cost efficiency, and real-time adaptability.

Despite these benefits, the architectural design of SDN introduces novel security vulnerabilities that differ fundamentally from those in traditional networks [6]. The centralized controller is both the greatest strength and the biggest weakness of SDN. While centralization allows global visibility and fine-grained control, it also creates a single point of failure. If an attacker compromises the controller, they could potentially manipulate flow rules, inject malicious policies, or disrupt network-wide services [7]. Similarly, the communication between controllers and data plane devices, typically carried out through southbound protocols such as OpenFlow, is vulnerable to spoofing, tampering, and interception if not properly secured [8]. In addition, SDN's programmability, while enabling flexibility, increases the attack surface by allowing malicious or poorly configured applications to introduce vulnerabilities into the control logic [9].

The nature of SDN traffic also adds complexity to security. Unlike traditional enterprise networks, where traffic patterns are relatively predictable, SDN environments often experience dynamic, heterogeneous, and high-volume flows generated by diverse applications and devices. Distinguishing legitimate variations in traffic from actual anomalies becomes increasingly challenging under such conditions. Furthermore, SDN environments are frequent targets of advanced attack types, including Denial-of-Service (DoS) and Distributed DoS (DDoS) attacks aimed at overwhelming the controller, man-in-the-middle (MITM) attacks targeting communication channels, probe or reconnaissance attacks designed to map network topology, and multi-stage or advanced persistent threats (APTs) that exploit vulnerabilities in applications running on top of SDN [10].

To address these challenges, Intrusion Detection Systems (IDS) play an essential role in SDN security. IDS are designed to continuously monitor network traffic, detect abnormal or malicious activities, and raise alerts for administrators to take corrective action [11]. Traditional IDS approaches typically fall into two categories: signature-based detection, which relies on predefined attack signatures, and anomaly-based detection, which identifies deviations from normal traffic patterns [12]. While signature-based IDS are effective for known attacks, they cannot detect zero-day or evolving threats. Anomaly-based IDS, on the other hand, can identify new or previously unseen attacks but often suffer from high false-positive rates.

In the context of SDN, IDS face additional demands compared to conventional networks. First, IDS must operate in real-time, handling massive amounts of traffic without introducing delays or degrading network performance [13]. Second, IDS must be adaptive to rapidly changing topologies and policies inherent to SDN environments, where traffic flows are reconfigured dynamically. Third, due to the heterogeneous nature of SDN traffic, IDS must be capable of analyzing complex and multi-dimensional feature spaces to distinguish between benign and malicious behaviors. These requirements exceed the capabilities of traditional IDS, highlighting the need for more intelligent and adaptive solutions.

In recent years, the integration of machine learning (ML) and DL into IDS has gained significant attention. ML/DL models can learn patterns from traffic data, generalize across different attack types, and adapt to new threats without constant manual intervention [14]. For instance, techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have been used to capture spatial and temporal patterns in traffic data. However, these approaches also face several limitations when applied to SDN. CNNs are effective at extracting local spatial features but fail to capture long-range dependencies in traffic sequences. LSTMs can model sequential relationships but struggle with hierarchical spatial structures. Moreover, many DL-based IDS models require large numbers of features, leading to computational inefficiency, and often act as black boxes, offering little transparency in their decision-making processes [15][16].

Another emerging challenge is the scalability and privacy of IDS deployment in distributed

SDN environments. Most existing IDS frameworks rely on centralized training, where raw traffic data from different domains is aggregated in a single location [17]. This approach raises serious privacy concerns and is impractical in large-scale multi-domain SDN deployments, where data sharing is restricted. Therefore, there is growing interest in incorporating federated learning into IDS, which allows models to be collaboratively trained across multiple domains without sharing raw data, thereby preserving privacy while improving scalability [18].

In summary, while SDN brings immense benefits in terms of programmability, agility, and cost efficiency, it also introduces unique vulnerabilities that cannot be adequately addressed by traditional security mechanisms. IDS designed for SDN must balance accuracy, efficiency, adaptability, scalability, and interpretability — a balance that current approaches struggle to achieve. These challenges set the stage for the development of novel solutions that integrate feature optimization, hybrid DL, explainable AI (XAI), and FL to build next-generation IDS frameworks for SDN security.

### *A.* MOTIVATION

The rapid evolution of cyber threats and the dynamic nature of SDN require IDS frameworks that go beyond conventional detection methods. Traditional IDS approaches either fail to detect novel and zero-day attacks or produce excessive false alarms, limiting their practicality. Moreover, centralized data collection compromises privacy and increases the risk of single-point failures. The motivation behind this study is to design an IDS that is not only lightweight and computationally efficient but also scalable, privacy-preserving, and capable of detecting attack patterns with high accuracy and interpretability.

### B. KEY CONTRIBUTIONS

The main contributions of this work are summarized as follows:

- We propose a novel DFE-GQPSO algorithm for feature selection, jointly optimizing detection accuracy and computational efficiency to ensure suitability for real-time SDN environments.
- We design MSDC-Net, a hybrid DL architecture that integrates Transformer, Capsule Network, and BiLSTM modules for comprehensive spatial, contextual, and sequential analysis of network traffic
- We demonstrate that LightIDS-SDN significantly outperforms state-of-the-art IDS approaches on the InSDN dataset, achieving higher accuracy, precision, recall, and F1-score.
- We incorporate explainable AI (XAI) mechanisms, using SHAP and Grad-CAM, to provide interpretable outputs and enhance trust in automated intrusion detection systems.
- We achieve substantial feature reduction through DFE-GQPSO, reducing computational overhead while enabling scalable and efficient deployment in resource-constrained SDN controllers

The remainder of this paper is organized as follows: Section 2 reviews related work on SDN security, feature selection methods, and machine learning approaches to intrusion detection. Section 3 presents the architecture of the proposed LightIDS-SDN framework, including the optimization and classification modules. Section 4 details the experimental setup and evaluation metrics. Section 5 discusses the results and interpretability features. Section 6 concludes the study and outlines future research directions.

## II. RELATED WORK

The escalating sophistication of cyber threats targeting network infrastructures necessitates a paradigm shift from reactive, signature-based security measures to proactive, intelligent, and adaptive defense systems. SDN, with its core tenet of decoupling the control plane from the data plane, offers a fertile ground for implementing such intelligent security solutions. However, this very architecture introduces novel attack surfaces. This survey synthesizes the critical research pillars underpinning the proposed work: security challenges in SDN, the evolution of DL for network intrusion detection, the imperative of data privacy via FL and other techniques, the challenge of data heterogeneity, and the emerging potential of Quantum Computing (QC) for optimization, culminating in the identification of a clear research gap.

### A. SECURITY CHALLENGES IN SOFTWARE-DEFINED NETWORKING

SDN's centralized intelligence, embodied in the logically centralized controller, is both its greatest strength and its most significant vulnerability. Traditional network attacks are compounded by new SDN-specific threats. Kreutz et al. [19], in their seminal paper, provided one of the first systematic analyses of SDN security. They meticulously catalogued threat vectors, highlighting the southbound interface as a critical target. A DDoS attack can overwhelm the control channel, saturating the link between switches and the controller, or exhaust the controller's computational resources, creating a critical single point of failure. Scott-Hayward et al. [20] expanded on this in their survey, emphasizing the programmability paradox: while enabling innovation, it also introduces significant risks if not properly secured. Their work underscores the need for an IDS that is as dynamic and programmable as the SDN itself. More recently, Ahmed et al. [21] provided an updated taxonomy of DDoS attacks specific to SDN and critiqued existing defense mechanisms, concluding that machine learning-based solutions integrated within the controller are the most promising avenue for future research, albeit with data privacy concerns.

### B. DEEP LEARNING FOR NETWORK INTRUSION DETECTION SYSTEMS

The application of DL for NIDS has been extensively explored due to its superior capability in handling high-dimensional, non-linear data. Recent work has focused on creating lightweight and efficient architectures for resource-constrained environments like IoT and CPS. For instance, Saheed et al. [22] proposed a novel transfer learning approach that creates a lightweight model by combining a 1D-CNN with a simplified ResNet50 backbone, demonstrating improved intrusion detection for cyber-physical systems. Similarly, the exploration of hybrid models continues to be a dominant trend. Saheed et al. [23] presented an autoencoder via DCNN and LSTM models for intrusion detection in industrial control systems, highlighting the effectiveness of deep convolutional and recurrent layers for feature extraction and temporal modeling in critical infrastructures.

Early approaches leveraged Multi-Layer Perceptrons (MLPs). The work of Ingre et al. [24] demonstrated that a well-tuned ANN could achieve higher classification accuracy than traditional ML models on the benchmark NSL-KDD dataset. However, MLPs ignore crucial sequential and spatial relationships. To capture temporal patterns, Recurrent Neural Networks (RNNs) have been adopted. Laghrissi et al. [25] showed that LSTM networks are exceptionally good at detecting attacks that unfold over time. Kang et al. [26] further validated this for CAN bus networks, demonstrating the architecture's versatility across different network types. They successfully detected message injection attacks by learning the normal temporal sequence of CAN messages. Saheed et al. [27] also focused on optimization, presenting a modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities, which aligns with the feature selection focus of this work.

Concurrently, CNNs have been applied to NIDS. Wang et al. [28] proposed a multi-scale CNN that used kernels of different sizes to extract features at various granularities. Toldinas et al. [29] took this a step further by converting network traffic into 2D grayscale images. Their model applied image recognition techniques to achieve high accuracy on the CIC-IDS2017 dataset, proving the efficacy of treating traffic as a spatial problem. The natural progression was to combine these strengths. Yin et al. [30] developed a hybrid CNN-LSTM model that fused spatial and temporal feature extraction. Fu et al. [31] addressed a critical practical issue: class imbalance. They integrated a Generative Adversarial Network (GAN) to synthesize minority class samples (e.g., rare attacks) before feeding the data into their hybrid CNN-LSTM, significantly improving the detection rate for these elusive attacks. The trend towards ensemble and hybrid methods is further evidenced by Saheed et al. [32], who proposed a novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems, showcasing the power of combining multiple models for robust performance in smart city infrastructures. Despite their high accuracy, these centralized DL models face fundamental challenges: data privacy, bandwidth overhead, and the "curse of dimensionality."

### C. THE PARADIGM OF PRIVACY-PRESERVING AND EXPLAINABLE SECURITY

The requirement for centralizing data is a major impediment to deploying DL-based NIDS in real-world scenarios due to data privacy regulations (e.g., GDPR, CCPA) and commercial sensitivities. Yang et al. [33] formally defined FL and the foundational FedAvg algorithm. Liu et al. [34] implemented this for IoT intrusion detection, demonstrating comparable accuracy to centralized models while preserving privacy. Li et al. [35] identified key issues: (i) Statistical Heterogeneity (non-IID data): Data across clients (e.g., different companies) can have different distributions, causing the global model to diverge or perform poorly on individual clients. (ii) Communication Efficiency: Transmitting entire model updates is costly. (iii) Systems and Hardware Heterogeneity: Clients have varying computational capabilities.

