International Transactions on Electrical Engineering and Computer Science
A. S. M. Al-Jabri et.al., Vol: 5, No: 2, pp: 105-122, June 2026

Artificial Intelligence Based Framework for DDoS Defense in Software Defined Networks

Ali Salim Malik Al-Jabri[1], Mina Malekzadeh[1]

Abstract: Software Defined Networking (SDN) offers a flexible and programmable alternative to traditional network architectures by decoupling the control and data planes and introducing a centralized controller and application layer. While this architecture enables dynamic traffic management, centralized policy enforcement, and seamless integration of network services, it also introduces critical vulnerabilities, particularly to Distributed Denial-of-Service (DDoS) attacks. The centralized nature of the SDN controller makes it a prime target, where resource exhaustion or flow table saturation can lead to widespread service disruption. These attacks can severely impact time-sensitive applications and compromise the overall stability of the network. In this work, an AI-driven framework is proposed for detecting and mitigating DDoS attacks in SDN environments. The framework incorporates models from diverse categories, including meta-learning, adaptive reinforcement learning, supervised learning, unsupervised clustering, and deep learning techniques. This broad integration enables a comprehensive evaluation of detection capabilities and ensures adaptability to various traffic patterns and attack scenarios. A hybrid Particle Swarm Optimization Genetic Algorithm (PSO-GA) approach is also presented to fine-tune models' parameters and thereby enhance the detection efficiency of the models. Extensive experiments are conducted using multiple performance metrics to evaluate each model under both optimized and non-optimized conditions. The results demonstrate the effectiveness of the proposed framework in enhancing SDN resilience against DDoS threats.

Keywords: Meta Learning, Deep Learning, Machine Learning, Genetic Algorithm, Particle Swarm Optimization, Software Defined Networking.

1. Introduction

Traditional networks use a two-layer architecture in which the control and data planes are tightly coupled inside each device. This design limits flexibility and programmability, and it forces network functions such as firewalls, load balancers, and monitoring tools to be deployed as device specific, vendor dependent appliances that require manual configuration.

History
Received: 24-03-2026; Revised: 23-04-2026;
Accepted: 02-05-2026

M. Mina
m.malekzadeh@hsu.ac.ir

[1]Department of Electrical and Computer Engineering Faculty, Hakim Sabzevari University, Sabzevar, Iran

These constraints make traditional networks rigid and difficult to automate. Software Defined Networking (SDN) overcomes these limitations by introducing a three-layer architecture that separates the control plane from the data plane and adds a software-based application layer [1]. The application layer consists of software services running on general purpose servers or virtual machines, where they define desired network behavior, such as routing, QoS enforcement, intrusion detection, or traffic engineering, and communicate this intent to the controller through northbound APIs [2]. The control layer, implemented by the SDN controller, centralizes network intelligence and translates application intent into forwarding rules [3]. The infrastructure layer contains the physical and virtual switches that forward packets based on rules installed by the controller via southbound APIs such as OpenFlow [4].

https://doi.org/10.62760/iteecs.5.2.2026.198

This separation enables programmability, centralized management, and rapid adaptation to changing network conditions. SDN has become a foundational technology across cloud, telecom, enterprise, and retail environments because its programmable architecture enables dynamic traffic management, automated provisioning, centralized policy control, and the use of commodity switches instead of proprietary hardware [5]. Cloud providers use SDN to optimize routing and automate large scale data center operations [6], while telecom operators combine SDN with NFV to virtualize core services and orchestrate 5G and edge deployments [7]. Enterprises benefit from simplified management and cost reduction, and SDN's programmability supports rapid security updates, real-time analytics, and adaptive DDoS mitigation across distributed environments.

