

A Lightweight Federated Deep Learning Framework for Real-Time Intrusion Detection and Autonomous Mitigation in Software-Defined Wireless Networks

Monika Dandotiya

monikadandotiyaphd@gmail.com

MITS: Madhav Institute of Technology and Science <https://orcid.org/0000-0002-5501-4210>

Rajni Ranjan Singh Makwana

Madhav Institute of Technology and Science

Babita Tiwari

Manipal University Jaipur <https://orcid.org/0009-0007-2839-5349>

Research Article

Keywords: Software Defined Networking (SDN), Wireless Network, Distributed Denial of Service (DDoS), Federated Deep Learning (FDL), Privacy-Aware Authentication (PAA), Intrusion Detection System (IDS), Network Security (NS), Real-time Attack Detection, and Mitigation (RADM)

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10312773/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A LIGHTWEIGHT FEDERATED DEEP LEARNING FRAMEWORK FOR REAL-TIME INTRUSION DETECTION AND AUTONOMOUS MITIGATION IN SOFTWARE-DEFINED WIRELESS NETWORKS

Monika Dandotiya¹, Rajni Ranjan Singh Makwana², Babita Tiwari³

¹Assistant Professor, School of Engineering Mathematics & Computing
Madhav Institute of Technology & Science Deemed University, Gwalior,
MP, India, Email-monikadandotiyaphd@gmail.com

²Associate Professor Center for Artificial Intelligence, Madhav Institute of
Technology & Science Deemed University, Gwalior, MP, India, Email-
rrsingh@mitsgwalior.in

³Assistant Professor, Department of CSE, Manipal University Jaipur, Jaipur,
Rajasthan, India, E-mail(s): babita.tiwari@jaipur.manipal.edu

*Corresponding author. E-mail(s): babita.tiwari@jaipur.manipal.edu

Contributing authors: Monika Dandotiya, Rajni Ranjan Singh Makwana

Abstract: In the context of Software Defined Networking (SDN), Distributed Denial of Service (DDoS) attack detection and mitigation are fundamental to improving network security. However, the prevailing methodologies didn't focus on protecting the Service Chaining Controller (SCC) in the control plane of the SDN, thus increasing the risk of hidden DDoS patterns. Therefore, this paper proposes a secure SCC-aware real-time DDoS detection and mitigation framework using enhanced federated deep learning and robust privacy-aware authentication. In the data plane, the users are registered and then log into the network. Before granting access, privacy-aware authentication is performed using Zero Michaelis-Menten Knowledge Proof (Z2MKP), followed by data sensing, data protection, data transmission, and rate limiting. In the control plane, hash verification is carried out to validate the legitimacy of the SCC. After successful hash verification, the data is transmitted to the application plane through the firewall. In the application plane, a local Intrusion Detection System (IDS) is trained by performing data collection, pre-processing, rug plot, feature extraction, DDoS attack classification, and explainability. The proposed Gated Symmetrical Gaussian Contrastive Regularization Recurrent Unit (GSGCRRU) significantly classifies the network packets into normal and DDoS. To adapt to the real-time network traffic patterns, Federated Reinforcement First Synchroton Learning (FRFSL) is employed. Finally, the normal packet data is stored on the server. Overall, the proposed approach had impressive DDoS detection performance with 99.9833% accuracy.

Keywords: *Software Defined Networking (SDN), Wireless Network, Distributed Denial of Service (DDoS), Federated Deep Learning (FDL), Privacy-Aware Authentication (PAA), Intrusion Detection System (IDS), Network Security (NS), Real-time Attack Detection, and Mitigation (RADM).*

1. INTRODUCTION

In the era of modern network management, Software Defined Networking (SDN) has emerged as a transformative paradigm due to its beneficial nature, such as flexibility and scalability

(Wabi et al., 2024). SDN enables dynamic configuration of network resources and centralized control by decoupling the control plane from the data plane (Kachavimath & Narayan, 2025) (Bahashwan et al., 2023). In SDN, the software-based controllers allow network administrators to handle traffic flows and adapt to real-time network conditions (Singh & Jain, 2024) (Yalda et al., 2024). However, SDN faces several security issues, including Distributed Denial of Service (DDoS) attacks that target the SDN controller (Fan et al., 2022). DDoS is an attack condition that disrupts a network or service through a massive amount of internet traffic from multiple compromised sources (Alashhab et al., 2024). Moreover, a DDoS attack disrupts legitimate traffic and degrades the network functions of the SDN (Mansoor et al., 2023).

Therefore, a real-time Intrusion Detection System (IDS) with robust authentication is fundamental to detect and mitigate the DDoS condition by monitoring traffic patterns, ensuring a protected SDN environment (Yungaicela-Naula et al., 2021). To ensure authorized user access, authentication schemes, such as signature-based authentication and certificate-based authentication, were developed (Nisa et al., 2024). However, the existing authentication schemes didn't address sensitive information loss. Therefore, privacy-aware authentication measures like zero-knowledge proof are essential to improve the resilience and reliability of the network. Furthermore, the traditional works established numerous Machine Learning (ML) and Deep Learning (DL) techniques to perform SDN-enabled IDS (Taheri et al., 2023).

The existing studies utilized Random Forest (RF), Decision Tree (DT), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU) to identify the presence of DDoS attacks in the SDN environment (Ali et al., 2023) (Aslam et al., 2023). However, the existing IDS models rely on static learning and centralized data collection, showing adaptability and scalability issues (Anand et al., 2025). Hence, Federated Learning (FL) has emerged as a potential solution to enable multiple local models to collaboratively train a global model without sharing raw data (Ma et al., 2022). The existing SDN security strategies heavily rely on traffic detection but insufficiently ensure the integrity of the SCC responsible for service-function chaining, thus resulting in increased DDoS risk. Therefore, this research is motivated by the need to provide an SCC protection-enabled real-time DDoS detection and mitigation framework in SDN using federated deep learning and robust privacy-aware authentication. Here, the proposed work introduces a novel MBCH to verify the legitimacy of the SCC before allowing control-plane data access. As a result, illegitimate service-chain and compromised information are prevented from being transferred toward the application plane, thus enhancing overall network resilience.

1.1 Problem statement

The potential challenges faced by the prevailing studies are illustrated below,

- ♦ In SDN, SCC played a vital role in maintaining the ordered sequence of network functions, where a packet or traffic flow must pass through before reaching its destination. None of the existing works concentrated on protecting the SCC in the SDN environment, thus resulting in a hidden DDoS pattern and network function collapse.
- ♦ Many of the conventional studies failed to interpret the logic behind the intrusion detection outcomes, showing less trust and improper mitigation decisions.
- ♦ The prevailing study (Alkhamisi et al., 2024) didn't provide DDoS mitigation strategies, potentially affecting the entire network's management capabilities.

- ♦ In real-world SDN environments, network conditions, traffic volume, attack patterns, and controller load varied continuously. Most existing federated frameworks relied on static data distribution, thereby struggling to adapt to the new traffic behaviour. As a result, the model misclassified the normal traffic as DDoS.
- ♦ The existing approach (Hirsi et al., 2024) failed to address unauthorized access to network data owing to the lack of authentication measures, thus resulting in sensitive network information loss and increasing the risk of insider threats.
- ♦ The prior framework (Hnamte & Hussain, 2023) had high bandwidth consumption and considerable transmission delay owing to the large volumes of raw network data transmission to the centralized model.

1.2 Research objectives

The fundamental goals of the research methodology are listed further,

- ♦ To protect the SCC in SDN, the proposed work introduces the Murmur Borda-Carnot Hashing (MBCH) algorithm-based hash verification. As a result, the risk of DDoS through network function is minimized, and network robustness is increased.
- ♦ In the proposed work, an effective Local Clausen Interpretable Model-agnostic Explanation (LCIME) is introduced to interpret the logic behind the intrusion detection, showing high trustworthiness.
- ♦ The proposed work encompasses the crucial mitigation modules, such as rate limiting and firewall deployment, thus ensuring high network security.
- ♦ To adapt to the real-time network conditions, an FRFSL is employed in the proposed work.
- ♦ A novel Zero Michaelis-Menten Knowledge Proof (Z2MKP) is utilized to perform privacy-aware authentication without revealing any sensitive information about the users. As a result, the risk of insider threats is minimized.
- ♦ In the research methodology, federated learning is established to minimize the bandwidth consumption and transmission latency by enabling local model training without transferring raw data to the central server.

1.3 Novel contributions of the proposed framework

The proposed work designs a novel unified architecture that integrates privacy-preservation, secure chaining controller, explainability, and a real-time DDoS detection framework for SDN. Unlike existing studies that mainly focus on traffic classification, the research methodology offers security across the data, control, and application planes in SDN, showing an improved security mechanism. In the proposed work, an effective Z2MKP offers privacy-aware user authentication before network access. Also, the MBCH safeguards the service chaining controller by verifying its hash integrity before control-plane information is forwarded to the application plane. Moreover, the GSGCRRU performs DDoS classification to ensure network security. In addition, the proposed work utilizes LCIME to provide detailed explanations for the classification decisions, thus ensuring transparency. To adapt to the continuously changing network traffic patterns, the proposed methodology employs FRFSL. Additionally, rate

limiting and firewall enforcement enable practical mitigation after malicious traffic is identified. Thus, the proposed work establishes a complete security pipeline to simultaneously address user authentication, SCC protection, DDoS detection, detection transparency, dynamic adaptation, and attack detection, which are basically considered as individual concerns in previous studies.

The remaining part of this article is organized as follows: Section 2 describes the methods and experimental design; Section 3 reviews the related works; Section 4 exhibits the mathematical modelling of the proposed methodology; Section 5 validates the performance of the proposed work, and Section 6 concludes the paper with future enhancements.

2. METHODS/EXPERIMENTAL OVERVIEW

Study design: This study follows a computational, experimental design rather than a clinical or human-subject design; accordingly, no participant recruitment, randomization, or statistical power calculation applies. The aim is to propose and validate a secure, Service Chaining Controller (SCC)-aware federated deep learning framework for real-time DDoS detection and mitigation across the data, control, and application planes of an SDN, and to compare its constituent modules against established baseline techniques under matched dataset and evaluation conditions.

Setting: The proposed framework was implemented and evaluated on the Python programming platform. [Authors: please state the exact Python version, deep-learning library and version (e.g., TensorFlow/PyTorch), and any SDN simulation environment or controller (e.g., Mininet, ONOS, Ryu) used, together with the hardware specifications (CPU/GPU, RAM) on which experiments were run, as these were not specified in the original submission.]