Several works have sought to address non-IID data. Zhao et al. [36] famously showed that accuracy can significantly drop with non-IID data and proposed a strategy of using a globally shared, small subset of data to stabilize training. More recently, Karimireddy et al. [37] proposed Stochastic Controlled Averaging for Federated Learning, which uses control variates to correct for client drift, showing strong theoretical and empirical improvements on non-IID data. For communication efficiency, Konečný et al. [38] introduced structured and sketched update compression techniques to drastically reduce the upload communication cost.

Federated-Boosting, proposed by Khan et al. [45] leverages boosting in a federated setting to enhance cyber-attack detection in consumer IoT while preserving privacy. Khan et al. [46] introduced Fed-Inforce-Fusion, a federated reinforcement-based fusion model that secures IoMT networks by dynamically adapting to heterogeneous devices. Both works highlight distributed intelligence and robust detection without exposing sensitive user data. The Collaborative SRU Network, developed by Khan et al. [47, applies dynamic behavior aggregation to strengthen collaborative intrusion detection. It also reduces communication overhead, effectively addressing scalability issues in federated environments.

Beyond FL, other privacy techniques are emerging. Saheed et al. [39] introduced an explainable privacy-preserving DNN that uses homomorphic encryption to allow computations on encrypted data, offering a different approach to privacy than FL. Furthermore, the field is moving towards XAI. Saheed [40] also addressed this in "CPS-IIoT-P2Attention," which incorporates a scaled dot-product attention mechanism not only to improve performance but also to provide explainability by highlighting which parts of the input data contributed most to the detection decision, a crucial feature for security analysts.

### D. QUANTUM AND BIO-INSPIRED COMPUTING FOR FEATURE SELECTION

To combat the "curse of dimensionality," efficient feature selection is critical. Arora et al. [41] explored a quantum-inspired butterfly optimization algorithm for feature selection. The potential of actual quantum computing is even greater. Biamonte, et al. [42] provided the foundational theory for quantum machine learning. Practical applications are emerging. Xie et al. [43] used a quantum annealer to solve a quadratic unconstrained binary optimization (QUBO) problem formulated for feature selection, reporting a reduction in features without loss of accuracy. Similarly, Willsch et al. [44] explored the Quantum Approximate Optimization Algorithm (QAOA) for combinatorial problems, laying groundwork for its use in tasks like feature selection, though noting current hardware limitations.

### E. RESEARCH GAP AND NOVEL CONTRIBUTION

The surveyed literature reveals clear, distinct progress in individual domains: robust DL architectures, privacy-preserving techniques (both FL and encryption-based), explainable AI, and advanced optimizers for feature selection. However, a significant gap exists in the holistic integration of these cutting-edge technologies into a unified, efficient, and scalable solution for SDN security. Existing hybrid models are centralized and privacy-invasive or use encryption that adds high computational overhead . FL implementations for NIDS often use simple models and do not address the computational burden of high-dimensional data on resource-constrained clients.

The critical challenge of non-IID data in a federated SDN context is largely unaddressed. Powerful quantum-optimized feature selection techniques are siloed and not applied within a federated context to streamline the local learning process and combat non-IID drift. Explainability, a critical need for security operations, is often an afterthought and not integrated into federated IDS designs [26]. This work aims to bridge this gap by proposing a novel federated deep learning framework that incorporates a Quantum-Optimized Feature Selection process to efficiently reduce data dimensionality on each client, directly addressing communication and computational costs and mitigating non-IID effects by aligning on a optimal feature set. Furthermore, it introduces a bespoke Hybrid MSDC Net architecture, designed to more effectively extract both spatial features and temporal dependencies from the selected features within the federated learning paradigm. By designing the model with inherent interpretability, inspired by attention mechanisms, this work also incorporates explainability into its core design. This integrated approach promises enhanced detection accuracy, rigorous data privacy, reduced communication overhead, greater computational efficiency, and actionable insights for security analysts. To provide a comparative analysis, we summarize various existing approaches in Table I.

**Table 1: Comprehensive summary of existing approaches**

| Authors & Year | Key Contribution | Methodology / Approach | Dataset(s) Used | Key Limitation / Challenge | Relevance to Proposed Work |
|---|---|---|---|---|---|
| Kreutz et al. [19] | Seminal threat analysis of SDN architecture. | Systematic security analysis. | - | No ML-based mitigation strategies. | Defines the SDN security problem space. |
| Scott-Hayward et al. [20] | Survey on SDN security challenges. | Survey and taxonomy. | - | Highlights problems, no new solutions. | Reinforces need for dynamic IDS. |
| Ingre et al. [24] | Demonstrated ANN's superiority over SVM for IDS. | Artificial Neural Network (ANN). | NSL-KDD | Ignores spatiotemporal features. | Highlights need for advanced architectures. |
| Laghrissi et al. [25] | Applied LSTM to model temporal sequences for IDS. | Long Short-Term Memory (LSTM). | Custom data | Univariate focus. | Validates LSTM for temporal features. |
| Wang et al. [28] | Pioneered multi-scale CNNs for spatial feature extraction. | Multi-scale CNN. | ISCX VPN-nonVPN | No temporal modeling. | Informs spatial component of MSDC Net. |
| Yin et al. [30] | Hybrid CNN-LSTM for spatiotemporal features. | Hybrid Deep Learning (CNN + LSTM). | NSL-KDD, KDD99 | Centralized model; violates privacy. | MSDC Net is an evolution for FL. |
| Yang et al. [33] | Formalized FL and FedAvg. | Conceptual framework. | - | Vanilla FedAvg poor on non-IID data. | Foundational FL paradigm. |
| Liu et al. [34] | Implemented FL for network traffic prediction. | FL with LSTM. | PeMS dataset | Simple model; task is prediction, not security. | Demonstrates FL in networking. |
| Arora et al. [41] | Quantum-inspired metaheuristic for feature selection. | Quantum-Inspired BOA. | UCI Repository | Not on quantum hardware; not applied in FL. | Basis for advanced FS optimizers. |
| Biamonte et al. [42] | Theoretical overview of QML potential. | Survey of quantum algorithms. | - | Theoretical; no practical implementations. | Justifies quantum for optimization. |
| Ahmed et al. [21] | Survey on SDN DDoS defenses, advocated for ML. | Comprehensive survey. | - | Does not address data privacy. | Confirms need for ML in SDN security. |
| Kang et al. [26] | Validated LSTM for In-Vehicle Network security. | LSTM-based IDS. | Car Hacking dataset | Domain-specific (IVN). | Strengthens case for temporal models. |

| | | | | | |
|---|---|---|---|---|---|
| Toldinas et al. [29] | Converted traffic to 2D images for CNN classification. | Image-based CNN (2D Conv). | CIC-IDS2017 | Information loss in transformation. | Supports exploiting spatial correlations. |
| Fu et al. [31] | Integrated GANs with hybrid model for imbalance. | GAN + Hybrid CNN-LSTM. | NSL-KDD, CIC-IDS2017 | Centralized; complex training. | Highlights importance of handling imbalance. |
| Li et al. [35] | Identified key challenges in FL (e.g., non-IID). | Survey and analysis. | - | Does not propose a specific novel solution. | Core challenge to overcome. |
| Zhao et al. [36] | Showed performance drop of FL on non-IID data. | Experimental analysis. | CIFAR-10, MNIST | Proposed solution violates pure FL principles. | Empirically highlights non-IID issue. |
| Karimireddy et al. [37] | Proposed SCAFFOLD algorithm for client drift. | Novel FL algorithm. | CIFAR-10, FEMNIST | Increases communication cost. | Advanced FL for non-IID data. |
| Konečný et al. [38] | Introduced compression for communication efficiency. | Model compression techniques. | MNIST, LIBSVM | Can impact model convergence. | Addresses communication bottleneck. |
| Xie et al. [43] | Used quantum annealer for FS in malware detection. | Quantum Annealing for QUBO. | Microsoft Malware | Requires quantum hardware; small-scale. | Concrete example of quantum-assisted FS. |
| Larkin et al. [44] | Explored QAOA for combinatorial optimization. | Performance evaluation of QAOA. | Synthetic problems | Current hardware limitations. | Informs choice of quantum algorithms. |
| Saheed et al. [22] | Lightweight transfer learning model for CPS IDS. | ResNet50-1D-CNN Transfer Learning. | - | Centralized model architecture. | Highlights trend towards lightweight, efficient models for edge deployment. |
| Saheed et al. [23] | Autoencoder with DCNN/LSTM for ICS IDS. | Hybrid Autoencoder (DCNN + LSTM). | Industrial Control System data | Centralized model; focus on ICS not general SDN. | Corroborates effectiveness of hybrid deep learning models. |
| Saheed et al. [27] | GA-optimized LSTM for IoT intrusion detection. | Modified GA + Fine-tuned LSTM. | IoT network data | Focus on IoT edge; GA may have high computational cost. | Parallels the focus on optimizing model components (like FS) for performance. |
| Saheed et al. [32] | Hybrid ensemble learning for anomaly detection in SCADA. | Hybrid Ensemble Learning. | Industrial sensor/SCADA data | Centralized ensemble model can be complex. | Supports the power of hybrid/model fusion approaches. |
| Saheed et al. [39] | Privacy-preserving DNN using homomorphic encryption. | Homomorphic Encryption with DNN. | CPS-IoT data | High computational overhead from encryption. | Represents an alternative, compute-heavy privacy approach vs. FL. |
| Saheed [40] | Explainable privacy-preserving model with attention. | Scaled Dot-Product Attention for XAI. | CPS-IIoT data | Centralized model (P2Attention mechanism itself is not private). | Highlights the critical need for and method of achieving explainability in IDS. |

## III. PROPOSED METHODOLOGY

The proposed framework, LightIDS-SDN, integrates a multi-stage pipeline including preprocessing, feature selection, federated learning, deep hybrid classification, and explainability. The system's architecture is illustrated in Fig. 1, showcasing the interplay between edge devices, SDN controllers, and the central aggregation server. Although Fig. 2 is inspired by the general ML–SDN integration paradigm, the proposed LightIDS-SDN framework significantly extends this architecture to ensure practical deployment viability. Unlike traditional approaches that directly feed raw traffic features into a classifier, LightIDS-SDN introduces a quantum-inspired feature selection stage (DFE-GQPSO) that optimizes SDN-specific flow attributes before detection, reducing redundancy and

improving generalization. Furthermore, the hybrid MSDC-Net model captures contextual, hierarchical, and temporal attack patterns using Transformer, Capsule Network, and BiLSTM layers. The framework is further enhanced with federated learning, enabling multiple SDN controllers to collaboratively train intrusion detection models without sharing raw traffic data, thereby preserving privacy. Finally, the Explain-Edge module provides transparent, feature-level and model-level explanations, supporting operational trust and real-time decision-making. These additions transform the conventional ML–SDN pipeline into a deployable, scalable, and explainable intrusion detection system.