Despite these advantages, SDN remains highly vulnerable to Distributed Denial of Service (DDoS) attacks [8]. The same features that make SDN flexible, centralized control, programmability, and dynamic flow installation, also create attack surfaces that adversaries can exploit. The controller represents a single point of failure: attackers can overwhelm it with forged flow requests, exhausting CPU and memory resources and causing network wide disruption, especially for latency sensitive applications such as VoIP, streaming, and financial services [9]. These risks are amplified by the limited processing capacity of SDN controllers compared to traditional distributed systems [10]. SDN's reliance on protocols like OpenFlow further exposes it to forged or malicious flow entries, while data plane switches can suffer from flow-table saturation, leading to packet loss and degraded performance. Moreover, SDN's dynamic rule updates increase the risk of misconfigurations or malicious rule injections that can redirect or amplify attack traffic, further compromising network integrity [11]. Artificial intelligence (AI) can play a critical role in preventing DDoS attacks by intelligently analyzing network traffic patterns, detecting anomalies in real time, and responding faster than traditional security systems [12 - 13]. This work is the first to unify meta learning, reinforcement learning, supervised, unsupervised, and deep learning models within a single SDN focused DDoS detection framework, enabling cross paradigm evaluation under identical conditions. It further introduces a hybrid optimizer specifically tailored for SDN intrusion detection, providing a uniquely dual mode assessment that isolates and quantifies optimization-driven performance gains. The key contributions of this work are as follows,

- Integrated AI-driven defense architecture: a unified framework is proposed that leverages a diverse set of deep learning (DL) and machine learning (ML) models, including meta learning, adaptive reinforcement learning (RL), supervised learning, and unsupervised clustering, to enable dynamic model selection based on comparative performance. This integration ensures adaptive DDoS detection across a wide range of traffic patterns and attack scenarios in SDN environments.
- Hybrid optimization strategy: a hybrid Particle Swarm Optimization Genetic Algorithm (PSO-GA) approach is proposed to fine-tune model parameters and improve detection accuracy while maintaining computational efficiency.
- Dual mode evaluation: each model within the framework is implemented and evaluated under both optimized and non-optimized conditions, offering a clear assessment of the impact of the intelligent optimization approach on DDoS detection performance.
- Comparative performance analysis: extensive experiments are conducted using multiple performance metrics, contributing in two distinct ways to identify the most effective model within the framework, and to quantify the benefits introduced by the optimization approach.

The remainder of this work is organized as follows. Section 2 reviews related work. Section 3 presents the proposed methodology. Section 4 discusses the experimental results and analysis. Section 5 concludes the work.

2. Related Works

DDoS attacks have emerged as one of the critical security challenges in SDN environments. Consequently, the detection and mitigation of such attacks is a major focus of research, leading to the development of various strategies aimed at enhancing network resilience. Among the existing solutions, AI techniques have been explored to demonstrate the

potential of intelligent systems to strengthen SDN security in this domain. DDoS detection in SDN has been explored using three machine learning models, SVM, NB, and KNN, along with an ANN-based deep learning model [14]. After applying feature-selection techniques to simplify the dataset and improve training efficiency, the models were evaluated, and KNN achieved the highest accuracy at 98.3%. The study shows that combining ML with feature selection improves detection performance while reducing computational overhead. However, the study is limited in scope due to its reliance on a narrow set of ML and DL models, the lack of adaptive and meta-learning mechanisms, and the absence of optimization techniques for fine tuning model parameters. The feature selection methods for ML models to detect DDoS attacks in SDN environments are also investigated in [15]. The study evaluates several feature selection methods including Information Gain, Correlation Coefficient, Chi-Square, Forward and Backward Selection, Recursive Feature Elimination, and Lasso, using the NSL-KDD dataset to train SVM, KNN, NB, RF, and DT models. Among these, the Random Forest classifier achieves the highest true positive rate, reaching 99.97% accuracy in detecting DDoS attacks. However, the study is limited to ML models and does not incorporate adaptive, deep learning, or meta learning mechanisms. It also lacks optimization strategies to dynamically adjust model parameters in response to evolving traffic patterns. An experimental evaluation for comparison of DDoS detection methods involving artificial intelligence in SDN networks is conducted in [16]. They employ classifiers, including RF, DT, and NB, as well as CNN and RNN for testing and training, with RF achieving the best DDoS detection performance through testing results. However, the research did not explore meta-learning approaches for dynamic model selection or hybrid optimization strategies to enhance detection parameter tuning. Additionally, the limited diversity of classifiers restricts the generalizability of the findings.