Materials: Two publicly available network intrusion-detection benchmark datasets were used to evaluate the generalizability of the proposed framework: the CIC-IDS 2017 dataset (2,401,124 samples; 2,273,097 normal and 128,027 DDoS instances) and the Ton-IoT dataset (461,043 samples; 161,043 normal and 300,000 DDoS instances). Both datasets provide network packet-level features such as destination port, total forward packets, flow bytes, packet length variance, flow duration, and minimum packet length. For each dataset, 80% of the samples were allocated for training and 20% for testing (CIC-IDS 2017: 1,920,899 training and 480,225 testing samples; Ton-IoT: 368,834 training and 92,209 testing samples).

Interventions/comparisons: Each module of the proposed framework was benchmarked against established baseline techniques under identical data and evaluation conditions: the proposed Gated Symmetrical Gaussian Contrastive Regularization Recurrent Unit (GSGCRRU) was compared with GRU, LSTM, RNN, and CNN for DDoS classification; the proposed Zero Michaelis-Menten Knowledge Proof (Z2MKP) was compared with ZKP, Homomorphic Encryption, Goldwasser-Micali-Rackoff, and the Fiat-Shamir Heuristic for privacy-aware authentication; the proposed Murmur Borda-Carnot Hashing (MBCH) was compared with Murmur Hashing, Tiger Hashing, FarmHash, and SHA for SCC hash verification; the proposed Federated Reinforcement First Synchronon Learning (FRFSL) was compared with RL, Q-Learning, SARSA, and DQN for adaptive global data aggregation; the proposed Leaky Quintic Bucket Algorithm (LQBA) was compared with the Leaky Bucket Algorithm, Token Bucket Algorithm, Fixed Window Counter, and Sliding Window Counter for rate limiting; and the proposed Local Clausen Interpretable Model-agnostic Explanation (LCIME) was compared with LIME, SHAP, Partial Dependence Plot, and Global Surrogate for explainability.

Outcome measures and analysis: Model performance was quantified using accuracy, precision, recall, and F-measure for DDoS classification; hash creation time and hash verification time for SCC integrity checking; authentication level, soundness, and completeness for privacy-aware authentication; cumulative reward and success rate for adaptive global data aggregation; throughput and latency for rate limiting; and fidelity, sparsity, and stability for explainability. As this is not a clinical or human-participant study, a statistical power calculation was not performed; instead, robustness of the proposed framework was assessed by evaluating DDoS classification performance independently on two distinct benchmark datasets (CIC-IDS 2017 and Ton-IoT) and by comparing outcomes against multiple established baseline algorithms for each module.

3. RELATED SURVEY

(Swami et al., 2023) implemented an interquartile range-based DDoS classification and mitigation approach in SDN. The interquartile range measure was applied to the received packets in messages at a time interval. Based on the variation, the attacker was detected. After attack classification, the malicious user was blocked from the SDN environment, ensuring attack mitigation. This framework had a high detection rate and computational efficiency. However, this approach struggled to handle a large-scale environment, showing scalability issues.

(Sayed et al., 2022) propounded a flow-assisted DDoS detection framework using optimal feature selection for SDN. Primarily, the historical attack detection datasets were gathered and then pre-processed for data conversion and normalization. Then, the features were extracted, followed by feature selection. Here, the Information Gain (IG) and RF were utilized to identify the relevant features. Finally, LSTM and autoencoder were utilized to identify the DDoS attacks in SDN. This model had high robustness owing to the rich feature set. But, this framework had considerable transmission latency, degrading network efficiency.

(Varghese & Muniyal, 2021) investigated a robust IDS approach for DDoS attack classification in an SDN environment. Here, the Data Plane Development Kit (DPDK) was employed to enable packet processing and monitoring in the data plane. Similarly, the statistical anomaly detection algorithm, named virtual network function, was utilized to detect DDoS attacks. This approach had better efficiency in DDoS detection. Yet, this framework had memory requirement issues and computational overhead.

(Alkhamisi et al., 2024) established an FL-powered DDoS attack classification for multi-controller SDN. Initially, the intrusion detection dataset was collected, and then features were extracted. To select the optimal features, the mayflies-harmony search algorithm was utilized. Similarly, the Graph Convolutional Neural Network-Gated Recurrent Unit (GCNN-GRU) was employed to classify the normal traffic and DDoS traffic in the SDN environment. This model had the minimum error rate in DDoS classification. However, this approach didn't provide DDoS mitigation strategies.

(Raza et al., 2024) developed a privacy-aware IDS in SDN using FL. In the first step, the collected input data was pre-processed for standardization, followed by feature conversion. Thereafter, FL with an optimized artificial neural network was utilized to perform multi-class attack classification in SDN. This model provided a highly scalable solution and enhanced privacy. Nevertheless, this framework had poor transparency owing to the lack of model interpretation.

(Hirsi et al., 2024) introduced DDoS threat detection and traffic classification in SDN using supervised ML. Here, the ML algorithms, such as logistic regression, support vector machine, RF, k-nearest neighbor, and extreme gradient boosting classifier, were introduced to detect the normal and DDoS traffic in SDN. This approach had minimal overfitting issues and false positives. Nevertheless, this framework didn't generalize well enough to mitigate unauthorized user access.

(Fotse et al., 2025) employed an FL-assisted DDoS attack classification in a large-scale SDN. Firstly, the intrusion detection dataset was gathered from publicly available sources, followed by pre-processing and feature extraction. Thereafter, FL with extreme gradient boosting was used to detect DDoS attacks in an effective and scalable manner. But, this approach struggled to adapt to the evolving and dynamic traffic patterns.

(Al-Ameer & Bhaya, 2023) implemented an enhanced IDS in SDN based on FDL. Primarily, the input data was pre-processed for transformation and normalization. Thereafter, the features were extracted and then input to the local LSTM model for DDoS classification. Subsequently, the model weights were encrypted using homomorphic encryption and then updated into the global LSTM model. This model had better resource utilization. However, this framework had a risk of sensitive information leakage.

(Hnamte & Hussain, 2023) propounded an effective DDoS attack classification framework in an SDN environment. The model utilized a deep convolutional neural network to identify the presence of DDoS traffic. After anomaly detection, a mitigation module (anomaly flow traceback and mitigation policy) was established to improve the model's resilience. But, this approach had maximum bandwidth consumption and latency owing to the large amount of network data.

(Gebremeskel et al., 2023) demonstrated DDoS attack detection and classification based on a hybrid model in multicontroller-assisted SDN. To identify the important features, principal component analysis, DT, RF regressor, and chi-squared test were utilized. Afterward, the LSTM algorithm significantly detected the DDoS attack, followed by attack mitigation. This framework had high reliability and efficiency in DDoS attack classification. But, this model had inter-controller communication overhead issues.

4. PROPOSED METHODOLOGY FOR REAL-TIME DDoS DETECTION AND MITIGATION IN SDN

In this paper, real-time DDoS detection and mitigation in secure SCC-enabled SDN is proposed based on GSGCRRU and Z2MKP. In the proposed work, MBCH is used to validate the legitimacy of the SCC in the control plane of the SDN, thereby minimizing the risk of DDoS. Also, the proposed Z2MKP is generalized well enough to authenticate users without revealing sensitive information, ensuring privacy-aware authentication. The proposed GSGCRRU proficiently classifies the normal and DDoS packets, thus improving network security. The diagrammatic illustration of the proposed work is given in Figure 1.

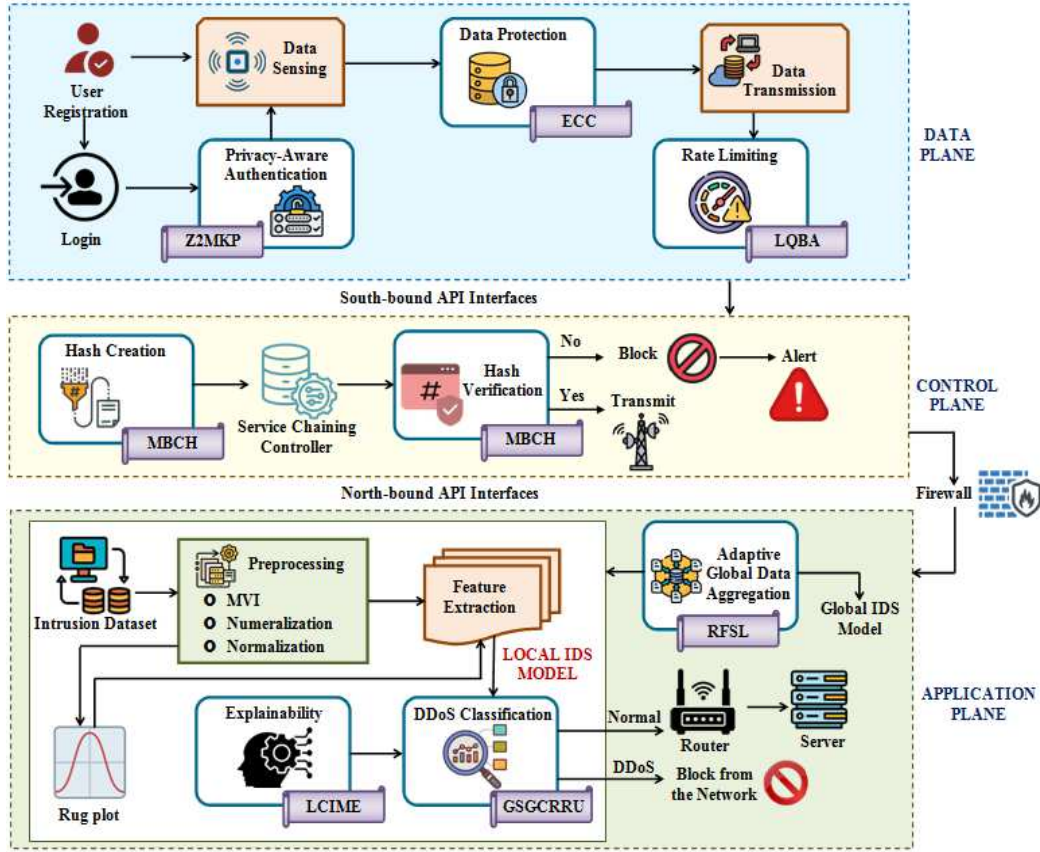


Figure 1: The structural representation of the research methodology

In the research methodology, the SCC-aware protected-SDN framework represents the major system-level novelty because existing DDoS detection systems primarily concentrate on malicious traffic classification and do not consider the integrity of the service chaining controller, which is responsible for the sequence of network functions. A compromised service chaining controller may propagate hidden DDoS and network-function disruption. Therefore, the proposed work introduces MBCH-based SCC hash verification to ensure SCC integrity. On the other hand, the modules, such as Z2MKP, GSGCRRU, LCIME, and FRFSL, are introduced as enhanced approaches. For SDN user-access protection, the Z2MKP adopts the concept of privacy-preserving knowledge-proof authentication. Likewise, the GSGCRRU integrates Gaussian representation and contrastive-learning principles to improve DDoS classification. To provide interpretable explanations, the proposed work adapts local explanation principles of LCIME. Further, FRFSL considers federated learning and reinforcement learning to handle changing SDN traffic conditions. Also, rate limiting and firewall enforcement are established as additional security mechanisms to provide immediate traffic control.