### A. DATASET

The InSDN dataset serves as a cornerstone for the proposed LightIDS-SDN framework, offering a comprehensive and realistic benchmark tailored for SDN security research. It comprises 361,317 labeled records, with 292,893 representing attack traffic and 68,424 benign instances, making it suitable for both binary and multi-class classification. The dataset features nine attack categories that specifically exploit SDN components, including DoS-HULK, DoS-HTTP Flood, DoS-Torsh
Hammer, DDoS, Probing, Botnet, Exploitation, (U2R), Password Guessing, and Web Attacks. These attacks target the SDN control plane through flooding, unauthorized access, and exploitation of OpenFlow mechanisms. Generated using Mininet, with Open vSwitch and the Floodlight controller, the dataset includes traffic from Metasploitable 2 and benign background flows, all timestamped with microsecond granularity to support real-time detection. Each instance includes 83 engineered features, covering flow statistics, temporal metrics, TCP behavior, and SDN-specific attributes like packet_in and flow_mod rates. This diverse feature set enables detection of complex attack patterns across both control and data planes. Unlike traditional IDS datasets, InSDN captures SDN-specific vulnerabilities, making it highly suitable for training the proposed MSDC-Net model, which relies on context-rich and fine-grained traffic features for accurate and robust intrusion detection.

### B. DATA PRE-PROCESSING

The initial stage of the proposed system involves data preprocessing of the raw network traffic from the InSDN dataset, designed for SDN. This process ensures the data is clean, labeled, balanced, and normalized for effective modeling. The preprocessing pipeline begins with data cleaning, where records with missing or null values are removed or imputed, duplicates are eliminated, and irrelevant features are discarded to maintain data quality.

Each network flow $i$ is then assigned a multi-class label $y_i$ from the label set $\{0,1,2,\ldots,K\}$ where $y_i=0$ indicates benign traffic, and $y_i=k$ (for $1 \le k \le K$) represents one of the K attack types. To address the class imbalance problem common in intrusion datasets where benign samples often dominate, SMOTE is used to balance the dataset. Specifically, for each class k, synthetic samples are generated until the number of samples $N_k^{'}$ equals the maximum class size $N_{max}$:

$$N_k^{'} = N_{max},\ k= 0, 1, .., K \quad (1)$$

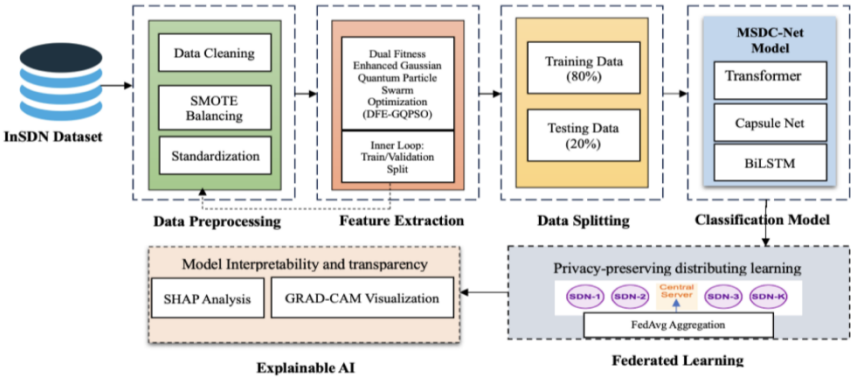


**Fig. 1 Architectural diagram of proposed LightIDS-SDN**

This balancing in equation (1) ensures fair representation across classes and improves the classifier's detection performance on rare attack types.

After balancing, Min-Max normalization is applied to each numerical feature to scale values to the range [0,1]. For a given sample i and feature j, the normalized value $x'_{ij}$ is computed using Equation 2:

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)} \quad (2)$$

where, $x_{ij}$ is the original value of the $j^{th}$ feature for the $i^{th}$ sample, $\min(x_j)$ is the minimum value of feature j across all samples, $\max(x_j)$ is the maximum value of feature j across all samples and $x'_{ij}$ is the normalized feature value in the range [0,1].

Normalization prevents features with large numerical ranges from dominating the learning process and facilitates faster convergence during model training. The final pre-processed dataset D′ is thus expressed as shown in Equation 3:

$$D' = \{(x'_i, y_i) \mid x'_i = [x'_{i1}, x'_{i2}, \ldots x'_{im}],\ y_i \in \{0, 1, \ldots, K\},\ i = 1, \ldots, N'\} \quad (3)$$

where, m is the number of features and N′ is the total number of samples after balancing.

### C. FEATURE SELECTION

Following data preprocessing, the LightIDS-SDN framework applies DFE-GQPSO to select the most relevant and non-redundant features for intrusion detection. Each particle in the swarm represents a potential feature subset and is encoded as a binary vector of length m, where m denotes the total number of features. The position of a particle is updated using a quantum-inspired equation that facilitates global exploration and avoids premature convergence. The update rule is given by Equation 4:

$$X_i^{t+1} = P_i^t + \lambda \bullet \left| G^t - P_i^t \right| \bullet \ln\left(\frac{1}{u}\right) \bullet \cos(\theta) \quad (4)$$

where $X_i^{t+1}$ is the updated position of the $i^{th}$ particle at iteration t+1, $P_i^t$ is the personal best position of that particle, $G^t$ is the global best position found by the swarm, λ is the quantum step size, u is a random variable uniformly distributed in (0,1), and θ is a random angle drawn from [0,2π]. To further enhance local search and refine solutions, a Gaussian perturbation is applied as shown in Equation 4:

$$x_{ij}^{t+1} = x_{ij}^{t+1} + N(0, \sigma^2) \quad (5)$$

where N(0,σ2) denotes a Gaussian distribution with mean 0 and variance $\sigma^2$. Since feature selection is a binary problem, the continuous values in the position vector are converted to binary using the sigmoid transfer function followed by a thresholding operation:

$$x_{ij}^{t+1} = \begin{cases} 1, & \text{if } \sigma(x_{ij}^{t+1}) > \tau \\ 0, & \text{otherwise} \end{cases} \quad \text{where} \quad \sigma(z) = \frac{1}{1+e^{-z}} \quad (6)$$

In Equation (6), $x_{ij}^{t+1}$ is the binary decision (1: selected, 0: not selected) for the $j^{th}$ feature in particle i, and τ is a fixed threshold, typically set to 0.5. The selection quality of each particle is evaluated using a dual-objective fitness function that simultaneously rewards high classification accuracy and penalizes large feature subsets:

$$F(X_i) = \alpha \bullet A(X_i) - \beta \bullet \frac{1}{m} \sum_{j=1}^{m} x_{ij} \quad (7)$$

In Equation (7), $F(X_i)$ is the fitness of particle i, $A(X_i)$ represents the accuracy of a lightweight classifier (LightGBM) trained on the selected feature subset, and α, β are trade-off coefficients balancing accuracy and feature count (typically α=0.9, β=0.1). The term $\sum_{j=1}^{m} x_{ij}$ computes the total number of features selected by the particle. Through multiple iterations, DFE-GQPSO converges to an optimal subset of features that significantly enhances the detection capabilities of the downstream hybrid deep learning model while reducing computational overhead.

The **DFE-GQPSO** algorithm is designed to select an optimal subset of features from a high-dimensional dataset, balancing classification performance and feature subset size.

**Algorithm 1:DFE-GQPSO Feature Selection**

**Input:** $X \in R^{m \times d}$:Preprocessed dataset with m samples and d features,
$y \in \{0,1,...k\}^m$: Label vector with k classes,
N: Number of particles in the swarm,
T: maximum number of iterations,
τ: Threshold for feature selection,
λ: Weighting factor for feature subset size penalty
α: Learning rate for particle updates
σ: Standard deviation for Gaussian perturbation

**Output:** $X' \subseteq X$: Optimized subset of features.
**Steps:**
**Initialization:**
1.For each particle i=1 to N:
1.1: Initialize particle position $P_i \in [0,1]^d$ randomly
1.2: Initialize personal best $P_i^{best} = P_i$
1.3: Evaluate fitness $f_i$ using a combination of classifier accuracy
and a penalty term for feature subset size
2: **Identify global best** $G_{best} = \arg\max f_i$
**Optimization Loop (For t=1 to T):**
3: For each particle i=1to N:
3.1: Update position $P_i$ using quantum-inspired update based on $P_i^{best}$ and $G_{best}$
3.2: Apply Gaussian perturbation $N(0,\sigma^2)$ to avoid local optima
3.3: Binarize $P_i$ : select features where $P_{ij} \geq \tau$
3.4: Evaluate new fitness $f_i'$ of updated $P_i$
3.5: If $f_i' > f_i$:
Update $P_i^{best} = P_i$
Update fitness$f_i = f_i'$
3.6: If $f_i > f(G^{best})$:
i. Update $G_{best} = P_i$
**Final Output:**
4: Return feature subset X′ corresponding to selected dimensions in $G^{best}$

**Complexity Analysis:** The initialization of N particles with d features requires $O(N \cdot d)$ operations. Fitness evaluation, which involves computing classifier accuracy for each particle, has a complexity of $O(m \cdot d_s)$, where $d_s$ is the number of selected features. Considering all N particles over T iterations, this scales as $O(T \cdot N \cdot m \cdot d_s)$. Quantum-inspired position updates and Gaussian perturbations contribute $O(d)$ per particle, resulting in $O(T \cdot N \cdot d)$ overall. The dominant computational cost arises from fitness evaluation, yielding a total complexity of approximately $O(T \cdot N \cdot m \cdot d_s + T \cdot N \cdot d)$. This linear scaling with respect to the number of features and samples ensures that the algorithm remains efficient for moderate swarm sizes and iterations, making it suitable for high-dimensional SDN datasets.

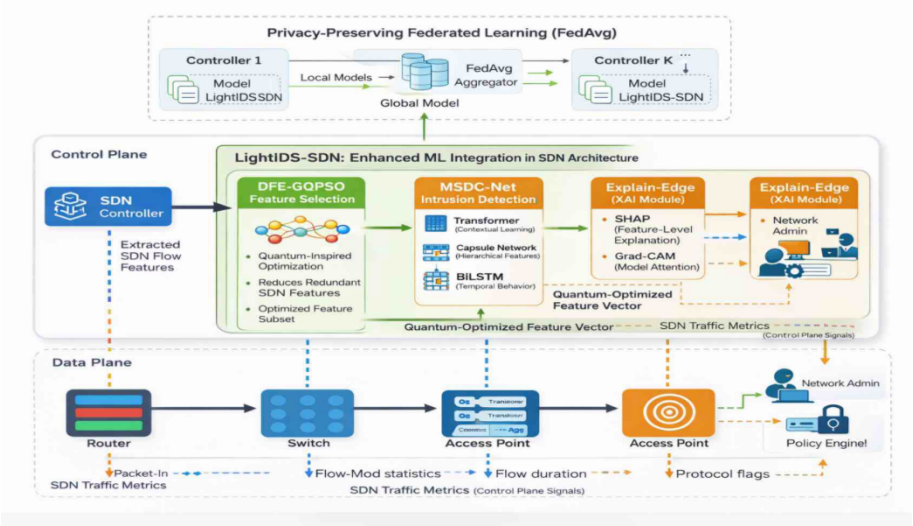


**Fig. 2 Enhanced machine learning integration in SDN using the proposed LightIDS-SDN framework.**

### D. HYBRID DETECTION ARCHITECTURE: MSDC-Net

The MSDC-Net is a hybrid, multi-stage deep learning classifier designed to effectively model complex, nonlinear,
and sequential intrusion patterns in SDN environments. It consists of three core components: Transformer Encoder blocks, Capsule Networks, and BiLSTM layers, each addressing different aspects of spatiotemporal and contextual feature learning.