Enhancing DDoS detection in SDN-IoT networks using CICIDS2017 and Edge IIoTsett datasets is provided in [17]. The model training is performed on KNN, RF, XGBoost, and Feed Forward Neural Network (FFNN) with hyperparameter tuning and 10-fold cross validation techniques to improve detection accuracy. The results show XGBoost effectiveness and efficiency for real time DDoS detection in SDN-IoT environments. However, the study is limited by its reliance on a static model and the absence of adaptive optimization techniques for fine tuning detection parameters. Additionally, the lack of meta learning and the limited diversity in ML and DL model selection reduce the framework's adaptability to evolving attack patterns. The CICIDS2017 dataset is also used in [18] to develop an Intrusion Detection System (IDS) for DDoS attack detection. The approach includes NB, RF, and SVM for ML models as well as MLP and Tree-CNN for DL models with three activation functions and PCA to reduce the dimensionality of the data. However, the study does not incorporate broader AI models from ML and DL classifiers, including unsupervised algorithms. It lacks adaptive model selection through meta learning and omits parameter fine-tuning strategies that could improve detection performance under diverse and dynamic traffic conditions. In [19], the authors highlight that despite its significance, prior research on DDoS attack detection in SDN environments has yielded limited results. To address this, they employ multiple public datasets with DL models, including CNN, LSTM, CNN-LSTM, SVC-SOM, and SAE-MLP. Among these, SAE-MLP results demonstrate the highest accuracy at 99.75%. While the study emphasizes dataset diversity, it does not provide the comparative performance of ML models, meta learning approaches for dynamic adaptation, and optimization strategies that could further refine detection capabilities.

Although many studies address DDoS mitigation, the problem remains challenging in SDN environments [20]. The referenced work proposes an IDS using several ML models, J48, RF, REP Tree, SVM, and RF combined with MLP, trained and evaluated on the CIC DoS dataset. The MLP-based approach achieves the best performance among the tested models. However, the study does not examine other deep learning architectures or explore optimization and meta learning techniques for adaptive tuning, limiting its applicability in more dynamic or evolving threat scenarios. The centralized control logic in SDN is identified as a prime target for malicious activities, particularly DDoS attacks [21]. To mitigate this vulnerability, a defense mechanism is proposed using

a Generative Adversarial Network (GAN) framework, trained and evaluated on the CICDDoS2019 dataset. Through the experimental results, they mention that the proposed system detects DDoS attacks compared to other approaches. However, the study focuses solely on a single DL model and overlooks the potential contributions of conventional ML models or meta-learning techniques in attack detection. Additionally, the lack of fine-tuning strategies restricts the model's robustness in dynamic environments. The centralized architecture of SDN is particularly susceptible to frequent and large scale DDoS attacks, especially targeting the control layer [22]. To mitigate this vulnerability, MLP and CNN models are trained using the CICDDoS-2019 and InSDN datasets, with SHAP-based feature selection to improve detection accuracy. The training process employs Bayesian and ADAM optimizers. The approach excludes comparative analysis with ML techniques, which could provide valuable context for interpreting the results. Moreover, it does not assess how other DL methods might perform under similar conditions. This narrow focus limits the comparative depth and scalability of the proposed solution. Anomaly-based DDoS detection in SDN is explored through the application of SVM, KNN, DT, MLP, and CNN models for identifying traffic patterns [23]. The findings reveal that CNN achieves the highest training accuracy at 97.808%, while SVM demonstrates superior generalization with a prediction accuracy of 95.5%. However, the study ultimately centers its analysis on CNN and MLP, without extending the investigation to other DL architectures. It also lacks comparative benchmarking against ML approaches and omits the use of advanced optimization methods, feature selection techniques, and parameter tuning, which are essential for enhancing detection precision and computational efficiency in dynamic network conditions.