The proposed SDN environment includes three planes, namely the data plane, control plane, and application plane. The detailed overview of the proposed approach is shown below,

4.1 Data plane

The data plane is responsible for forwarding and processing network packets based on flow rules defined by the control plane. The data plane includes user registration, privacy-aware authentication, data transmission, and rate limiting.

4.1.1 User registration

In the data plane, the users are registered in the SDN using their username, password, and email.

$$\gamma_s = (\gamma_1, \gamma_2, \dots \dots \gamma_s) \text{ Here, } s = 1 \text{ to } S \quad (1)$$

Where, S demonstrates the registered users γ_s . After registration, the γ_s users log into the network for data access/data transmission. Before granting access, privacy-aware authentication is carried out to ensure that only authorized users have access to the network.

4.1.2 Privacy-aware authentication

In this step, a robust Z2MKP is utilized to authenticate the user without sharing their sensitive information. Conventional authentication mechanisms may expose secret credentials of authentication-related information. Thus, the proposed Z2MKP is designed as a privacy-aware authentication mechanism that performs authentication without directly revealing sensitive user information. Moreover, the Z2MKP secures user access in the data plane, reducing unauthorized access. The existing Zero Knowledge Proof (ZKP) offers strong security against impersonation. However, the ZKP randomly generates a session integer, thus resulting in secret information leakage. Hence, the proposed work introduces the Michaelis-Menten (MM) equation to select the session integer while ensuring better security. The role of the MM equation is to generate a controlled and bounded session integer based on input parameters, thereby reducing the predictability associated with naïve random integer selection. Also, an enhanced Z2MKP improves session freshness and resistance to replay attempts.

Initially, the session integer (ℓ) is determined using the MM equation. In the proposed work, the MM equation replaces random session-integer selection with a controlled nonlinear selection mechanism, enhancing session-value unpredictability.

$$\ell = \frac{\phi}{\kappa + \gamma_s(\iota)} \quad (2)$$

Where, ϕ depicts the maximum session generation rate, κ indicates the Michaelis constant, and $\gamma_s(\iota)$ illustrates the user identifier strength. Based on the session integer, the user derives a public key (∂).

$$\partial = \varpi^\ell \text{ mod } \wp \quad (3)$$

Here, ϖ indicates the generator, mod depicts the modulus function, and $\wp$ illustrates the large prime number. For authentication purposes, the public key is stored in the server without sharing the secret key.

Authentication request – Thereafter, the user sends a request to the verifier without revealing their secret information. Next, the verifier issues a challenge (λ) to the users.

Proof generation – Here, the user generates a response (Φ) using their secret (h) and the verifier's challenge.

$$\Phi = \ell + \lambda h \bmod (\wp - 1) \quad (4)$$

Then, the pair of (∂, Φ) is sent to the verifier.

Verification – Next, the verifier validates the authenticity of the response by using the expression below,

$$\varpi^\Phi = \partial \cdot \rho^\lambda \bmod \wp \quad (5)$$

Where, ρ denotes the public key stored on the server during registration time. After successful authentication, the data is sensed from the registered users.

4.1.3 Data protection

Here, the sensed data (δ) is encrypted using the Elliptic Curve Cryptography (ECC) for secure data transmission. ECC had a high security level and lower computational overhead. Initially, the elliptic curve is computed as below,

$$q^2 = p^3 + xp + y \quad (6)$$

Where, (p, q) indicates the plane coordinates and (x, y) demonstrates the constant values. Thereafter, the public key and private key are generated to perform encryption and decryption, respectively. Here, the private key is randomly created. Based on the private key (β) , the public key (α) is computed as below,

$$\alpha = \beta \cdot \Im \quad (7)$$

Where, $\Im$ denotes the curve base point. After key generation, the δ is encrypted based on the two cipher components (U_1, U_2) .

$$U_1 = \text{rd} \cdot \Im \quad (8)$$

$$U_2 = \delta + \text{rd} * \alpha \quad (9)$$

Here, rd depicts the random session integer. Contrarily, the sensed data is retrieved using the private key before DDoS detection.

$$\delta = U_2 - \beta * U_1 \quad (10)$$

Thus, the encrypted data (ϕ) is transmitted from the data plane to the control plane. During data transmission, rate limiting is applied to mitigate abnormal traffic.

4.1.4 Rate limiting

To perform rate limiting in the φ , the Leaky Quintic Bucket Algorithm (LQBA) is employed in the proposed work. The rate-limiting process is carried out to restrict the excessive packet transmission within a particular time interval. The existing Leaky Bucket Algorithm (LBA)

has improved model stability and memory efficiency. Leaky Bucket rate limiting mitigates DDoS conditions by enabling strict query flow control. Yet, the traditional LBA was sensitive to the choice of rate limit factor. An improper rate limit increases the computational overhead. Hence, the research methodology utilizes the Quintic Function (QF) to determine the appropriate rate limit factor, improving the model's significance. The QF selects the proper rate limit factor by modelling nonlinear traffic flow variations, thus reducing computational overhead. The role of the QF is to control the permitted traffic flow, improving the stability of the rate limiting.

The proposed LQBA restricts the repeated packets from the same IP address within a particular time period. Firstly, the bucket (V_{bucket}) is initialized with a fixed capacity (v) that indicates the maximum number of packets held by the bucket at a specific time.

$$V_{\text{bucket}} = \max(v) \quad (11)$$

Then, the leak rate limit factor (η) is computed using the QF to decide the packet dropping process. QF selects the suitable rate-limit factor by nonlinearly mapping the current traffic intensity to a bounded candidate value, thereby avoiding unnecessarily restrictive rate limiting.

$$\eta(\varphi) = c\varphi^5 + d\varphi^4 + e\varphi^3 + f\varphi^2 + g\varphi + h \quad (12)$$

Where, (c, d, e, f, g, h) indicates the quintic parameters. Based on the rate limit factor η , the bucket leaks packets to ensure normal data traffic. Here, the bucket's capacity is constantly evaluated before including the packets in the bucket. If the bucket's capacity is available, then the packet is added to the bucket; otherwise, the packet is discarded to mitigate the DDoS traffic. Subsequently, the rate-limited packets are transmitted to the control plane via the south-bound Application Programming Interfaces (API), which enable communication between the data plane and control plane.

4.2 Control plane

The control plane manages the routing and traffic control decisions.

4.2.1 Service chaining controller

In the control plane, SCC determines the ordered sequence of network functions that a packet or traffic flow must pass through before reaching its destination. Attackers compromise the SCC to manipulate the service chain configurations, thus resulting in network function collapse and hidden DDoS. Therefore, the proposed work is generalized well enough to protect the SCC in the control plane, thus minimizing the DDoS risk.

4.2.2 Hash creation

At the time of network setup, the hash value of the controller ID (Z) is created for future verification. In the proposed work, MBCH is utilized to perform hash creation. The existing Murmur Hashing (MH) algorithm had high speed and a low collision rate. Nevertheless, the existing MH had poor randomness owing to the improper seed value initialization. Hence, the proposed approach establishes the Borda-Carnot (BC) equation to select the proper seed value (initial hash), thus improving the security level. The BC produces a data-dependent or state-dependent seed, thus increasing randomness under varying input conditions. The role of BC is to select a more suitable and less predictable initial seed for MH, improving the distribution of

generated hash values. As a result, predictable collision patterns are reduced, and the efficiency of the SCC integrity verification is improved.

Initially, the BC is used to transform the input parameters into candidate seed values and select a seed that produces a lower collision rate, thus improving hashing integrity.

$$I = \text{mod} \left(\mu \cdot \vartheta \frac{xy}{2} (yz - zx)^2, \Xi \right) \quad (13)$$

Where, I indicates the seed value, μ denotes the scaling constant, Ξ depicts the modulus constant, and (ϑ, xy, yz, zx) demonstrates the borda-carnot parameters. Then, Z is divided into 32-bit blocks to enhance the computational efficiency.

$$Z \rightarrow (z_1, z_u, \dots \dots z_U) \text{ Where, } u = 1 \text{ to } U \quad (14)$$

Here, U illustrates the number of divided blocks z_u . Each block is processed by performing constant multiplication, left rotation, mixing, exclusive OR (XOR) operation, and constant addition.

$$\hat{z}_u = z_u \times v_1 \quad (15)$$

$$\tilde{z}_u = \tilde{\mathfrak{R}}(\hat{z}_u, v_2) \quad (16)$$

$$\ddot{z}_u = \tilde{z}_u \cdot v_3 \quad (17)$$

Where, $\tilde{\mathfrak{R}}$ indicates the rotation shift operation, $\hat{z}_u$, $\tilde{z}_u$, and $\ddot{z}_u$ demonstrate the updated blocks after multiplication, rotation, and mixing, respectively, and $(v_1, v_2, v_3, v_4, v_5)$ illustrates the fixed constants. According to the updated blocks, the updated hash value is obtained as below,

$$\hat{I} = I \oplus \ddot{z}_u \quad (18)$$

$$\tilde{I} = \tilde{z}_u(\hat{I}, v_4) \quad (19)$$

$$I''' = \tilde{I} \times v_5 + v_6 \quad (20)$$

Here, $\hat{I}$, $\tilde{I}$, and I''' demonstrate the updated hash values at each mixing round. Lastly, the bit-mixing steps are iteratively applied to obtain an even hash distribution. Thus, the created hash value is mentioned as I''' .