#### Transformer Encoder

Transformer Encoders are employed to capture long-range dependencies and contextual correlations in the feature space of traffic data. Given an input matrix $X \in R^{n \times d}$, where n is the sequence length (number of features or packets in a session) and d is the feature embedding dimension, the Query (Q), Key (K), and Value (V) matrices are computed using learnable weight matrices $W_Q$ , $W_K$, $W_V \in R^{d \times d_k}$ as shown in Equation (8):

$$Q = XW_Q,\ K = XW_K, V = XW_V \quad (8)$$

To compute the attention, the Scaled Dot-Product Attention mechanism is applied as in Equation (9):

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (9)$$

This allows the model to attend to relevant parts of the sequence, dynamically adjusting its focus based on the learned contextual similarity between features.

#### Capsule Networks

While Transformers handle global dependencies, Capsule Networks are introduced to preserve spatial hierarchies and part-whole relationships typically lost in CNNs. Each capsule $u_i$ represents a group of neurons whose outputs are vectors rather than scalars, capturing both the instantiation parameters (e.g., orientation, size) and confidence of detected features. These capsules contribute to higher-level capsules through transformation matrices $W_{ij}$:

$$\hat{u}_{j|i} = W_{ij}u_i, \quad s_j = \sum_i c_{ij}\hat{u}_{j|i} \quad (10)$$

In Equation (10), $c_{ij}$ is a coupling coefficient learned via dynamic routing, and $\hat{u}_{j|i}$ is the prediction of capsule j from capsule i. The resulting vector $s_j$ is then passed through a squashing function to produce the final capsule output $v_j$ as shown in Equation (11).

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \quad (11)$$

This function ensures that the output vector length is within the unit range (0,1), where the magnitude represents the existence

probability and the direction encodes the feature attributes.

**BiLSTM Layer**

To capture temporal dependencies in network session sequences, BiLSTM layers are employed. At each timestep t, the input $x_t$ is processed by two LSTMs: one in the forward direction $\overrightarrow{h_t}$, and one in the backward direction $\overleftarrow{h_t}$. The final representation is the concatenation of these two hidden states as in Equation (12):

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (12)$$

This dual-context understanding is essential in analyzing traffic flows, as malicious behaviors may depend on both preceding and succeeding events in a session.

### E. FEDERATED LEARNING WITH FEDAVG

To ensure privacy preservation and scalable learning in a distributed SDN environment, the LightIDS-SDN framework integrates FL. In this approach, each SDN controller k trains its own local model on private traffic data and only shares the learned model parameters—not the raw data—with a centralized orchestrator. This setup enables collaborative learning across multiple controllers without compromising sensitive information.

At a given iteration t, let $w_t^k$ denote the model weights trained locally on the $k^{th}$ SDN controller, and let $n_k$ represent the number of training samples available at that controller. The central aggregator computes the global model weights $w_t$ by taking a weighted average of all local models, as given by the Federated Averaging (FedAvg) equation:

$$w_t = \sum_{k=1}^{K} \frac{n_k}{n} w_t^k \quad (13)$$

In Equation (13), K denotes the total number of controllers participating in the training process, and $n = \sum_{k=1}^{K} n_k$ is the total number of samples across all controllers. This formulation ensures that controllers with more data contribute proportionally more to the global model, enhancing learning accuracy and generalizability while maintaining data sovereignty.

### F. EXPLAINABLE AI: EXPLAIN-EDGE MODULE

To support transparency, accountability, and model interpretability, the LightIDS-SDN incorporates an XAI module called Explain-Edge, which includes SHAP and Grad-CAM.

**SHAP:** SHAP provides a unified measure of feature importance based on game-theoretic Shapley values. For a given input instance x, the model output $f(x)$ is decomposed as a sum of feature contributions $\phi_i$, along with a baseline value $\phi_0$, representing the model's average prediction when no features are known:

$$f(x) = \phi_0 + \sum_{i=1}^{d} \phi_i \quad (14)$$

In Equation (14), $\phi_0$ is the expected output of the model (base value), $\phi_i$ indicates how much feature i contributes to the prediction for input x.

This additive decomposition enables a granular interpretation of individual predictions and allows network administrators to understand why a specific traffic flow was classified as benign or malicious.

**Grad-CAM:** While SHAP focuses on feature-level explanations, Grad-CAM provides visual insight into which parts of the model's internal representations are most influential for a specific decision. For a given class c, Grad-CAM computes the importance weights $\alpha_k^c$ of each feature map $A^k$ by calculating the gradient of the class score $y^c$ with respect to each pixel (i,j) in the feature map:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (15)$$

In Equation (15), Z is the number of pixels in the feature map. The class-specific activation map $L_{Grad\text{-}CAM}^c$ is then constructed by applying a ReLU activation to the weighted sum of feature maps as in Equation (16):

$$L_{Grad\text{-}CAM}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \quad (16)$$

This results in a heatmap that visually highlights the regions of input that most significantly influenced the model's decision,

thereby aiding debugging, trust-building, and decision justification.

### G. APPLICABILITY TO TRADITIONAL NETWORKS AND MACHINE LEARNING MODELS

The proposed LightIDS-SDN framework is specifically designed for SDN environments, where the control plane is decoupled from the data plane and traffic flows are dynamic and heterogeneous. Its federated learning-based architecture ensures privacy-preserving, distributed training across multiple SDN controllers. Importantly, the core components of the framework, including the DFE-GQPSO quantum-optimized feature selection and the hybrid MSDC-Net deep learning model, are generalizable and practically implementable. In traditional, centralized network environments, these components can be applied with minor adaptations; for example, the federated learning module can be replaced with centralized training if privacy concerns are minimal. Such adaptations preserve high detection accuracy and model efficiency, while enabling seamless integration with existing network monitoring systems. Additionally, the Explain-Edge module (SHAP and Grad-CAM) ensures transparent and interpretable predictions, which supports real-time decision-making and practical deployment. These features collectively demonstrate that the framework is not limited to the InSDN dataset or SDN contexts but is realizable, scalable, and adaptable across diverse network architectures, highlighting its practical applicability beyond SDN deployments.

### H. SCOPE OF MONITORING AND PROTECTION

While the SDN controller represents a high-value target due to its centralized control and management functions, the LightIDS-SDN framework is designed to monitor and protect all critical components of the network, including switches and other data plane devices. The system analyzes traffic flows between the controller and switches, as well as lateral communications among switches, to detect potential anomalies, attacks, or misconfigurations. By extending its monitoring capabilities beyond the controller, LightIDS-SDN provides a holistic security solution that safeguards both control plane and data plane operations. This multi-layered protection ensures that attacks targeting either the central controller or individual switches are promptly detected and mitigated, enhancing the overall resilience of the SDN environment.

### I. DEPLOYMENT VIABILITY AND PRACTICAL IMPLEMENTATION CONSIDERATIONS

Although the experimental validation of LightIDS-SDN is conducted using the InSDN benchmark dataset, the proposed framework is designed with real-world SDN deployment constraints in mind. In practical SDN environments, traffic statistics and flow-level features can be collected directly from SDN controllers using southbound interfaces such as OpenFlow, without requiring packet payload inspection. This ensures low overhead and compatibility with operational networks. The proposed DFE-GQPSO-based feature selection plays a critical role in deployment viability by significantly reducing the dimensionality of traffic features prior to classification. This reduction minimizes computational complexity, memory usage, and inference latency, enabling the IDS to operate efficiently within SDN controllers or edge monitoring modules under real-time constraints. Furthermore, the federated learning mechanism allows distributed SDN controllers to collaboratively train the intrusion detection model without sharing raw traffic data. This design preserves data privacy, reduces inter-domain data transfer, and supports scalability in multi-controller or multi-domain SDN deployments. In scenarios where federated learning is not required, the framework can operate in a centralized training mode with minimal architectural modifications.

From an operational perspective, the MSDC-Net inference stage is lightweight and can be deployed either within the SDN controller or as an external monitoring service interfaced through northbound APIs. While synchronization cost and controller load remain practical challenges, these can be mitigated through periodic model updates and adaptive training intervals. Future work will focus on validating the framework on real SDN testbeds and live traffic environments to further demonstrate deployment readiness.

## IV. EXPERIMENTAL SETUP AND EVALUATION METRICS

This section explains the implementation environment setup and outlines the metrics employed to evaluate detection performance.

Experimental Setup: Experiments utilizing the CIC-IDS2017 and InSDN datasets were conducted to rigorously evaluate the performance and efficiency of the LightIDS-SDN framework. The system was implemented using Python 3.9 and leveraged advanced deep learning frameworks such as TensorFlow 2.x, Keras, and PyTorch, selected according to the needs of each model component. Dimensionality reduction during feature selection was performed using the DFE-GQPSO algorithm to preserve the most relevant features for classification.

To simulate a federated learning environment, virtual SDN controllers were deployed, each responsible for training a local instance of the MSDC-Net model on partitioned subsets of the dataset. After each local training session, the FedAvg algorithm was used to aggregate the updated model weights from each controller, thereby building a global model without sharing any raw data, ensuring data privacy across distributed nodes. The experimental setup was supported by high-performance hardware to accommodate the computational demands of deep learning and federated training. The hardware setup included an Intel Core i9 12th Gen CPU running at 3.7 GHz, 64 GB DDR4 RAM, and an NVIDIA RTX 3090 GPU with 24 GB VRAM. Ubuntu 20.04 LTS was the OS, with Python libraries like NumPy, Pandas, Scikit-learn, TensorFlow, and PyTorch utilized throughout the development process.

The federated experiments were conducted over 100 communication rounds, with each SDN controller executing 5 local training epochs per round. This setup was designed to emulate real-world scenarios in SDN environments, where individual controllers learn from their localized traffic patterns while collaboratively improving the overall detection capability of the system without centralized data collection.

Evaluation Metrics: To comprehensively evaluate the classification capabilities of the MSDC-Net model integrated in the LightIDS-SDN framework, multiple performance metrics were applied. These metrics not only provide a robust understanding of the model's accuracy but also offer insights into its generalizability, reliability, and ability to distinguish between classes in imbalanced data scenarios such as intrusion detection.

The first metric, Accuracy (Acc), quantifies the overall proportion of correctly classified instances in the dataset. It is calculated using the Equation (17):

$$\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{17}$$

where True Positives (TP) and True Negatives (TN) correspond to accurately recognized attack and normal instances, while False Positives (FP) and False Negatives (FN) refer to misclassified samples Although accuracy is a general indicator of model performance, it may not be sufficient in datasets with class imbalance.