A modular SDN-based architecture is proposed for detecting transport and application-layer DDoS attacks, utilizing three ML models, including SVM, RF, and KNN, alongside four DL models, including MLP, CNN, GRU, and LSTM [24]. Evaluated on the CICDoS2017 and CICDDoS2019 datasets, the DL models achieved better performance. However, the study does not implement tuning methods that are crucial for boosting detection precision and computational efficiency. Moreover, it overlooks the use of unsupervised learning and meta-learning techniques that could improve responsiveness and model agility. DDoS attacks are recognized as significant threats to modern networks, necessitating robust and precise detection strategies [25]. In response, a feature selection-based intrusion detection approach is introduced, employing a CNN model to classify DDoS traffic effectively. Addressing challenges like data imbalance and computational complexity, the model is trained and evaluated on the CICDDoS2019 dataset. The results confirm that the proposed method, enhanced by effective preprocessing and feature selection, surpasses existing DDoS detection techniques in both accuracy and efficiency. However, the approach does not assess the performance of ML models, nor does it incorporate other DL frameworks. The absence of meta-learning further limits the system's ability to dynamically adjust to evolving network behaviors and attack patterns.

DDoS attacks are recognized as a major security concern, posing significant risks to the availability and reliability of network services [26]. To address this issue, an optimized SDN based detection framework is introduced, utilizing a one-dimensional Convolutional Neural Network (1D-CNN). The 1D-CNN is trained on labeled traffic data and fine-tuned using the NSGA-II algorithm to be compared against ML models, including LR, RF, SVM, and KNN. The approach demonstrates the potential of deep learning for advanced cybersecurity defense. While the proposed framework showcases the strengths of 1D-CNN, its dependence on a single DL architecture narrows its adaptability to diverse traffic scenarios and evolving attack signatures. Furthermore, it does not incorporate unsupervised learning methods or meta learning strategies, which could have enriched the system's ability to generalize across unseen patterns and dynamically adjust to shifting network conditions. A detailed taxonomy of DDoS defense mechanisms in SDN is presented, emphasizing the role of ML and DL techniques [27]. It highlights the importance of feature selection algorithms and publicly available DDoS datasets, while advocating for the creation of SDN-specific datasets to improve detection accuracy. Key research challenges are outlined to guide future advancements in securing SDN against evolving

DDoS threats. While the referenced study offers a valuable survey of ML and DL-based DDoS defense mechanisms in SDN, it does not provide an experimental evaluation and quantitative comparison of the proposed frameworks, limiting its practical applicability and benchmarking depth. Authors in [28] propose a DL-based DDoS detector for SDN using a hybrid DNN-LSTM model trained on an optimized CIC-DDoS2019 dataset. With added attention and transformer components, the model reaches about 99% validation accuracy and 98.84% testing accuracy. However, the study focuses on a single hybrid architecture and does not incorporate broader ML–DL model diversity. [29] Addresses DDoS threats in SDN-IoT networks by training ML models, KNN, RF, XGBoost, and FFNN on the CICIDS2017 and Edge-IIoTset datasets, using hyperparameter tuning and cross validation to improve accuracy. To reduce latency, the best model (XGBoost) is deployed at the network edge, achieving over 99.997% accuracy in binary detection and fast inference on live traffic. However, the study focuses primarily on ML-based detection and does not explore deeper DL architectures, hybrid designs, or multi-scenario evaluation.