4.2.3 Hash verification

Before transmitting the data through the control plane, the hash verification is done to evaluate the legitimacy of the SCC. The authenticity of the SCC is verified by comparing the recomputed hash value and the hash value stored in the server. Here, the hash value is recomputed by using the proposed MBCH, which is explained in Section 4.2.2. If both the hash values are matched, then the SCC is considered as normal; otherwise, the SCC is blocked from the network, and an alert is then sent to the network administrator. After successful hash verification, the rate-limited packets (ς) are transmitted to the application plane via the firewall. Also, the northbound API is utilized to enable communication between the control plane and

application plane. The firewall acts as a primary defence mechanism that inspects incoming and outgoing traffic to block unauthorized access. A firewall is a barrier between a trusted internal network and untrusted external networks, thereby ensuring DDoS mitigation.

4.3 Application plane

The application plane facilitates network services, such as IDS. The trained local IDS model is explained below.

4.3.1 Intrusion detection dataset

In the first step, the intrusion detection dataset is gathered from publicly available sources to train the proposed IDS framework. The collected dataset provides detailed insight into network packet features, such as source ID, source port, protocol, and flow duration.

$$H_l = \{H_1, H_2, \dots, H_L\} \quad (21)$$

Where, $l = 1, 2, \dots, L$ demonstrates the number of input data H_l .

4.3.2 Preprocessing

Next, the H_l is pre-processed for Missing Value Imputation (MVI), numeralization, and normalization to enhance the data quality. The missing values in the H_l is imputed by considering the mean value of the non-missing information. Then, the missing value-imputed data is converted into numerical vectors in the numeralization. In normalization, the numeralized data (ψ) is represented within the range of 0 to 1 using z-score normalization.

$$\omega = \frac{\psi - \Omega}{\Theta} \quad (22)$$

Where, ω depicts the pre-processed data (normalized data), Ω denotes the mean value of the ψ , and Θ indicates the standard deviation of the ψ .

4.3.3 Rug plot

From the pre-processed data ω , the rug plot is constructed to visualize the data distribution among the normal and attack traffic patterns, thereby improving the overall accuracy of the IDS. Essentially, a rug plot is a one-dimensional data visualization technique that shows the distribution of a single quantitative variable using small lines along the axis. The constructed rug plot is shown as ζ .

4.3.4 Feature extraction

In this step, the ω and ζ are inputted to the feature extraction phase, where the crucial features are extracted to improve the model's learning ability. From ω , the features like protocol, source bytes, destination bytes, destination port, flow duration, total forward packets, total backward packets, total length of forward packets, maximum forward packet length, packet length mean, packet length standard deviation, and maximum idle are extracted. Similarly, the rug plot features, including density, outlier statistics, range and spread features, central tendency

indicators, skewness or distribution shape, overlap metrics, and local density gradient, are extracted. Thus, the extracted features are mentioned as $(\tilde{\tau})$.

4.3.5 DDoS classification

Next, the $\tilde{\tau}$ is fed to the proposed GSGCRRU that classifies the packets into normal and DDoS, thus resulting in high network security. The prevailing works struggled to learn complex temporal traffic relationships, specifically under highly dynamic traffic conditions. Thus, the proposed GSGCRRU enhances DDoS classification through gated temporal learning and contrastive regularization. The existing Gated Recurrent Unit (GRU) had effective long-term dependency handling and faster training. But, improper weight regularization and activation function cause overfitting issues and vanishing gradient issues, respectively. To initialize suitable weight values, the Log Contrastive Regularization (LCR) is established. LCR selects a suitable weight value by penalizing redundant feature correlations, thereby preventing overfitting. During weight initialization, the LCR improves the discriminative quality of learned features by suppressing redundant correlations among feature representations. Moreover, the proposed work employs the Symmetrical Gaussian Error Linear Unit (SGELU) activation function to enhance the learning efficiency of the neurons, avoiding vanishing gradient issues. The SGELU activation function ensures smooth and symmetric gradient flow, effectively mitigating vanishing gradients and accelerating convergence. Also, the SGELU enhances nonlinear feature representations through smooth symmetric activation, thereby improving classification stability. The design of the proposed GSGCRRU is given in Figure 2.

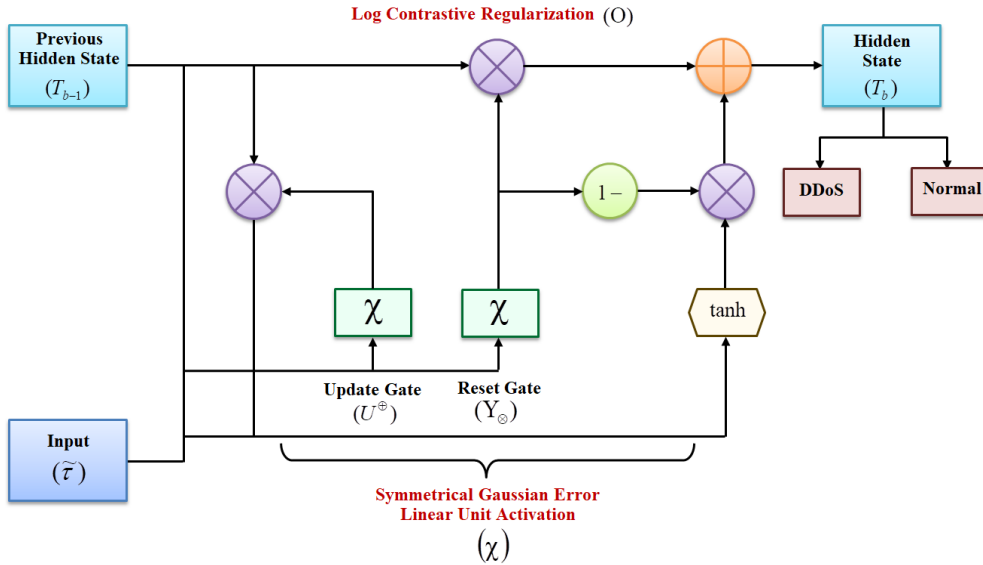


Figure 2: The schematic representation of the proposed GSGCRRU

Primarily, the input and previous hidden state are fed into the update gate. Here, the update gate ($U^\oplus$) determines how much of the information from the previous hidden state (T_{b-1}) is added to the current hidden state.

$$U^\oplus = \chi(O \cdot (\tilde{\tau}, T_{b-1})) + \alpha_{\text{bias}} \quad (23)$$

$$\chi(\tilde{\tau}) = \psi \cdot \tilde{\tau} \cdot e_{\nabla} \left(\frac{\tilde{\tau}}{\sqrt{2}} \right) \quad (24)$$

Where, χ indicates the SGELU activation function, ψ denotes the fixed hyperparameter, e_v illustrates the error function, and α_{bias} depicts the bias value. The SGELU aims to provide stable gradient propagation, improving learning convergence. Also, the proposed work uses the LCR to initialize suitable weight values (O). The role of LCR is to optimize weight initialization by reducing redundant feature correlations.

$$O = \lambda^\circ + \sigma \sum_b \|m(T_b) - m(\tilde{\tau})\|^2 \quad (25)$$

Here, λ° illustrates the loss function, σ depicts the regularization parameter, and m exhibits the initialization function. Likewise, the reset gate ($Y_\otimes$) decides how much of the information from the previous hidden state is ignored from the hidden state.

$$Y_\otimes = \chi(O \cdot (\tilde{\tau}, T_{b-1})) + \alpha_{\text{bias}} \quad (26)$$

Subsequently, the candidate hidden state ($\ddot{T}_b$) is calculated to hold new information that needs to be stored in the current hidden state.

$$\ddot{T}_b = \tan(O \cdot (Y_\otimes * \tilde{\tau}, T_{b-1})) + \alpha_{\text{bias}} \quad (27)$$

Where, $\tan$ indicates the hyperbolic tangent activation function. Based on the update gate and candidate hidden state, the current hidden state is computed.

$$T_b = (1 - Y_\otimes) * T_{b-1} + Y_\otimes \otimes \ddot{T}_b \quad (28)$$

Finally, the proposed GSGCRRU classifies the network packets into normal (N_{orm}) and DDoS (D_{dos}). The outcome (K_{out}) of the proposed GSGCRRU is shown below,

$$K_{\text{out}} = [N_{\text{orm}}, D_{\text{dos}}] \quad (29)$$

The pseudo code of the proposed GSGCRRU is given further.

Input: Extracted features $\tilde{\tau}$

Output: Normal N_{orm} and DDoS D_{dos}

Begin

Initialize $\tilde{\tau}, U^\oplus, O, \ddot{T}_b$ and α_{bias}

For 1 to each $\tilde{\tau}$ do,

Transmit input to update gate

Evaluate $U^\oplus = \chi(O \cdot (\tilde{\tau}, T_{b-1})) + \alpha_{\text{bias}}$

Activate SGELU function

$$\chi(\tilde{\tau}) = \psi \cdot \tilde{\tau} \cdot e_v\left(\frac{\tilde{\tau}}{\sqrt{2}}\right)$$

Initialize weight values using LCR

$$O = \lambda^\circ + \sigma \sum_b \|m(T_b) - m(\tilde{\tau})\|^2$$

Perform reset gate

$$Y_\otimes = \chi(O \cdot (\tilde{\tau}, T_{b-1})) + \alpha_{\text{bias}}$$

Compute hidden state

$$T_b = (1 - Y_\otimes) * T_{b-1} + Y_\otimes \otimes \ddot{T}_b$$

If (actual == target)

```

        {
            Stop
        }
    Else
    {
        Back propagation
    }
End If
End For
Return  $K_{out}$ 
End

```

Furthermore, the K_{out} is interpreted to prove the model's trustworthiness.

4.3.6 Explainability

Based on $\tilde{\tau}$ and K_{out} , the proposed LCIME provides explanations for individual detections made by the proposed IDS model. Many DL-based DDoS detection models provide detections without adequately explaining why a traffic sample is classified as malicious, thus reducing operator confidence. Here, an effective LCIME is developed to provide local explanations for classifier decisions, making individual detections more understandable. The traditional Local Interpretable Model-agnostic Explanation (LIME) offers significant insights into how the model makes detections. But, LIME was sensitive to the choice of weight value initialization, thus resulting in poor interpretation. Hence, the proposed work establishes the Clausen Function (CF) to assign the weight value for each feature. The CF offers a smooth and non-linear weighting strategy that captures complex feature interactions, thus resulting in stable outcomes. Overall, the role of CF in enhanced LCIME is to generate adaptive feature weights instead of arbitrarily initialized weights by mapping the feature-related input into a bounded nonlinear weight. Thus, the LCIME allows more influential features to receive higher importance, whereas less relevant features receive lower weights.