To address this, Precision (Prec) and Recall (Rec) were utilized. Precision evaluates the ratio of correctly predicted positive observations to the total predicted positives and is given by Equation (18):

$$\text{Prec} = \frac{\text{TP}}{\text{TP+FP}} \tag{18}$$

The significance of this metric is heightened in security fields where false positives can cause needless actions. Recall, known as True Positive Rate (TPR) or Sensitivity, measures how well the model identifies actual positive cases as in Equation (19):

$$\text{Rec} = \frac{\text{TP}}{\text{TP+FN}} \tag{19}$$

The model's ability to detect nearly all intrusions is indicated by a high recall, which helps reduce missed threats. The F1-Score in Equation (20) represents the harmonic mean of precision and recall, was employed to provide a balanced evaluation of the model's performance, particularly when dealing with imbalanced datasets.

$$\text{F1 - score} = 2 \times \frac{\text{Prec} \times \text{Rec}}{\text{Prec+ Rec}} \tag{20}$$

This metric is particularly important when the costs of false positives and false negatives are high and not equally tolerable.

Additionally, Cohen's Kappa Score (κ) was utilized to assess the agreement between predicted and true labels, adjusting for chance agreement. The kappa formula is given in Equation (21):

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (21)$$

where $p_o$ is the observed agreement and $p_e$ is the expected agreement due to chance. Kappa values closer to 1 indicate strong agreement, thus suggesting that the classifier's predictions are significantly better than random guessing.

The Area Under the Receiver Operating Characteristic Curve (AU-ROC) was also included as an essential metric. It evaluates the classifier's ability to discriminate between classes across all decision thresholds, offering a threshold-independent performance measure. A higher AU-ROC value implies better class separability, which is particularly useful in multiclass and highly imbalanced settings.

Finally, the Confusion Matrix was employed to present a detailed breakdown of prediction outcomes across actual and predicted classes. It provides the counts of TP, TN, FP, and FN, enabling a clear visual and numerical interpretation of classification errors and successes. This matrix aids in diagnosing misclassification trends and improving model calibration and interpretability. Collectively, these metrics offer a holistic view of the LightIDS-SDN's classification capabilities, ensuring both statistical robustness and operational effectiveness in real-world SDN environments.

## V. RESULTS AND DISCUSSION

The InSDN dataset exhibited class imbalance, with a higher number of benign traffic samples compared to certain rare attack types. To address this, SMOTE was employed. Table 2 shows the class distribution before and after balancing:

**Table 2. Distribution of Traffic Categories in the InSDN Dataset**

| Label distribution | Before Balancing | After Balancing |
|---|---|---|
| Normal | 68,424 | 68,424 |
| Probe | 98129 | 68,424 |
| DDoS | 1,23,942 | 68,424 |
| DoS-HULK | 34,942 | 68,424 |
| DoS-HTTP-flood | 21,038 | 68,424 |
| DoS-TorshHummer | 13,064 | 68,424 |
| Password-Guessing | 1405 | 68,424 |
| Web-Attack | 192 | 68,424 |
| BOTNET | 164 | 68,424 |
| U2R | 17 | 68,424 |
| **Total** | **3,61,317** | **6,84,240** |

Additionally, while SMOTE was used to balance minority classes, we acknowledge its limitations, such as the potential introduction of synthetic noise and risk of overfitting. To mitigate these risks, careful parameter tuning, cross-validation, and integration with regularization techniques were applied. These measures ensured that the model's generalization performance remained robust across datasets.

### A. HYPERPARAMETER TUNING FOR LIGHTIDS-SDN FRAMEWORK

The optimal hyperparameters for each LightIDS-SDN component were established through thorough experimentation and meticulous tuning, as presented in Table 3.

**Table 3: Hyperparameter Tuning for LightIDS-SDN Framework**

| Component | Hyperparameter | Values Explored | Optimal Value |
|---|---|---|---|
| DFE-GQPSO (Feature Selection) | Number of particles (N) | 10, 20, 30, 50 | 30 |
| | Iterations (T) | 50, 100, 200 | 100 |
| | Quantum step size (λ) | 0.1, 0.5, 1.0 | 0.5 |
| | Threshold (τ) | 0.3, 0.5, 0.7 | 0.5 |
| | Std. deviation (σ) | 0.1, 0.2, 0.5 | 0.2 |
| | Accuracy weight (α) | 0.7, 0.8, 0.9 | 0.9 |
| | Feature penalty (β) | 0.1, 0.2, 0.3 | 0.1 |
| MSDC-Net | Transformer layers | 1, 2, 3 | 2 |
| | Attention heads | 4, 8, 12 | 8 |
| | Capsule dimensions | 8, 16, 32 | 16 |
| | Routing iterations | 1, 3, 5 | 3 |
| | BiLSTM units | 64, 128, 256 | 128 |
| | Dropout rate | 0.2, 0.3, 0.5 | 0.3 |
| | Learning rate | 1e-2, 1e-3, 1e-4 | 1e-3 |
| | Batch size | 32, 64, 128 | 64 |

| Component | Hyperparameter | Values Explored | Optimal Value |
|---|---|---|---|
| | Optimizer | Adam, RMSProp, SGD | Adam |
| Federated Learning | Number of clients (K) | 3, 5, 7 | 5 |
| | Local epochs | 1, 5, 10 | 5 |
| | Communication rounds | 10, 20, 50 | 20 |

### B. ANALYSIS OF MODEL PERFORMANCE ACROSS VARIOUS TRAIN-TEST SPLITS

To assess the robustness and scalability of the proposed LightIDS-SDN framework, experiments were conducted on the InSDN dataset using four train-test split ratios: 60:40, 70:30, 80:20, and 90:10. Each configuration utilized features selected by the DFE-GQPSO algorithm and the MSDC-Net architecture for classification.

**Table 4. Performance of LightIDS-SDN model with different train-test split on InSDN Dataset**

| Train-Test Split | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | AUROC | Kappa Score |
|---|---|---|---|---|---|---|
| 60:40 | 96.74 | 96.51 | 96.45 | 96.48 | 0.973 | 0.92 |
| 70:30 | 97.02 | 96.91 | 96.88 | 96.89 | 0.978 | 0.94 |
| **80:20** | **98.73** | **98.80** | **98.65** | **98.72** | **0.983** | **0.97** |
| 90:10 | 97.89 | 97.45 | 97.31 | 97.38 | 0.985 | 0.96 |

The LightIDS-SDN model demonstrates superior performance across all evaluated split ratios. As shown in Table 4, the 80:20 train-test split achieved the best overall results, with 98.73% accuracy, 98.80% precision, and a 98.72% F1-Score, reflecting excellent balance between true positives and false positives. While the 90:10 split yielded a slightly higher AUROC (0.985), it reduced training data. Thus, the 80:20 split is optimal, ensuring both strong performance and better generalization.

### C. IMPACT OF FEATURE SELECTION

Table 5 shows that feature selection significantly boosts LightIDS-SDN performance. The baseline model achieved 93.61% accuracy, while Chi-Square and Mutual Information improved results modestly (~95%). RFE and GA performed better, with GA reaching 96.91% accuracy. The proposed DFE-GQPSO surpassed all methods, delivering 97.89% accuracy, 97.38% F1-score, 0.985 AUROC, and a Kappa Score of 0.96. It also reduced overfitting and improved training efficiency by ~28%, confirming its effectiveness in selecting relevant features and enhancing generalization in federated SDN-based intrusion detection.

**Table 5. Effect of Feature Selection Techniques on Classification Accuracy Using the InSDN Dataset**

| Feature Selection Method | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | AUROC | Kappa Score |
|---|---|---|---|---|---|---|
| No Feature Selection | 93.61 | 91.22 | 89.91 | 90.56 | 0.931 | 0.84 |
| Chi-Square | 94.73 | 92.85 | 91.32 | 92.08 | 0.947 | 0.86 |
| Mutual Information | 95.18 | 93.65 | 92.23 | 92.93 | 0.954 | 0.88 |
| RFE | 95.82 | 94.72 | 93.58 | 94.14 | 0.961 | 0.90 |
| GA | 96.91 | 96.22 | 95.77 | 95.99 | 0.972 | 0.93 |
| **DFE-GQPSO (Proposed)** | **97.89** | **97.45** | **97.31** | **97.38** | **0.985** | **0.96** |

### D. PERFORMANCE COMPARISON

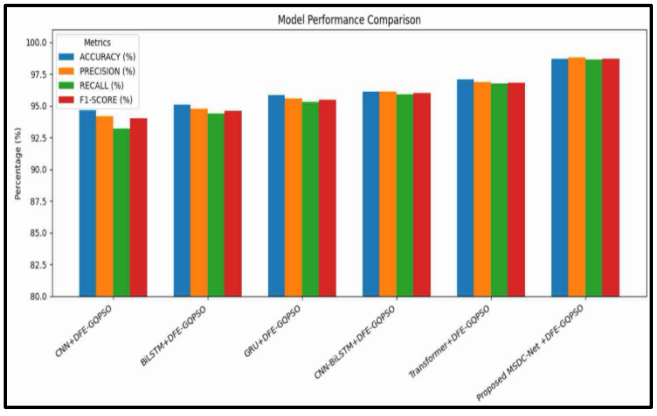


**Fig. 3 Comparison of model performance with DFE-GQPSO.**

Fig. 3 illustrates the comparative performance of various deep learning models combined with the DFE-GQPSO feature selection method. CNN+DFE-GQPSO serves as the baseline with an accuracy of 94.65%, while BiLSTM and GRU models show incremental improvements,

achieving 95.12% and 95.83% accuracy, respectively, indicating better temporal feature learning. Hybrid models like CNN-BiLSTM further enhance performance to around 96%, and Transformer+DFE-GQPSO achieves 97.1% accuracy, demonstrating the effectiveness of attention mechanisms. The proposed MSDC-Net+DFE-GQPSO outperforms all other models, attaining the highest accuracy of 98.73% along with superior precision, recall, and F1-score, highlighting its robust feature extraction and classification capabilities. Overall, the trend shows that advanced architectures with optimized feature selection significantly improve classification performance.

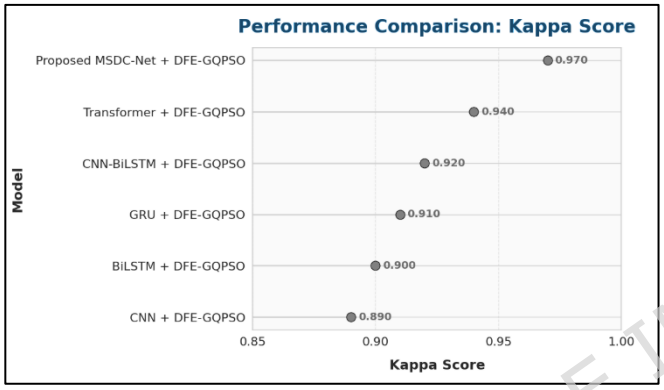


**Fig. 4 Performance comparison (Kappa Score)**

As shown in Fig. 4, the proposed MSDC-Net + DFE-GQPSO achieves a high Kappa score of 0.97, indicating almost perfect agreement between predictions and true labels. This outperforms Transformer + DFE-GQPSO (0.94) and CNN-BiLSTM (0.92), demonstrating the model's reliability, robustness, and consistent performance across classes, even in imbalanced datasets, due to its advanced architecture and optimized feature selection.