These efforts highlight the complexity of securing SDN infrastructures against evolving threats and underscore the need for adaptive, intelligent solutions that can operate effectively in dynamic network conditions. While the studies have explored machine learning and deep learning techniques for DDoS detection in SDN environments, many existing solutions face limitations in adaptability, model diversity, and optimization. Most solutions rely on static models and lack mechanisms for selecting from a broad spectrum of models, including supervised, unsupervised, reinforcement learning, meta-learning, and deep learning, thereby constraining their responsiveness to diverse threat scenarios. Furthermore, intelligent optimization strategies are frequently neglected, resulting in reduced detection performance and elevated computational overhead. A notable gap also persists in evaluating the impact of optimization by comparing model performance before and after parameter tuning, which limits a comprehensive understanding of its effectiveness. In contrast, the proposed framework introduces a diverse AI-driven framework that integrates a heterogeneous ensemble of DL and ML models, including supervised, unsupervised, reinforcement learning, and meta learning, to support adaptive and context aware model selection. It further employs a hybrid Particle Swarm Optimization Genetic Algorithm (PSO-GA) approach to fine-tune detection parameters to enhance detection accuracy and computational efficiency.

3. Methodology

This work aims to develop an intelligent comparative framework for the detection and mitigation of a diverse array of DDoS attacks in SDN environments. The framework leverages a diverse set of DL and ML models, including meta-learning, adaptive RL, supervised learning, and unsupervised clustering, to ensure robust and adaptive DDoS detection across a wide range of traffic patterns and attack scenarios. The proposed architecture is structured into three main stages: data collection & preprocessing, model development & training, and model optimization & evaluation, as shown in Fig. 1.

3.1 Data Collection & Preprocessing

The experiments conducted in this work utilize SDN-DDoS-Traffic-Dataset.csv [30], specifically designed to address limitations found in publicly available datasets. To simulate realistic network conditions, two distinct SDN topologies are introduced for representing normal operation and incorporating DDoS attack scenarios. Each topology consists of 12 switches and 24 hosts managed by a Ryu controller. The dataset captures both benign and malicious traffic, providing a comprehensive environment for evaluating the proposed framework's detection and mitigation capabilities. With over 1,048,575 records, the dataset can capture a wide range of traffic behaviors and patterns. This setup enables covering a diverse array of DDoS attack traffic scenarios and allows models to be trained to recognize complex differences between normal traffic and advanced attacks. To ensure data quality and prevent model bias, a comprehensive preprocessing pipeline is applied. All preprocessing steps are executed strictly within the training folds during 10-fold cross-validation to prevent data leakage and ensure that no information from the test folds influences model training.