- ♦ In the first step, the instances are selected from the feature set $\tilde{\tau}$ to initiate the interpretation process.
- ♦ Next, the selected instances ($S_{in}()$) are slightly perturbed to create an augmented dataset ($\tilde{B}$).

$$S_{in} \xrightarrow{\text{augment}} \tilde{B} \quad (30)$$

- ♦ In the augmented dataset, the weight value (Wt) is assigned to each instance (A''') using CF. Here, CF chooses the proper feature weight by applying a nonlinear transformation to the feature contribution.

$$Wt(A''') = - \int_0^{\hat{\xi}} \log \left| 2 \sin \frac{A'''}{2} \right| dA''' \quad (31)$$

- ♦ Based on the weighted dataset, the local model (L_{model}) is trained to explain the detection made by the complex model.

$$W_t \xrightarrow{\text{train}} L_{\text{model}} \quad (32)$$

- ◆ Here, the coefficient of the local model is visualized to outline the features that significantly contributed to the detection.

The above-mentioned process illustrates the local IDS model training in federated learning. Generally, traffic patterns change rapidly in the SDN. However, federated learning models heavily rely on static local data, showing less adaptability to new normal traffic behaviors. If the global model is trained based on the outdated local models, the model misclassifies the normal traffic as DDoS attacks. Hence, the weight values ($\vec{I}$) of the trained local IDS are dynamically updated in the global IDS model through adaptive global data aggregation.

4.3.7 Adaptive global data aggregation

To adapt to the real-time network conditions, the proposed FRFSL is integrated with federated learning. The conventional federated-learning systems generally perform model aggregation without sufficiently considering continuously changing traffic patterns. Hence, the proposed FRFSL combines federated learning with reinforcement-based adaptation to dynamically handle changing network traffic patterns. The existing Reinforcement Learning (RL) allows for continuous learning and adaptation in dynamic SDN environments. Further, the RL had high convergence and scalability. But, the existing RL had suboptimal outcomes owing to the random discount factor selection. Hence, the proposed work employs the First Synchroton Function (FSF) to select a suitable discount factor, thus improving the reliability of the proposed work. FSF effectively determines the discount factor by synchronizing reward variations with learning updates. By choosing a suitable discount factor value, FSF effectively balances immediate and future rewards, thus ensuring convergence reliability in real-time adaptation.

- ◆ The environment includes local model updates, network traffic statistics, controller load, latency, and global performance metrics.
- ◆ The agent (global server) observes the environment and identifies the current network state (traffic drift, accuracy variation, and delay). Based on the observed state ($\vec{v}$), the agent performs an action ($\vec{\vartheta}$) (weight aggregation).
- ◆ Then, the environment returns a reward based on the performance of the updated global model. According to the reward feedback (R_w), the agent updates its policy ($\vec{Q}$) in a dynamic manner.

$$R_w \xrightarrow{\text{update}} \vec{Q} \quad (33)$$

- ◆ Moreover, the proposed work uses the FSF to determine the discount factor ($\hat{\delta}$). Here, the policy (q-value) is updated as below,

$$Q''(\vec{v}, \vec{\vartheta}) = \vec{Q}(\vec{v}, \vec{\vartheta}) + \ell \left(R_w + \hat{\delta} \max_{\vec{Q}} \vec{Q}(\vec{v}, \vec{\vartheta}) - \vec{Q}(\vec{v}, \vec{\vartheta}) \right)_{\text{rate}} \quad (34)$$

$$\hat{\partial}(\vec{I}) = \vec{I} \int_{\vec{I}}^{\infty} \hat{h}_{\frac{5}{3}}(\vec{t}) d\vec{t} \quad (35)$$

Where, $Q''(\ddot{v}, \ddot{\theta})$ indicates the updated q-value with state $\ddot{v}$ and action $\ddot{\theta}$, respectively, ℓ_{rate} denotes the learning rate, $\hat{h}$ exhibits the discount weighting function, $\vec{t}$ illustrates the time step, and $\vec{d}$ demonstrates the derivative parameter.

Thus, the RL agent learns a policy that adapts aggregation weights dynamically to maintain stable and accurate detection with respect to the dynamic network conditions. The pseudo code of the proposed FRFSL is given below,

Input: Local model weights $\vec{I}$
Output: Global data aggregation

Begin
 Initialize $\vec{I}, \ddot{\theta}, \text{Rw}, \hat{\partial}$ and Q''
 For 1 to each local IDS model
 Set RL environment
 Observe environment AND perform action
 Feedback reward
 $\text{Rw} \xrightarrow{\text{update}} \vec{Q}$
 Initialize discount factor using FSF
 $\hat{\partial}(\vec{I}) = \vec{I} \int_{\vec{I}}^{\infty} \hat{h}_{\frac{5}{3}}(\vec{t}) d\vec{t}$
 Update q-value

$$Q''(\ddot{v}, \ddot{\theta}) = \vec{Q}(\ddot{v}, \ddot{\theta}) + \ell \left(\text{Rw} + \hat{\partial}_{\max_{\vec{Q}}} \vec{Q}(\ddot{v}, \ddot{\theta}) - \vec{Q}(\ddot{v}, \ddot{\theta}) \right)_{\text{rate}}$$

 End For
 Return adaptive global weight update decisions

End

Overall, the proposed methodology enhances network security and resilience through adaptive DDoS detection, mitigation strategies, privacy-aware authentication, and SCC security.

5. RESULTS AND DISCUSSION

In this section, the performance of the proposed approach is evaluated through empirical analysis and comparative validation. The implementation of the proposed work is done on the working platform of PYTHON.

5.1 Dataset description

To validate the generalizability of the proposed DDoS classification framework, the two datasets, namely the CIC-IDS 2017 dataset and the Ton-IoT dataset, are utilized. The source links of the datasets are given in the reference section. The CIC-IDS 2017 and Ton-IoT datasets provide detailed insight into network packet features, such as destination port, total forward packets, flow bytes, packet length variance, flow duration, and minimum packet length.

Table 1: The dataset distributions

Dataset	Normal	DDoS	Total
CIC-IDS 2017	2273097	128027	2401124
Ton-IoT	161043	300000	461043

From the whole data, 80% and 20% are allocated for training and testing, respectively. For the CIC-IDS 2017 dataset, the training and testing involve 1920899 and 480225 data, respectively. For the Ton-IoT dataset, the training and testing involve 368834 and 92209 data, respectively.

5.2 Experimental validation for hash verification

Here, the performance of the proposed MBCH is evaluated by considering the hash creation time and hash verification time.

Table 2: Performance analysis of the proposed MBCH

Techniques	Hash creation time (ms)	Hash verification time (ms)
Proposed MBCH	478	482
MH	622	624
TH	842	845
FH	924	925
SHA	1104	1110

Table 2 validates the performance of the proposed MBCH and existing algorithms, such as MH, Tiger Hashing (TH), FarmHash (FH), and Secure Hashing Algorithm (SHA), to illustrate the model's efficiency in hash verification. The presence of the BC equation-based seed value selection aids in minimizing the time complexity of the hash verification process. Here, the proposed MBCH obtained hash creation time and hash verification time of 478ms and 482ms, respectively. But, the traditional SHA took 1104ms and 1110ms to perform hash creation and hash verification, respectively. Due to the improper seed value selection, the prevailing approaches had limited performance. Thus, the analysis outcomes showed that the proposed work had low time complexity.

5.3 Empirical analysis for DDoS classification regarding the CIC-IDS 2017 dataset

With respect to the CIC-IDS 2017 dataset, the numerical assessment of the proposed GSGCRRU and traditional classifiers, such as GRU, LSTM, RNN, and Convolutional Neural Network (CNN), is carried out.

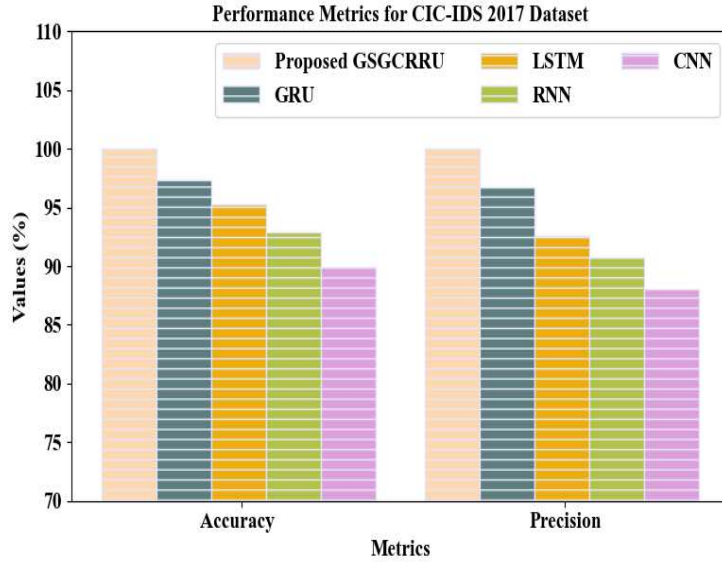


Figure 3: Quantitative analysis for DDoS classification regarding the CIC-IDS 2017 dataset

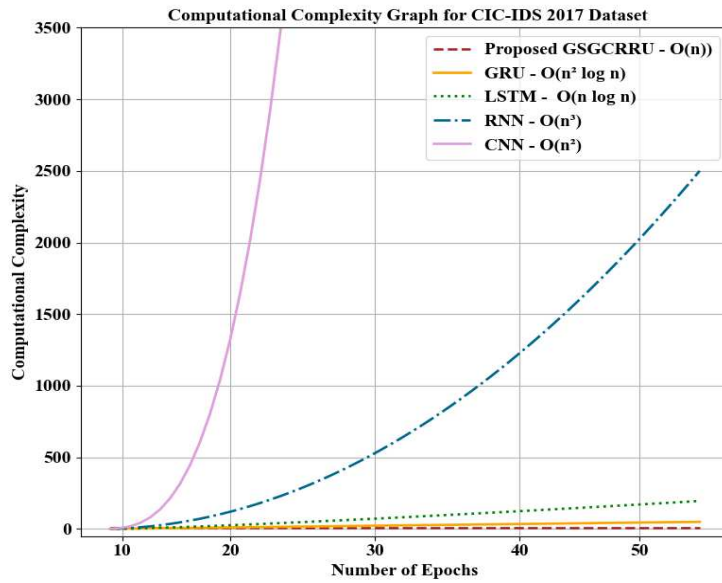


Figure 4: Complexity analysis based on the CIC-IDS 2017 dataset

In Figures 3 and 4, the performance of the proposed GSGCRRU and traditional algorithms is validated with respect to accuracy, precision, and computational complexity. The proposed GSGCRRU obtained accuracy and precision of 99.9833% and 99.9847%, respectively. Due to the improper activation function, the conventional approaches achieved an average accuracy and precision of 93.8356% and 92.0230%, respectively. Owing to the inclusion of the SGELU activation function, the learning efficiency of the proposed GSGCRRU is improved. As the number of epochs increases, the proposed GSGCRRU attained stable complexity. But, the traditional algorithms had a sudden spike in computational complexity. Hence, the superiority of the research methodology was proved.