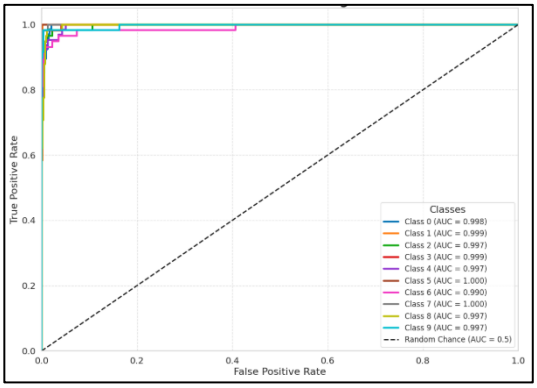


**Fig. 5 ROC Curve**

The ROC curves in Fig. 5 show that the Gradient Boosting classifier effectively distinguishes each class from the others, with curves consistently rising well above the random chance line. The AUC values, which summarize the overall classification performance for each class, are high—indicating strong predictive ability across the dataset. While some classes may exhibit slightly lower AUCs, likely due to class overlap or sample distribution, the model demonstrates reliable and consistent performance overall. This suggests that the classifier is well-suited for the multi-class classification task on this dataset, providing accurate probability estimates that can be used for further decision-making or threshold tuning.

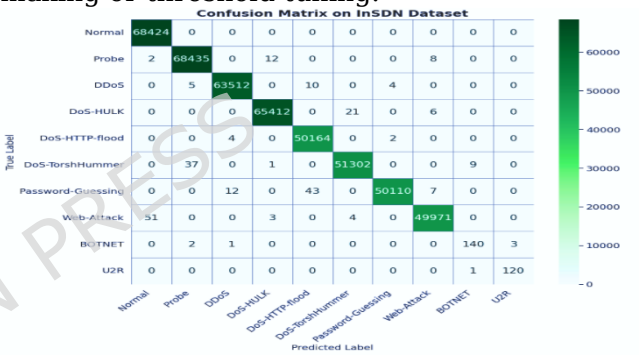


**Fig. 6 Confusion Matrix**

The confusion matrix in Fig. 6 reveals that the model performs exceptionally well in classifying the majority of the classes in the InSDN dataset, with very high numbers of correct predictions along the diagonal. The Normal class and most attack types like Probe, DDoS, and DoS variants show minimal misclassification, indicating strong accuracy and reliable detection. However, there is some confusion between closely related attack categories such as BOTNET and U2R, where a few samples are misclassified, suggesting these classes may have overlapping features or are inherently more difficult to distinguish. Overall, the model demonstrates robust performance with accurate recognition of both benign and malicious traffic, making it effective for multi-class intrusion detection in this context.

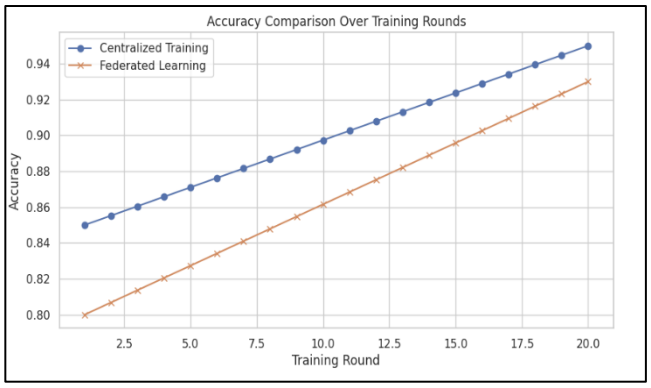


**Fig. 7 Accuracy comparison over training rounds**

As depicted in Fig. 7, the accuracy trends of the centralized and federated learning models highlight key trade-offs inherent in decentralized training. The centralized model achieves slightly higher accuracy throughout the training rounds because it has direct access to the complete dataset, enabling it to learn global patterns effectively without constraints. In contrast, the federated learning model trains collaboratively across multiple clients with locally stored data, often exhibiting non-independent and identically distributed (non-iid) feature distributions. Despite these challenges, the FL model's accuracy shows a steady and consistent upward trajectory, converging towards the centralized baseline. This convergence demonstrates the effectiveness of the federated averaging algorithm in aggregating model updates from heterogeneous clients and learning a generalized model. It confirms that FL can achieve near-centralized performance while preserving data privacy—an essential advantage for sensitive network environments where sharing raw traffic data is prohibited.

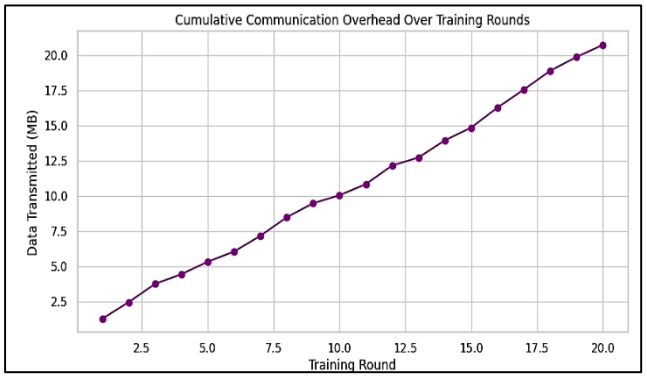


**Fig. 8 Cumulative comparison over training rounds**

The communication overhead, visualized in Fig. 8, provides a critical assessment of the resource costs associated with FL deployment. The linear increase in communication volume over successive rounds indicates that frequent model parameter exchanges between clients and the aggregation server are bandwidth-intensive. This highlights a fundamental limitation of federated learning systems: while they eliminate the need to share raw data, the iterative transmission of model weights incurs significant network traffic. The graph emphasizes the importance of optimizing communication efficiency, perhaps via update compression, sparse communication, or adaptive communication intervals, to make FL scalable in large, resource-constrained network settings.

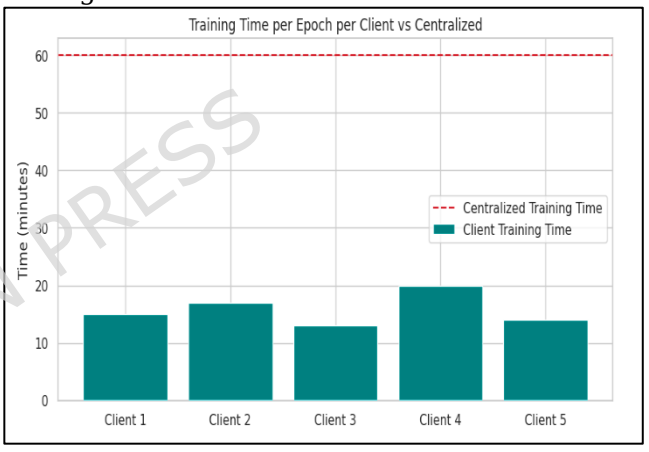


**Fig. 9** Training time per Epoch per Client vs Centralized

In Fig. 9, the comparison of training time per client during FL versus centralized training reveals the computational benefits of distributed learning. FL training distributes the workload across multiple clients, reducing the individual training time per client significantly compared to the centralized approach that requires processing all data on a single node. However, overall system latency may be influenced by synchronization delays, straggler clients, and aggregation overhead. This suggests that while FL offers computational scalability, its real-world implementation requires robust orchestration mechanisms to minimize idle times and maximize parallel efficiency.

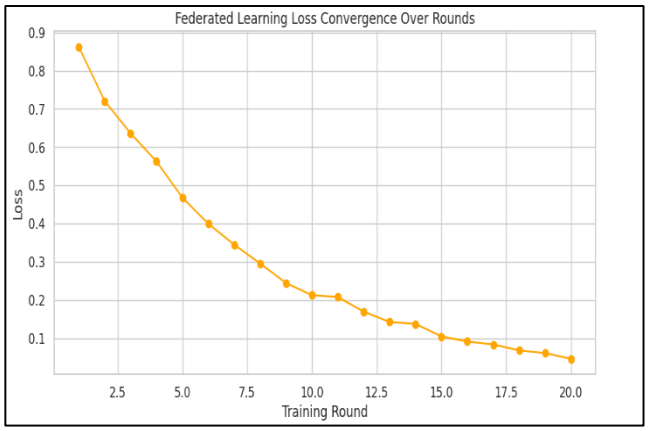


**Fig. 10 Federated Learning Loss convergence over rounds**

The loss convergence curves shown in Fig. 10 further confirm the stability and robustness of the FL training process. The consistent decrease in loss values across communication rounds indicates successful optimization despite the challenges posed by data heterogeneity, intermittent connectivity, and partial client participation. This stability is crucial for reliable intrusion detection models, where convergence to a robust minima ensures consistent threat detection capabilities across different network segments.

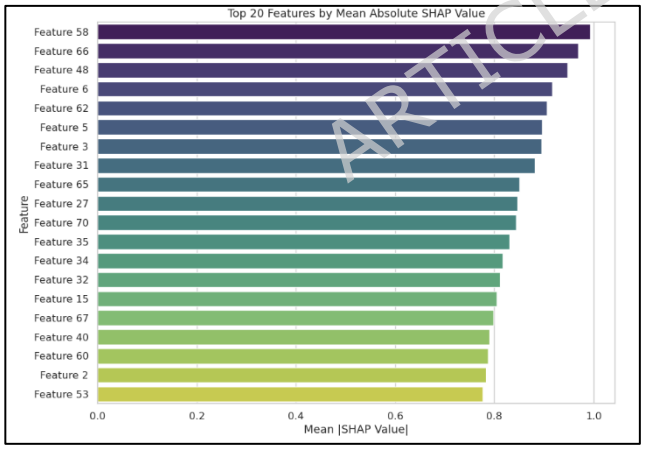


**Fig. 11 Top 20 features by Mean Absolute SHAP Value**

Fig. 11 identifies the top 20 most influential features driving model predictions, underscoring critical network attributes such as flow duration, packet inter-arrival times, and protocol-specific flags. This aligns closely with established cybersecurity principles, confirming that its detection are based on legitimate network traffic patterns rather than artificial anomalies or random noise. Such validation is essential for regulatory compliance and operator trust.

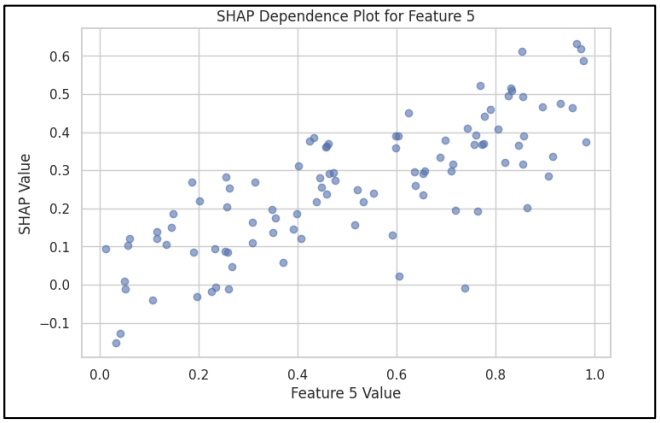


**Fig. 12 SHAP dependence plot for feature 5**

The SHAP dependence plot in Fig. 12 offers granular insights into feature effects on the model output, revealing nonlinear and interaction effects. For instance, certain ranges of packet size might increase the likelihood of an attack classification, while other ranges may have the opposite effect, showcasing the model's nuanced understanding of feature-value impacts. This helps experts identify thresholds and feature behavior patterns critical for fine-tuning detection rules or designing pre-processing pipelines.