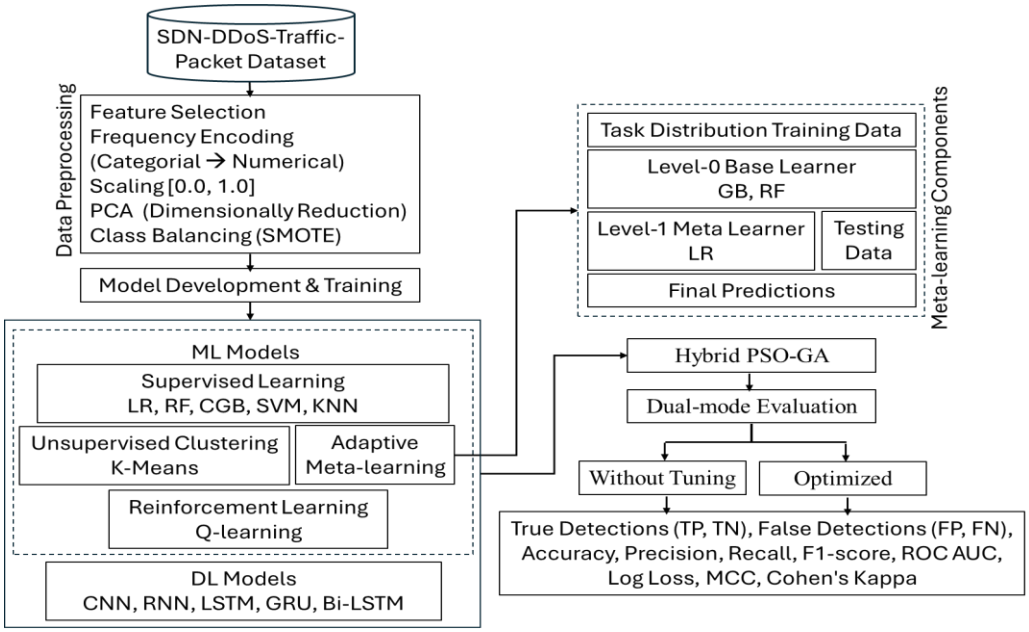

Fig. 1: Process flow diagram

3.1.1 Categorical Encoding

Categorical attributes are transformed using frequency encoding, where each category is replaced by its relative frequency in the training data. This approach avoids the dimensionality explosion associated with one hot encoding while preserving the statistical distribution of categorical values. After encoding, the original categorical columns are removed to eliminate redundancy, as shown in the following code snippet.

```
# Frequency Encoding for categorical columns
categorical_cols = df.select_dtypes(include="object").columns
for col in categorical_cols:
    frequency_map = df[col].value_counts().to_dict()
    df[f"{col}_frequency"] = df[col].map(frequency_map)
    df.drop([col], axis=1, inplace=True)
```

3.1.2 Feature Scaling

Because raw features vary significantly in magnitude, we apply StandardScaler to normalize each feature to have a mean of 0 and a standard deviation of 1 (within the range of [0.0, 1.0]). Scaling is performed using statistics computed only from the training fold, ensuring that no information from the test fold is introduced during normalization. This step ensures that all features contribute proportionally during model training.

3.1.3 Feature Selection

We begin by removing redundant, constant, or semantically irrelevant features. This step reduces noise, lowers computational overhead, and mitigates the risk of overfitting. In this context, as illustrated in the following code snippet, the selected features span multiple layers of the network stack to capture structural, temporal, and behavioral characteristics of both benign and attack traffic.

```
relevant_columns = [
    'host, 'src_ip', 'dst_ip', 'pkt_count', 'byte_count',
    'duration_nsec', 'tot_duration', 'flows', 'packet_per_massg',
    'pktper_flow', 'byte_per_flow', 'pkt_rate', 'pair_flow', 'Protocol', 'port_no']
```

3.1.4 Dimensionality Reduction

To further reduce computational complexity and highlight the most informative structure in the data, Principal Component Analysis (PCA) is applied. PCA is fit on the training fold and then applied to the corresponding test fold. In this work, the feature space is reduced to two principal components *(PCA(n_components=2))*, which capture the majority of the variance while enabling efficient model training and visualization. Fig. 2 shows the process.