5.4 Numerical investigation for DDoS classification regarding the Ton-IoT dataset

In order to prove the efficacy of the proposed GSGCRRU in DDoS classification regarding the Ton-IoT dataset, the numerical analysis is carried out.

Table 3: Performance assessment for Ton-IoT dataset

Techniques	Accuracy (%)	Recall (%)	F-measure (%)
Proposed GSGCRRU	99.9342	99.9758	99.9573
GRU	97.4533	97.5336	96.3894
LSTM	94.5433	94.3433	92.4553
RNN	92.5343	92.5534	90.5893
CNN	90.7583	89.8593	89.5893

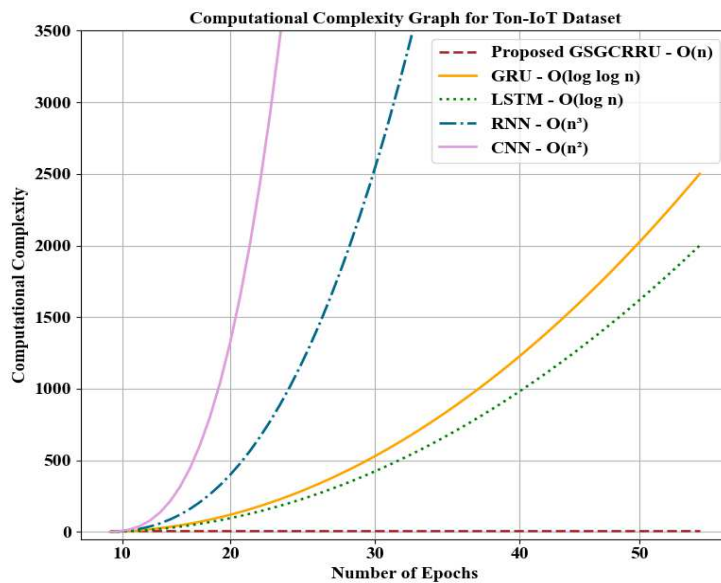


Figure 5: Computational complexity evaluation

The performance assessment of the proposed GSGCRRU and existing approaches, such as GRU, LSTM, RNN, and CNN, is done in Table 3 and Figure 5 with respect to the Ton-IoT dataset. The proposed GSGCRRU achieved accuracy, recall, and f-measure of 99.9342%, 99.9758%, and 99.9573%, respectively. As the number of epochs increases, the complexity of the proposed GSGCRRU is gradually increased. But, the traditional approaches had poor outcomes with high computational overhead owing to the overfitting issues. By mitigating the overfitting issues via LCR, the proposed work had impressive performance in DDoS classification.

5.5 Experimental assessment for privacy-aware authentication

To assess the robustness of the proposed Z2MKP and existing techniques, such as ZKP, Homomorphic Encryption (HE), Goldwasser-Micali-Rackoff (GMR), and Fiat-Shamir Heuristic (FSH), the comparative analysis is done as below,

Table 4: Comparative validation for privacy-aware authentication

Techniques	Authentication level (%)	Soundness	Completeness
Proposed Z2MKP	98.824	0.0048	0.9989
ZKP	95.342	0.012	0.982
HE	92.534	0.019	0.976
GMR	90.553	0.021	0.973
FSH	89.624	0.028	0.966

The authentication level, soundness, and completeness of the proposed Z2MKP and traditional methods are validated in Table 4 to prove the model's efficiency in privacy-aware authentication. Completeness means that an honest prover always convinces an honest verifier of the true statement. Soundness means that a dishonest prover cannot convince an honest verifier consistently with the false statement. The proposed Z2MKP attained authentication level, soundness, and completeness of 98.824%, 0.0048, and 0.9989, respectively. Owing to the improper session integer selection, the conventional ZKP had a 95.342% authentication level, 0.012 soundness, and 0.982 completeness. Due to the presence of MM equation-based session integer initialization, the proposed work had better robustness in privacy-aware authentication.

5.6 Performance evaluation for adaptive global data aggregation

By considering the quality factors, such as cumulative reward and success rate, the performance analysis for adaptive global data aggregation is carried out.

Table 5: Performance validation of the proposed FRFSL

Techniques	Cumulative reward	Success rate
Proposed FRFSL	0.92	0.89
RL	0.85	0.86

QL	0.74	0.73
SARSA	0.68	0.65
DQN	0.53	0.52

To exhibit the significance of the proposed FRFSL in adaptive global data aggregation, the performance of the proposed FRFSL is compared with the existing techniques, such as RL, Q-Learning (QL), State-Action-Reward-State-Action (SARSA), and Deep-Q Network (DQN) in Table 5. Due to the inclusion of FSF-based discount factor selection, the proposed FRFSL obtained cumulative reward and success rate of 0.92 and 0.89, respectively. Similarly, the existing RL, QL, SARSA, and DQN attained cumulative rewards of 0.85, 0.74, 0.68, and 0.53, respectively. Also, the existing approaches attained an average success rate of 0.69. But, the traditional techniques had limited efficacy owing to the improper discount factor selection. Thus, the assessment outcomes showed that the proposed approach had high adaptability and scalability.

5.7 Performance validation for rate limiting

In the same manner, the empirical assessment of the proposed LHABA and existing approaches, such as LBA, Token Bucket Algorithm (TBA), Fixed Window Counter (FWC), and Sliding Window Counter (SWC), is carried out to illustrate the model's prominence in rate limiting.

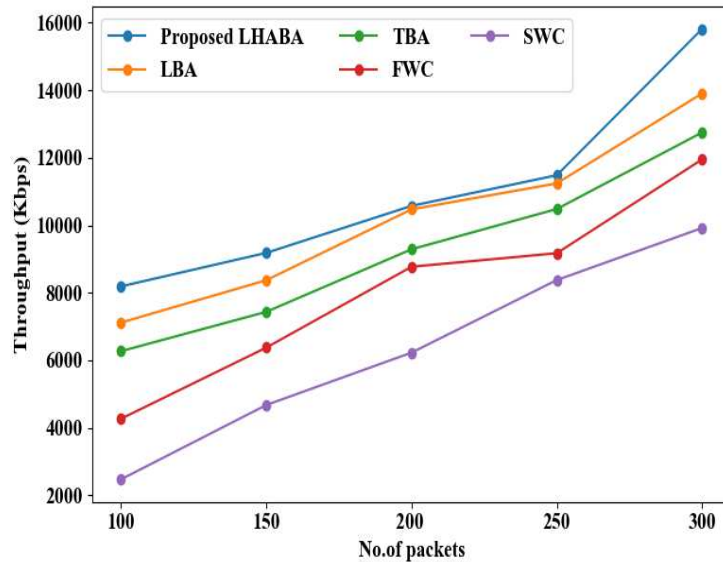


Figure 6: Throughput analysis

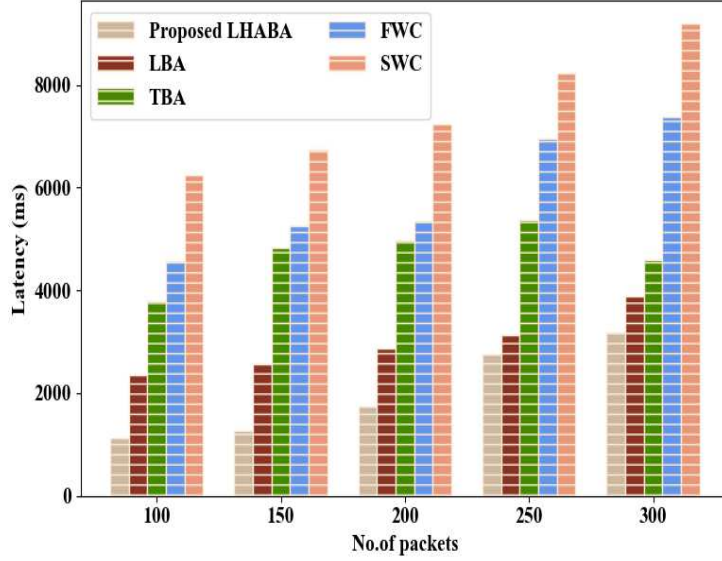


Figure 7: Latency assessment

Figures 6 and 7 depict the performance assessment of the proposed LHABA and existing approaches regarding rate limiting. The utilization of QF effectively determines the rate limit factor, thus resulting in impressive performance. The proposed LHABA achieved throughput and latency of 8191Kbps and 1130ms for 100 packets, respectively. For the same 100 packets, the existing LBA obtained throughput and latency of 7119Kbps and 2353ms, respectively. However, the conventional approaches had considerable latency owing to the improper rate limit factor selection. Thus, the assessment outcomes exhibited the robustness of the proposed approach.

5.8 Numerical analysis for explainability regarding the CIC-IDS 2017 dataset

Regarding the CIC-IDS 2017 dataset, the performance analysis of the proposed LCIME and existing algorithms, such as LIME, SHapley Additive exPlanations (SHAP), Partial Dependence Plot (PDP), and Global Surrogate (GS), is carried out to illustrate the model's significance in explainability.