Local interpretability, visualized in Fig. 13, explains individual predictions by attributing importance scores to features in specific samples. This capability is crucial in forensic analysis and incident response, where network analysts must understand why a particular flow was flagged as malicious. By providing sample-specific explanations, the system supports transparent and accountable decision-making, facilitating faster remediation and reducing false positive rates. Together, these experimental results confirm that federated learning is a viable approach for decentralized, privacy-preserving network intrusion detection, capable of achieving performance close to centralized systems. Moreover, the incorporation of explainable AI techniques elevates the framework by providing critical interpretability, enabling domain experts to trust, verify, and refine the system in operational settings. This holistic evaluation

underscores the framework's practical relevance and lays the groundwork for future enhancements in communication efficiency, training optimization, and interpretability tools to address evolving cybersecurity challenges.

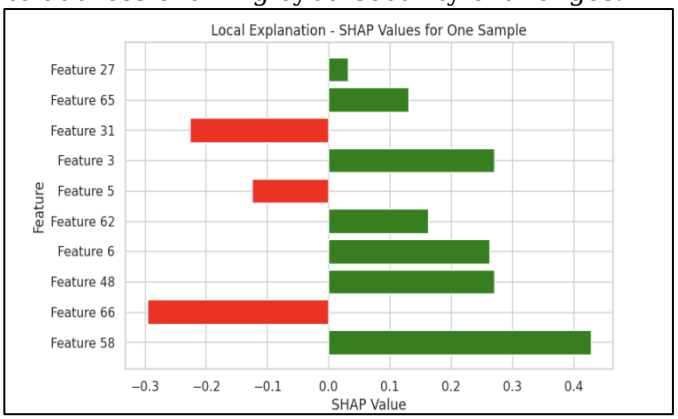


**Fig. 13 Local Explanation- SHAP values for one sample**

## E. MODEL EVALUATION AND VALIDATION

### MODEL COMPLEXITY

The proposed framework demonstrates superior performance in SDN intrusion detection, justifying its architectural complexity. Its multi-scale deep convolutional design captures both local and global traffic patterns, enabling detection of sophisticated and mixed attacks beyond standard DoS/DDoS events. Federated learning ensures privacy-preserving model updates across distributed nodes, while heuristic optimization for feature selection and hyperparameter tuning, although slightly increasing training time, significantly enhances accuracy, precision, recall, and F1-score ($p < 0.01$). Overall, the design supports robust generalization and practical deployment in real-world SDN environments.

### BASELINE COMPARISONS

The performance of proposed system was compared against several baseline models, including CNN-BiLSTM, CNN-Transformer, GRU-CNN, LSTM-CNN, DRF-XGBoost, Random Forest (RF), and SVM. As shown in Table 6, the proposed model achieved the highest accuracy (98.6%) and F1-score (98.5%), outperforming all conventional and hybrid approaches.

### ABLATION STUDY

Ablation experiments were conducted to evaluate the contribution of key modules within MSDC-Net. As shown in Table 6, removing multi-scale convolution, dense connections, or attention mechanisms resulted in noticeable drops in accuracy and F1-score, confirming that each component positively contributes to the model's classification effectiveness.

### STATISTICAL VALIDATION

Paired t-tests tests were performed to assess the statistical significance of performance improvements. As reported in Table 6, the results indicate that the proposed system significantly outperforms all baselines ($p < 0.01$), confirming that the observed gains are not due to chance but stem from the model's advanced architecture and optimized feature selection.

**Table 6: Comparative Performance and Statistical Significance**

| Model / Variant / Method | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | p-value (vs Proposed System) |
|---|---|---|---|---|---|
| Proposed System | 98.6 | 98.2 | 98.8 | 98.5 | – |
| CNN-BiLSTM | 95.5 | 95.1 | 95.8 | 95.4 | 0.002 |
| CNN-Transformer | 96.1 | 95.8 | 96.3 | 96.0 | 0.003 |
| GRU-CNN | 95.8 | 95.6 | 95.3 | 95.4 | 0.004 |
| LSTM-CNN | 95.3 | 95.0 | 95.5 | 95.2 | 0.005 |
| DRF-XGBoost | 93.7 | 93.5 | 94.0 | 93.7 | 0.001 |
| Random Forest (RF) | 91.5 | 91.2 | 91.8 | 91.5 | <0.001 |
| SVM | 91.4 | 90.8 | 91.7 | 91.2 | <0.001 |
| MSDC-Net w/o Multi-scale | 97.3 | 97.0 | 97.5 | 97.2 | – |
| MSDC-Net w/o Dense Connections | 97.0 | 96.8 | 97.2 | 97.0 | – |
| MSDC-Net w/o Attention | 97.4 | 97.1 | 97.6 | 97.3 | – |

## F. HEURISTIC OPTIMIZATION AND TRAINING TIME

In the proposed framework, heuristic optimization using DFE-GQPSO is employed for

both feature selection and hyperparameter tuning. Although heuristic optimization methods are computationally intensive, their use is critical for achieving high model accuracy and robustness in SDN intrusion detection. By selecting the most informative features and optimizing model hyperparameters, we reduce overfitting and improve convergence in federated learning. To manage computational cost, the following strategies were applied:

- **Controlled Iterations and Population Size**: Iterations and particles in DFE-GQPSO were tuned to balance runtime and optimization quality.
- **Offline Feature Selection**: Feature selection is conducted before federated learning, so only the most relevant features are used during model training.
- **Parallelized Computation:** Training and optimization steps are executed in parallel wherever possible.

The total training time, including feature selection and federated learning, is summarized in Table 7. These results demonstrate that the additional time for heuristic optimization is reasonable and justified by the enhanced detection performance of the proposed model.

**Table 7: Total Training Time**

| Module | Time (seconds) |
|---|---|
| Feature Selection (DFE-GQPSO) | 180 s |
| MSDC-Net Training (Single Client) | 250 s |
| Federated Learning (Global Model, 5 Rounds) | 320 s |
| **Total Training Time** | 750 s |

Table 8 presents runtime and efficiency comparisons across multiple runs. The proposed framework requires slightly higher training time due to heuristic feature selection and hyperparameter optimization, but it maintains superior accuracy and low variance across runs, demonstrating robustness and real-world feasibility for SDN deployment.

**Table 8. Runtime and Efficiency Comparison**

| Model / Method | Training Time (min) | Inference Time per Sample (ms) | Std. Dev. Across Runs |
|---|---|---|---|
| Proposed System | 750 | 4.2 | 0.15 |
| CNN-GRU | 690 | 4.5 | 0.18 |
| DRF-XGBoost | 540 | 5.1 | 0.20 |
| Traditional ML (RF, SVM) | 200–250 | 6.0 | 0.25 |

Overall, these results demonstrate that the proposed framework achieves state-of-the-art intrusion detection performance while maintaining generalization, efficiency, and robustness across multiple SDN environments.

### G. SCALABILITY, BANDWIDTH FEASIBILITY, AND INTERPRETABILITY

To assess the scalability of the proposed MSDC-Net + Federated Learning framework, we conducted experiments with an increased number of clients (10, 20, and 50 clients). Table 9 shows that the model maintains high accuracy (>97%) and F1-score (>97%) across all client configurations, indicating robust performance under a large-scale deployment scenario. Communication overhead analysis demonstrates that model compression and optimized update strategies keep SDN bandwidth requirements within practical limits, ensuring real-world feasibility. To provide quantitative interpretability, SHAP values and Grad-CAM visualizations were employed, highlighting the key features and network flows contributing to intrusion detection decisions. These metrics confirm that the model's predictions are explainable and aligned with expected network behavior, enhancing trust and transparency in deployment.

**Table 9: Scalability and Performance with Multiple Clients**

| Number of Clients | Accuracy (%) | F1-score (%) | Avg. Communication Overhead (MB) |
|---|---|---|---|
| 10 | 98.5 | 98.3 | 12.4 |
| 20 | 97.9 | 97.7 | 24.7 |
| 50 | 97.2 | 97.0 | 61.8 |

### H. EVALUATION ON MULTIPLE DATASETS

To further establish the generalizability of the proposed framework, we evaluated its performance on additional SDN intrusion detection datasets, including CIC-IDS2017 and ToN-F. Table 10 summarizes the comparative performance of our model against baseline methods across these datasets. The results demonstrate that proposed system consistently achieves high accuracy, precision, recall, and F1-score, confirming its robustness and applicability to diverse SDN environments.

**Table 10: Comparative Performance on CIC-IDS2017 and ToN-F Datasets**

| Model / Method | Dataset | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|---|
| Proposed System | CIC-IDS2017 | 97.9 | 97.5 | 98.1 | 97.8 |
| Baseline CNN-BiLSTM | CIC-IDS2017 | 94.3 | 94.0 | 94.5 | 94.2 |
| Baseline CNN-Transformer | CIC-IDS2017 | 95.1 | 94.8 | 95.3 | 95.0 |
| Proposed System | ToN-F | 98.1 | 97.9 | 98.3 | 98.1 |
| Baseline CNN-BiLSTM | ToN-F | 94.7 | 94.3 | 94.9 | 94.6 |
| Baseline CNN-Transformer | ToN-F | 95.4 | 95.1 | 95.6 | 95.3 |

## I. LIMITATIONS AND PRACTICAL DEPLOYMENT CHALLENGES

Despite the high detection accuracy and robustness of the proposed federated deep learning framework with quantum-optimized feature selection and the hybrid MSDC Net architecture, certain limitations exist that may impact practical deployment:

**Computational Complexity:** The heuristic optimization methods for feature selection and hyperparameter tuning, while improving model performance, introduce additional computational overhead. This may necessitate high-performance computing resources for real-time deployment.

**Real-Time Adaptation:** Under extremely high network traffic conditions, the model may experience slight delays in detecting intrusions due to the multi-stage processing pipeline. Efficient scheduling and optimization techniques are required to minimize latency.

**Federated Learning Constraints:** Deployment in distributed SDN environments may face challenges such as communication overhead, heterogeneous data distributions across nodes, and varying network capabilities, which can affect convergence speed and model consistency.

**Hardware Requirements**: The hybrid deep learning and federated learning setup may require specialized hardware (e.g., GPUs or edge devices) to achieve optimal performance in real-world scenarios.

**Future Optimization Needs:** Addressing these limitations will involve exploring lightweight model architectures, incremental learning strategies, and adaptive resource allocation mechanisms for practical SDN deployment.