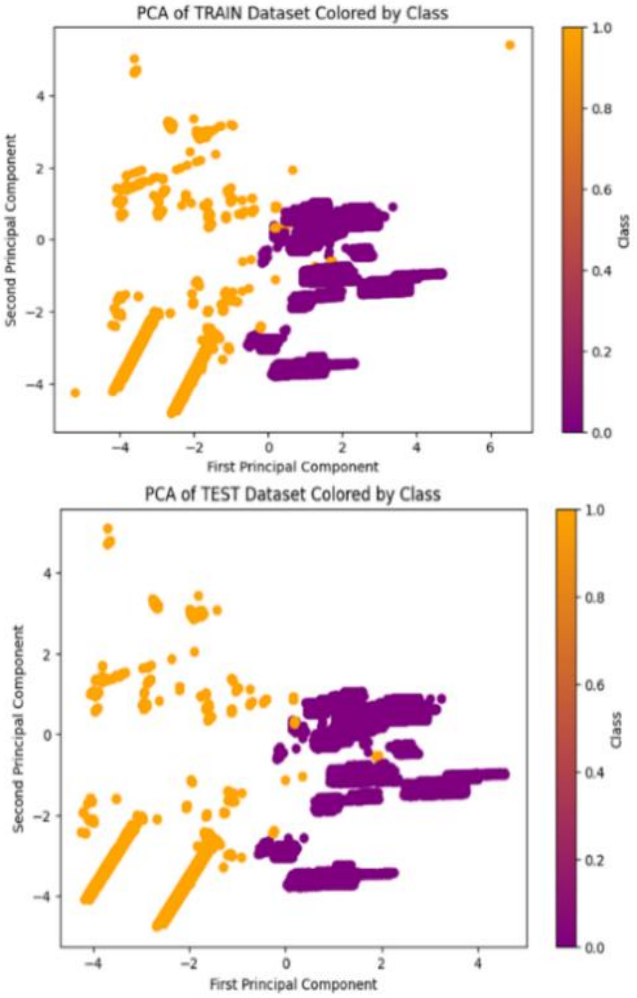

Fig. 2: PCA Features

3.1.5 Class Balancing

The dataset exhibits class imbalance, with benign flows significantly outnumbering malicious ones. To address this, we apply the Synthetic Minority Oversampling Technique (SMOTE) with *n_neighbors = 5* and *sampling_strategy = 'auto'*. SMOTE is applied only to the training portion of each fold, ensuring that synthetic samples do not leak into the evaluation set. This step improves model generalization and reduces bias toward the majority class. After completing all preprocessing steps within each fold, the data is used for model training and evaluation under the 10-fold cross-validation framework. This leakage-free pipeline ensures that the reported results are statistically reliable, reproducible, and reflective of real-world performance.

3.2 Model Selection

To detect DDoS security threats in SDN systems, we develop a multi-layer comparative framework that integrates a diverse set of DL and ML models as meta-learning, adaptive RL, supervised learning, and unsupervised clustering. Each category contributes distinct strengths to the defense strategy. To capture complex, non-linear, and temporal patterns inherent in SDN traffic, the framework integrates DL models, including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), The ML component incorporates Logistic Regression (LR), Random Forest (RF), XGBoost (XGB), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) to provide fast, interpretable, and computationally efficient detection across structured traffic features. K-Means clustering is used to identify unseen or emerging attack behaviors without labeled data. For adaptive decision-making, Q-Learning is included to support dynamic response strategies in evolving SDN environments. The framework further includes a meta-learning model with adaptive capability as one of its primary applications. In dynamic environments, tasks can change over time. The meta-learning model adapts to these changes by dynamically changing its structure and connections based on the input data. In addition to task distribution, the meta-model architecture comprises two key components: a base learner and a meta-learner. The base learners are responsible for learning from limited data on specific tasks and are typically diverse to capture a wide range of patterns in the input. The meta-learner, on the other hand, is trained to rapidly adapt the base learners by learning from their predictions rather than raw data. As a result, simpler models often perform more effectively in meta-learning due to their efficiency and generalization capabilities. For meta-learning implementation, the framework employs a two-layer stacking ensemble to improve attack classification performance. For a level-0 base learner, the framework includes Gradient Boosting Classifier and Random Forest Classifier due to their robustness and flexibility. The Gradient Boosting Classifier sequentially refines its predictions by fitting learners to the residuals of prior iterations, thereby capturing complex nonlinear relationships and interactions among features. The

random forest classifier, by contrast, reduces overfitting and provides generalization by averaging multiple decision trees via bootstrap sampling. The combination of these two heterogeneous base learners is intended to provide complementary decision boundaries and error patterns, expanding the representational capacity available to the stacking process. For the level-1 meta-learner, Logistic Regression is chosen. It serves as a simple, interpretable combiner that can weigh the probabilistic outputs of the base learners. Then, stacking is used as an ensemble technique to train the level-0 base learners independently and feed their outputs as input features for the level-1 meta learner to make the final output.