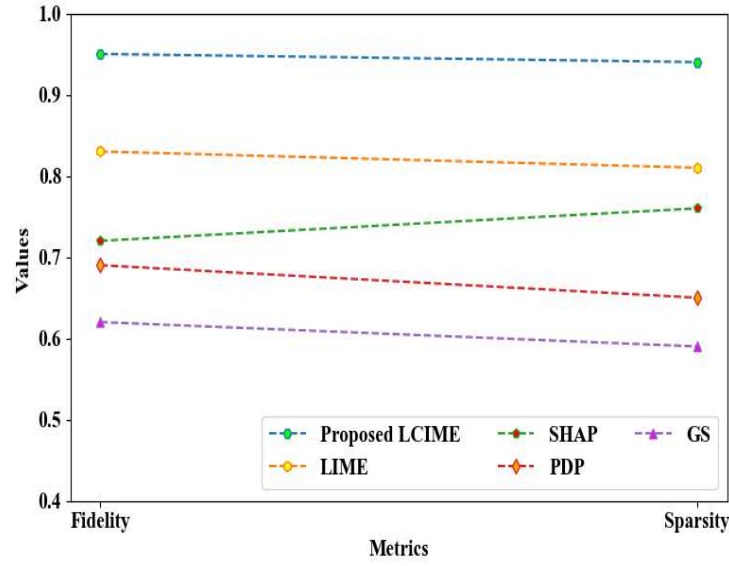


Figure 8: Experimental validation for explainability

The fidelity and sparsity of the proposed LCIME and traditional methods are evaluated in Figure 8 to prove the model's efficiency in explainability. Owing to the presence of CF-based weight value assignment, the proposed LCIME obtained fidelity and sparsity of 0.95 and 0.94, respectively. Yet, the conventional methods had limited efficiency and interpretability owing to the poor weight value initialization. Hence, the transparency of the research methodology was evidenced.

5.9 Empirical assessment for explainability regarding the Ton-IoT dataset

Likewise, the performance of the proposed LCIME and traditional algorithms is evaluated regarding the Ton-IoT dataset.

Table 6: Stability assessment

Techniques	Stability
Proposed LCIME	0.95
LIME	0.91
SHAP	0.84
PDP	0.72

GS	0.67
----	------

The stability assessment of the proposed LCIME and existing techniques is shown in Table 6. Here, the proposed LCIME obtained a stability of 0.95, whereas the conventional algorithms attained an average stability of 0.78. Therefore, the proposed approach had better efficiency in explainability when compared to the traditional algorithms.

5.10 Comparative assessment

The performance of the proposed work is validated by comparing it with associated studies regarding IDS in SDN.

Table 7: Comparative analysis of the research methodology

Author's name	Methods	Accuracy (%)	Precision (%)
Proposed approach	GSGCRRU	99.9833	99.9847
(Makuvaza et al., 2021)	DNN	97.59	97.21
(Bakar et al., 2024)	GNN	99.67	99.25
(Bhayo et al., 2023)	NB, DT, and SVM	97.2	-
(Songa & Karri, 2024)	Rule-based classifier	99.92	-
(Zhao et al., 2023)	CNN and Bidirectional LSTM	95.67	95.82

Table 7 compares the performance of the proposed approach and existing studies regarding DDoS classification in the SDN environment. The proposed GSGCRRU obtained accuracy and precision of 99.9833% and 99.9847%, respectively. The utilization of LCR-based weight initialization and SGELU activation aids in reducing overfitting and vanishing gradient issues, respectively. Similarly, the traditional approaches established Deep Neural Network (DNN), Graph Neural Network (GNN), Naïve Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM), rule-based classifier, CNN, and Bidirectional LSTM (BiLSTM) to recognize the presence of DDoS patterns in the SDN environment. Here, the existing DNN (Makuvaza et al., 2021) attained accuracy and precision of 97.59% and 97.21%, respectively. Also, the traditional CNN with BiLSTM (Zhao et al., 2023) had 95.67% accuracy and 95.82% precision. The conventional studies had limited performance owing to the poor weight initialization and activation function. Therefore, the analysis outcomes showed that the proposed approach is more effective and resilient in DDoS classification than existing works.

6. CONCLUSION

This article proposed a real-time DDoS detection and mitigation in secure SCC-enabled SDN based on GSGCRRU and Z2MKP. To ensure privacy-aware authentication, a robust Z2MKP was established. In the proposed work, MBCH was employed to authenticate the legitimacy of the SCC in the control plane via hash verification, minimizing the risk of DDoS threats. Likewise, the proposed methodology encompassed mitigation strategies, such as rate limiting and a firewall, to improve network resilience. Further, the proposed GSGCRRU proficiently classifies the network packets into normal and DDoS regarding two datasets, namely CIC-IDS 2017 and Ton-IoT. To adapt to the real-time network patterns and ensure scalability, FRFSL was introduced in the proposed approach. Regarding the CIC-IDS 2017 and Ton-IoT datasets, the proposed GSGCRRU achieved an accuracy of 99.9833% and 99.9342%, respectively. For adaptive global data aggregation, the proposed FRFSL had a 0.92 cumulative reward. Also, the proposed Z2MKP obtained an authentication level of 98.824%, showing better security. The proposed framework provided a robust and scalable DDoS detection framework for an SDN environment. But, the proposed work primarily focused on DDoS attack classification and mitigation.

Future scope: In the future, this work will concentrate on detecting the numerous types of cyber threats across all layers in the SDN environment using advanced deep learning schemes to improve network robustness.

LIST OF ABBREVIATIONS

API: Application Programming Interface

BC: Borda-Carnot

BiLSTM: Bidirectional Long Short-Term Memory

CIC-IDS 2017: Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset 2017

CNN: Convolutional Neural Network

DDoS: Distributed Denial of Service

DL: Deep Learning

DNN: Deep Neural Network

DPDK: Data Plane Development Kit

DQN: Deep-Q Network

DT: Decision Tree

ECC: Elliptic Curve Cryptography

FDL: Federated Deep Learning

FL: Federated Learning

FRFSL: Federated Reinforcement First Synchronon Learning

FSH: Fiat-Shamir Heuristic

GCNN-GRU: Graph Convolutional Neural Network-Gated Recurrent Unit

GMR: Goldwasser-Micali-Rackoff

GNN: Graph Neural Network
GRU: Gated Recurrent Unit
GSGCRRU: Gated Symmetrical Gaussian Contrastive Regularization Recurrent Unit
HE: Homomorphic Encryption
IDS: Intrusion Detection System
IG: Information Gain
LBA: Leaky Bucket Algorithm
LCIME: Local Clausen Interpretable Model-agnostic Explanation
LCR: Log Contrastive Regularization
LIME: Local Interpretable Model-agnostic Explanation
LQBA: Leaky Quintic Bucket Algorithm
LSTM: Long Short-Term Memory
MBCH: Murmur Borda-Carnot Hashing
MH: Murmur Hashing
ML: Machine Learning
MM: Michaelis-Menten
NB: Naive Bayes
NS: Network Security
PAA: Privacy-Aware Authentication
QF: Quintic Function
RADM: Real-time Attack Detection and Mitigation
RF: Random Forest
RL: Reinforcement Learning
RNN: Recurrent Neural Network
SCC: Service Chaining Controller
SDN: Software Defined Networking
SGELU: Symmetrical Gaussian Error Linear Unit
SHA: Secure Hashing Algorithm
SHAP: SHapley Additive exPlanations
SVM: Support Vector Machine
TH: Tiger Hashing
Ton-IoT: Telemetry of IoT Dataset
XOR: Exclusive OR
Z2MKP: Zero Michaelis-Menten Knowledge Proof
ZKP: Zero Knowledge Proof

DECLARATIONS

Availability of data and materials

The datasets analysed during the current study are publicly available. The CIC-IDS 2017 dataset is available at <https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset>, and the Ton-IoT dataset is available at <https://research.unsw.edu.au/projects/toniot-datasets>.

Competing interests

The authors declare that they have no competing interests.

Funding

The authors declare that no funding was received for this research.

Authors' contributions

MD conceived the study, developed the proposed framework, performed the experiments, and drafted the manuscript. RRS participated in the design of the methodology, supervised the technical development, and reviewed the manuscript. BT participated in the study design, performed the analysis of the results, and helped revise the manuscript. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

Authors' information

Monika Dandotiya is an Assistant Professor in the School of Engineering Mathematics & Computing, Madhav Institute of Technology & Science Deemed University, Gwalior, Madhya Pradesh, India. Rajni Ranjan Singh Makwana is an Associate Professor at the Center for Artificial Intelligence, Madhav Institute of Technology & Science Deemed University, Gwalior, Madhya Pradesh, India. Babita Tiwari is an Assistant Professor in the Department of CSE, Manipal University Jaipur, Jaipur, Rajasthan, India.

REFERENCES

Dataset:

CIC-IDS 2017 - <https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset>

TON-IOT - <https://research.unsw.edu.au/projects/toniot-datasets>

Al-Ameer, A. A. A., & Bhaya, W. S. (2023). Enhanced Intrusion Detection in Software-Defined Networks Through Federated Learning and Deep Learning. *Ingenierie Des Systemes d'Information*, 28(5), 1213–1220. <https://doi.org/10.18280/isi.280509>

Alashhab, A. A., Zahid, M. S., Isyaku, B., Elnour, A. A., Nagmeldin, W., Abdelmaboud, A., Abdullah, T. A. A., & Maiwada, U. D. (2024). Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model. *IEEE Access*, 12, 51630–51649. <https://doi.org/10.1109/ACCESS.2024.3384398>

- Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Applied Sciences*, 13(5), 1–27. <https://doi.org/10.3390/app13053183>
- Alkhamisi, A., Katib, I., & Buhari, S. M. (2024). Federated Learning-Based Security Attack Detection for Multi-Controller Software-Defined Networks. *Algorithms*, 17(7), 1–22. <https://doi.org/10.3390/a17070290>
- Anand, N., Saifulla, M. A., Babu Ponnuru, R., Reddy Alavalapati, G., Patan, R., & Gandomi, A. H. (2025). Securing Software Defined Networks: A Comprehensive Analysis of Approaches, Applications, and Future Strategies Against DoS Attacks. *IEEE Access*, 13, 64473–64515. <https://doi.org/10.1109/ACCESS.2024.3520478>
- Aslam, N., Srivastava, S., & Gore, M. M. (2023). A Comprehensive Analysis of Machine Learning- and Deep Learning-Based Solutions for DDoS Attack Detection in SDN. *Arabian Journal for Science and Engineering*, 49(3), 1–41. <https://doi.org/10.1007/s13369-023-08075-2>
- Bahashwan, A. A., Anbar, M., Manickam, S., Al-Amiedy, T. A., Aladaileh, M. A., & Hasbullah, I. H. (2023). A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking. *Sensors*, 23, 1–48. <https://doi.org/10.3390/s23094441>
- Bakar, R. A., Marinis, L. De, Cugini, F., & Paolucci, F. (2024). FTG-Net-E: A hierarchical ensemble graph neural network for DDoS attack detection. *Computer Networks*, 250, 1–13. <https://doi.org/10.1016/j.comnet.2024.110508>
- Bhayo, J., Shah, S. A., Hameed, S., Ahmed, A., Nasir, J., & Draheim, D. (2023). Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. *Engineering Applications of Artificial Intelligence*, 123, 1–17. <https://doi.org/10.1016/j.engappai.2023.106432>
- Fan, C., Kaliyamurthy, N. M., Chen, S., Jiang, H., Zhou, Y., & Campbell, C. (2022). Detection of DDoS Attacks in Software Defined Networking Using Entropy. *Applied Sciences*, 12(1), 1–16. <https://doi.org/10.3390/app12010370>
- Fotse, Y. S. N., Tchendji, V. K., & Velempini, M. (2025). Federated Learning Based DDoS Attacks Detection in Large Scale Software-Defined Network. *IEEE Transactions on Computers*, 74(1), 101–115. <https://doi.org/10.1109/TC.2024.3474180>
- Gebremeskel, T. G., Gameda, K. A., Krishna, T. G., & Ramulu, P. J. (2023). DDoS Attack Detection and Classification Using Hybrid Model for Multicontroller SDN. *Wireless Communications and Mobile Computing*, 2023, 1–18. <https://doi.org/10.1155/2023/9965945>
- Hirsi, A., Audah, L., Salh, A., Alhartomi, M. A., & Ahmed, S. (2024). Detecting DDoS Threats Using Supervised Machine Learning for Traffic Classification in Software Defined Networking. *IEEE Access*, 12, 166675–166702. <https://doi.org/10.1109/ACCESS.2024.3486034>
- Hnamte, V., & Hussain, J. (2023). An efficient DDoS attack detection mechanism in SDN environment. *International Journal of Information Technology (Singapore)*, 15(5), 1–11.

<https://doi.org/10.1007/s41870-023-01332-5>

- Kachavimath, A. V., & Narayan, D. G. (2025). An Efficient DDoS Attack Detection in SDN using Multi-Feature Selection and Ensemble Learning. *Procedia Computer Science*, 252, 241–250. <https://doi.org/10.1016/j.procs.2024.12.026>
- Ma, X., Liao, L., Li, Z., Lai, R. X., & Zhang, M. (2022). Applying Federated Learning in Software-Defined Networks: A Survey. *Symmetry*, 14, 1–27. <https://doi.org/10.3390/sym14020195>
- Makuvaza, A., Jat, D. S., & Gamundani, A. M. (2021). Deep Neural Network (DNN) Solution for Real-time Detection of Distributed Denial of Service (DDoS) Attacks in Software Defined Networks (SDNs). *SN Computer Science*, 2(2), 1–10. <https://doi.org/10.1007/s42979-021-00467-1>
- Mansoor, A., Anbar, M., Bahashwan, A. A., Alabsi, B. A., & Rihan, S. D. A. (2023). Deep Learning-Based Approach for Detecting DDoS Attack on Software-Defined Networking Controller. *Systems*, 11(6), 1–21. <https://doi.org/10.3390/systems11060296>
- Nisa, N., Khan, A. S., Ahmad, Z., & Abdullah, J. (2024). TPAAD: Two-phase authentication system for denial of service attack detection and mitigation using machine learning in software-defined network. *International Journal of Network Management*, 34(3), 1–20. <https://doi.org/10.1002/nem.2258>
- Raza, M., Saeed, M. J., Riaz, M. B., & Sattar, M. A. (2024). Federated Learning for Privacy-Preserving Intrusion Detection in Software-Defined Networks. *IEEE Access*, 12, 69551–69567. <https://doi.org/10.1109/ACCESS.2024.3395997>
- Sayed, M. S. El, Le-Khac, N. A., Azer, M. A., & Jurcut, A. D. (2022). A Flow-Based Anomaly Detection Approach With Feature Selection Method Against DDoS Attacks in SDNs. *IEEE Transactions on Cognitive Communications and Networking*, 8(4), 1862–1880. <https://doi.org/10.1109/TCCN.2022.3186331>
- Singh, C., & Jain, A. K. (2024). A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, 8, 1–17. <https://doi.org/10.1016/j.prime.2024.100543>
- Songa, A. V., & Karri, G. R. (2024). An integrated SDN framework for early detection of DDoS attacks in cloud computing. *Journal of Cloud Computing*, 13(1), 1–22. <https://doi.org/10.1186/s13677-024-00625-9>
- Swami, R., Dave, M., & Ranga, V. (2023). IQR-based approach for DDoS detection and mitigation in SDN. *Defence Technology*, 25, 76–87. <https://doi.org/10.1016/j.dt.2022.10.006>
- Taheri, R., Ahmed, H., & Arslan, E. (2023). Deep learning for the security of software-defined networks: a review. *Cluster Computing*, 26(5), 3089–3112. <https://doi.org/10.1007/s10586-023-04069-9>
- Varghese, J. E., & Muniyal, B. (2021). An Efficient IDS Framework for DDoS Attacks in SDN Environment. *IEEE Access*, 9, 69680–69699. <https://doi.org/10.1109/ACCESS.2021.3078065>

- Wabi, A. A., Idris, I., Olaniyi, O. M., & Ojeniyi, J. A. (2024). DDOS attack detection in SDN: Method of attacks, detection techniques, challenges and research gaps. *Computers and Security*, 139, 1–33. <https://doi.org/10.1016/j.cose.2023.103652>
- Yalda, K. G., Hamad, D. J., Tapus, N., & Okumus, I. T. (2024). Security Issues in Software-Defined Networking (SDN) Environments. In *2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 1–8. https://events.roedu.net/event/7/papers/225/files/331-SDN_Security%20V.5.pdf
- Yungaicela-Naula, N. M., Vargas-Rosales, C., & Perez-Diaz, J. A. (2021). SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. *IEEE Access*, 9, 108495–108512. <https://doi.org/10.1109/ACCESS.2021.3101650>
- Zhao, J., Liu, Y., Zhang, Q., & Zheng, X. (2023). CNN-AttBiLSTM Mechanism: A DDoS Attack Detection Method Based on Attention Mechanism and CNN-BiLSTM. *IEEE Access*, 11, 136308–136317. <https://doi.org/10.1109/ACCESS.2023.3334916>

FIGURE TITLES AND LEGENDS

Figure 1. Structural representation of the proposed research methodology. The architecture spans three SDN planes. The data plane performs user registration, Zero Michaelis-Menten Knowledge Proof (Z2MKP)-based privacy-aware authentication, Elliptic Curve Cryptography (ECC)-based data protection, and Leaky Quintic Bucket Algorithm (LQBA)-based rate limiting. The control plane performs Murmur Borda-Carnot Hashing (MBCH)-based hash creation and verification of the Service Chaining Controller (SCC), blocking and alerting the network administrator if verification fails, and transmitting data through a firewall on success. The application plane performs dataset preprocessing, rug-plot visualization and feature extraction, DDoS classification using the proposed Gated Symmetrical Gaussian Contrastive Regularization Recurrent Unit (GSGCRRU), local explanation via LCIME, and adaptive global model aggregation via Federated Reinforcement First Synchrotron Learning (FRFSL). South-bound and north-bound APIs connect the three planes.

Figure 2. Schematic representation of the proposed GSGCRRU. The diagram illustrates the internal gating structure of the proposed Gated Symmetrical Gaussian Contrastive Regularization Recurrent Unit. The previous hidden state and current input are processed through an update gate and a reset gate, both activated using the Symmetrical Gaussian Error Linear Unit (SGELU, denoted χ). Log Contrastive Regularization (O) is used to initialize the gate weights, and a hyperbolic tangent activation produces the candidate hidden state, which is combined with the previous hidden state to yield the current hidden state and the final normal/DDoS classification output.

Figure 3. Quantitative analysis for DDoS classification on the CIC-IDS 2017 dataset. Bar chart comparing the accuracy and precision (%) achieved by the proposed GSGCRRU against four baseline classifiers (GRU, LSTM, RNN, and CNN) on the CIC-IDS 2017 dataset, showing consistently higher performance for the proposed model on both metrics.

Figure 4. Computational complexity analysis for the CIC-IDS 2017 dataset. Line plot of computational complexity as a function of the number of training epochs (10-55) for the proposed GSGCRRU, denoted $O(n)$, compared with GRU ($O(n^2 \log n)$), LSTM ($O(n \log n)$), RNN ($O(n^3)$), and CNN ($O(n^2)$). The proposed model maintains stable, near-linear complexity

growth, whereas the baseline algorithms, particularly RNN and CNN, show steep increases as the number of epochs grows.

Figure 5. Computational complexity evaluation for the Ton-IoT dataset. Line plot of computational complexity versus number of training epochs (10-55) for the proposed GSGCRRU compared with GRU, LSTM, RNN, and CNN on the Ton-IoT dataset, confirming the stable complexity behaviour of the proposed model observed on the CIC-IDS 2017 dataset (Figure 4) on a second, independent dataset.

Figure 6. Throughput analysis for rate limiting. Line plot of throughput (Kbps) versus number of packets (100-300) for the proposed Leaky Quintic Bucket Algorithm (LQBA, also referred to as LHABA) compared with the Leaky Bucket Algorithm (LBA), Token Bucket Algorithm (TBA), Fixed Window Counter (FWC), and Sliding Window Counter (SWC). The proposed algorithm sustains higher throughput than all baseline techniques as packet volume increases.

Figure 7. Latency assessment for rate limiting. Bar chart of latency (ms) versus number of packets (100-300) for the same five rate-limiting techniques evaluated in Figure 6, showing that the proposed LQBA/LHABA consistently achieves lower latency than LBA, TBA, FWC, and SWC across all tested packet volumes.

Figure 8. Experimental validation for explainability on the CIC-IDS 2017 dataset. Line plot comparing fidelity and sparsity scores of the proposed Local Clausen Interpretable Model-agnostic Explanation (LCIME) against LIME, SHAP, Partial Dependence Plot (PDP), and Global Surrogate (GS), showing higher and more stable fidelity and sparsity values for the proposed explanation method relative to the baseline explainability techniques.