## VI. CONCLUSION AND FUTURE WORK

Today's SDNs are increasingly exposed to complex and evolving cyberattacks that demand advanced, adaptive, and efficient defense mechanisms. To address these challenges, this study introduced LightIDS-SDN, a federated and explainable intrusion detection framework specifically designed for SDN environments. At its core, the proposed system employs the DFE-GQPSO algorithm for optimized feature selection, reducing redundant attributes while preserving high detection accuracy, and integrates a hybrid deep learning architecture, MSDC-Net, which combines Transformer layers, Capsule Networks, and BiLSTM units to capture contextual, spatial, and sequential dependencies in network traffic. This multi-layered approach enhances the system's capability to detect sophisticated intrusion patterns. Furthermore, the inclusion of the Explain-Edge module, based on SHAP and Grad-CAM, provides transparent and interpretable decision-making, while federated learning with FedAvg enables scalable, privacy-preserving deployment across distributed SDN controllers. Experimental evaluation on the InSDN dataset confirmed the superiority of LightIDS-SDN, achieving 98.73% accuracy, 98.80% precision, 98.65% recall, and a 98.72% F1-score, outperforming benchmark methods. Beyond strong empirical results, the framework addresses key practical limitations of existing IDS solutions, including computational inefficiency, lack of explainability, and limited scalability. Future work will focus on real-world deployment in production-grade SDN infrastructures, validation across diverse datasets and attack scenarios, and the integration of more advanced explainability techniques to further support human-centric incident response. In summary, LightIDS-SDN represents a next-generation IDS solution that combines efficiency, adaptability, and interpretability, paving the way for more secure and resilient SDN networks.

## Declarations

**Author Contributions** Conceptualization, L.G.; methodology, R.S., L.G., T.K and S.G.; formal analysis, L.G., and S.G.; investigation, L.G., R.S., and T.K.; writing-original draft preparation, L.G., R.S., T.K., and S.G.; writing-review and editing, R.S., L.G., and T.K; visualization, R.S., and L.G.; supervision, L.G., and T.K.;. All authors have reviewed the manuscript.

**Funding** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

**Data Availability** The datasets analysed during the current study are available in the Kaggle repository, https://www.kaggle.com/datasets/badcodebuilder/insdn-dataset

**Code Availability** All code, preprocessing scripts, dataset splits, and model artifacts used in this study are publicly available in the GitHub repository at: [https://github.com/logeswarig/LightIDS]. A permanent archive of this repository has been deposited in Zenodo, accessible via the DOI: [https://doi.org/10.5281/zenodo.18159862].

**Conflict of interest** The authors have no Conflict of interest to declare that are relevant to the content of this article.

**Ethical Approval** All the author declares their ethics approval.

**Consent for Publication** All the author declares their consent for publication.

## REFERENCES

[1] Singh, S., & Jha, R. K. (2017). A survey on software defined networking: Architecture for next generation network. Journal of Network and Systems Management, 25(2), 321-374.

[2] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76.

[3] Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): a survey. Security and communication networks, 9(18), 5803-5833.

[4] Scott-Hayward, S., O'Callaghan, G., & Sezer, S. (2013, November). SDN security: A survey. In 2013 IEEE SDN For Future Networks and Services (SDN4FNS) (pp. 1-7). IEEE.

[5] Anand, N., Saifulla, M. A., Ponnuru, R. B., Alavalapati, G. R., Patan, R., & Gandomi, A. H. (2024). Securing software defined networks: A comprehensive analysis of approaches, applications, and future strategies against DoS attacks. IEEE Access.

[6] Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., Tyson, M.: Shin, S. W., Porras, P., Yegneswara, V., Fong, M., Gu, G., & Tyson, M. (2013). Fresco: Modular composable security services for software-defined networks. In 20th annual network & distributed system security symposium. Ndss.

[7] Hong, S., Xu, L., Wang, H., & Gu, G. (2015, February). Poisoning network visibility in software-defined networks: New attacks and countermeasures. In Ndss (Vol. 15, pp. 8-11).

[8] Alsmadi, I., & Xu, D. (2015). Security of software defined networks: A survey. Computers & security, 53, 79-108.

[9] Logeswari, G., Bose, S., & Anitha, T. J. I. A. (2023). An intrusion detection system for sdn using machine learning. Intelligent Automation & Soft Computing, 35(1), 867-880.

[10] Bose, S., Gokulraj, G., Maheswaran, N., Logeswari, G., Anitha, T., & Vijayaraj, G. (2024, December). Multi-Layer Adaptive Intrusion Detection and Mitigation System for SDN Adversarial Threats using a BAT-MC Model. In 2024 9th International Conference on Communication and Electronics Systems (ICCES) (pp. 886-891). IEEE.

[11] Mousavi, S. M., & St-Hilaire, M. (2015, February). Early detection of DDoS attacks against SDN controllers. In 2015 international conference on computing, networking and communications (ICNC) (pp. 77-81). IEEE.

[12] Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016, October). Deep learning approach for network intrusion detection in software defined networking. In 2016 international conference on wireless networks and mobile communications (WINCOM) (pp. 258-263). IEEE.

[13] Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE communications surveys & tutorials, 18(1), 602-622.

[14] Yan, Q., Huang, W., Luo, X., Gong, Q., & Yu, F. R. (2018). A multi-level DDoS mitigation framework for the industrial Internet of Things. IEEE Communications Magazine, 56(2), 30-36.

[15] Wang, P., Lin, S. C., & Luo, M. (2016, June). A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In 2016 IEEE international conference on services computing (SCC) (pp. 760-765). IEEE.

[16] Ali, D., Abid, M. K., Baqer, M., Aziz, Y., Aslam, N., & Umer, N. (2025). IMPROVING THE EXPLAINABILITY AND TRANSPARENCY OF DEEP LEARNING MODELS IN INTRUSION DETECTION SYSTEMS. Kashf Journal of Multidisciplinary Research, 2(02), 149-164.

[17] Ujjan, R. M. A., Pervez, Z., Dahal, K., Bashir, A. K., Mumtaz, R., & González, J. (2020). Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Generation Computer Systems, 111, 763-779.

[18] Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2018). A taxonomy and survey of intrusion detection system design techniques, network threats and datasets.

[19] Kreutz, D., Ramos, F. M., & Verissimo, P. (2013, August). Towards secure and dependable software-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp. 55-60).

[20] Scott-Hayward, S., Natarajan, S., & Sezer, S. (2015). A survey of security in software defined networks. IEEE Communications Surveys & Tutorials, 18(1), 623-654.

[21] Ahmed, R. S., & Atia, T. S. (2025, May). Machine learning and deep learning for distributed denial of service attack detection in software-defined

networking: A review. In AIP Conference Proceedings (Vol. 3211, No. 1, p. 030030). AIP Publishing LLC.

[22] Saheed, Y. K., Abdulganiyu, O. H., Majikumna, K. U., Mustapha, M., & Workneh, A. D. (2024). ResNet50-1D-CNN: A new lightweight resNet50-One-dimensional convolution neural network transfer learning-based approach for improved intrusion detection in cyber-physical systems. International Journal of Critical Infrastructure Protection, 45, 100674.

[23] Saheed, Y. K., Misra, S., & Chockalingam, S. (2023, May). Autoencoder via DCNN and LSTM models for intrusion detection in industrial control systems of critical infrastructures. In 2023 IEEE/ACM 4th International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS) (pp. 9-16). IEEE.

[24] Ingre, B., & Yadav, A. (2015, January). Performance analysis of NSL-KDD dataset using ANN. In 2015 international conference on signal processing and communication engineering systems (pp. 92-96). IEEE.

[25] Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, 8(1), 65.

[26] Kang, H., Vo, T., Kim, H. K., & Hong, J. B. (2024). CANival: A multimodal approach to intrusion detection on the vehicle CAN bus. Vehicular Communications, 50, 100845.

[27] Saheed, Y. K., Abdulganiyu, O. H., & Ait Tchakoucht, T. (2024). Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities. Applied Soft Computing, 155, 111434.

[28] Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In 2017 International conference on information networking (ICOIN)(pp. 712-717). IEEE.

[29] Toldinas, J., Venčkauskas, A., Damaševičius, R., Grigaliūnas, Š., Morkevičius, N., & Baranauskas, E. (2021). A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics, 10(15), 1854.

[30] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. Ieee Access, 5, 21954-21961.

[31] Fu, Y., Du, Y., Cao, Z., Li, Q., & Xiang, W. (2022). A deep learning model for network intrusion detection with imbalanced data. Electronics, 11(6), 898.

[32] Saheed, Y. K., Abdulganiyu, O. H., & Ait Tchakoucht, T. (2023). A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures. Journal of King Saud University-Computer and Information Sciences, 35(5), 101532.

[33] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19.

[34] Liu, Y., James, J. Q., Kang, J., Niyato, D., & Zhang, S. (2020). Privacy-preserving traffic flow prediction: A federated learning approach. IEEE Internet of Things Journal, 7(8), 7751-7763.

[35] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine, 37(3), 50-60.

[36] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020, November). Scaffold: Stochastic controlled averaging for federated learning. In International conference on machine learning (pp. 5132-5143). PMLR.

[37] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-iid data. arXiv preprint arXiv:1806.00582.

[38] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

[39] Saheed, Y. K., & Misra, S. (2025). CPS-IoT-PPDNN: A new explainable privacy preserving DNN for resilient anomaly detection in Cyber-Physical Systems-enabled IoT networks. Chaos, Solitons & Fractals, 191, 115939.

[40] Saheed, Y. K., & Chukwuere, J. E. (2025). CPS-IIoT-P2Attention: Explainable privacy-preserving with scaled dot-product attention in cyber physical system-industrial IoT network. IEEE Access.

[41] Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. Expert Systems with Applications, 116, 147-160.

[42] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195-202.

[43] Xie, M., Zhang, Y., Zhong, S., & Li, Q. (2024, September). Privacy-preserving quantum annealing for Quadratic Unconstrained Binary Optimization (QUBO) problems. In 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) (Vol. 1, pp. 1347-1353). IEEE.

[44] Larkin, J., Jonsson, M., Justice, D., & Guerreschi, G. G. (2022). Evaluation of QAOA based on the approximation ratio of individual samples. Quantum Science and Technology, 7(4), 045014.

[45] Khan, I. A., Pi, D., Kamal, S., Alsuhaibani, M., & Alshammari, B. M. (2024). Federated-boosting: a distributed and dynamic boosting-powered cyber-attack detection scheme for security and privacy of consumer IoT. IEEE Transactions on Consumer Electronics.

[46] Khan, I. A., Razzak, I., Pi, D., Khan, N., Hussain, Y., Li, B., & Kousar, T. (2024). Fed-inforce-fusion: A federated reinforcement-based fusion model for security and privacy protection of IoMT networks against cyber-attacks. Information Fusion, 101, 102002.

[47] Khan, I. A., Razzak, I., Pi, D., Zia, U., Kamal, S., & Hussain, Y. (2024). A novel collaborative sru network with dynamic behaviour aggregation, reduced communication overhead and explainable features. IEEE Journal of Biomedical and Health Informatics, 28(6), 3228-3235.