3.3 Model Optimization & Evaluation

The inclusion of a diverse set of learning models within the framework results in a high-dimensional hyperparameter space, presenting a significant challenge for effective tuning. To address this, a two-stage hybrid optimization approach, PSO-GA, is introduced, combining PSO and GA to efficiently search and refine hyperparameters across the framework. To initiate the PSO-GA optimization process, hyperparameter configurations are defined for all the learning models in the framework. These configurations specify bounded search spaces (lower bound (lb) and upper bound (lu)) as well as baseline values before optimization. Then, PSO is employed to explore the search space by iteratively adjusting particle positions toward better-performing regions based on both individual and collective experience. As particles update their positions, the algorithm converges toward optimal settings that enhance model performance. The final output is a set of tuned parameters specific to each model's characteristics and detection goals for subsequent optimization via the GA stage. GA is applied to further refine hyperparameters by evolving a population of candidate solutions over generations. Each individual represents a set of parameters for a specific model in the framework. It evaluates each individual's fitness based on model accuracy, aiming to minimize error and discover high performing configurations without exhaustive search. Tables. 1, 2, and 3 summarize the hyperparameter settings for DL, ML, and meta-learning models, respectively, which serve as the foundation for the swarm based search and genetic refinement stages of the PSO-GA procedure. They summarize the hyperparameter settings for DL, ML, and meta learning models, respectively, for the PSO-GA procedure.

Table. 1: DL hyperparameter settings with and without PSO-GA

Model name	Hyperparameter & Baseline Values	PSO-GA bounds
CNN	Filters: filters=64	lb=[16], lu=[256]
CNN RNN LSTM GRU Bi-LSTM	Learning Rate: learning_rate=0.001	lb=[0.0001], lu=[0.01]
	Hidden Units: hidden_units=256	lb=[32], lu=[512]
	Dropout Rate: dropout_rate=0.3	lb=[0.1], lu=[0.5]
	Batch Size: batch_size=32	lb=[16], lu=[128]
	Epochs: epochs=50	lb=[10], lu=[200]

Table. 2: ML hyperparameter settings with and without PSO-GA

Model name	Hyperparameter & Baseline Values	PSO-GA bounds
LR	Regularization Strength: C=1.0	lb=[0.01], lu=[10]
	Solver: solver='lbfgs'	lb=[0], lu=[3]
RF	Number of Trees: n_estimators=100	lb=[10], lu=[200]
	Max Depth: max_depth=10	lb=[2], lu=[50]
XGBoost	Learning Rate: eta=0.1	lb=[0.01], lu=[0.3]
	Number of Trees: n_estimators=100	lb=[50], lu=[500]
	Max Depth: max_depth=6	lb=[3], lu=[15]
SVM	Regularization Strength: C=1.0	lb=[0.01], lu=[100]
	Kernel Coefficient: gamma='scale'	lb=[0.0001], lu=[1.0]
	Kernel Type: kernel='rbf'	lb=[0], lu=[3]
KNN	Number of Neighbors: n_neighbors=5	lb=[1], lu=[30]
	Distance Metric: metric='euclidean'	lb=[0], lu=[2]
	Weight Function: weights='uniform'	lb=[0], lu=[1]
K-Means	Number of Clusters: n_clusters=8	lb=[2], lu=[20]
	Max Iterations: max_iter=300	lb=[100], lu=[500]
	Initialization Method: init='k-means++'	lb=[0], lu=[1]

Q-Learning	Learning Rate: alpha=0.1	lb=[0.01], lu=[1.0]
	Discount Factor: gamma=0.9	lb=[0.5], lu=[0.99]
	Exploration Rate: epsilon=0.9	lb=[0.01], lu=[1.0]

Table. 3: Meta-learning hyperparameter settings with and without PSO-GA

Component	Hyperparameter & Baseline Values	PSO-GA bounds
GB (L0)	Number of Estimators: n_estimators=100	lb=[50], lu=[500]
	Learning Rate: learning_rate=0.1	lb=[0.01], lu=[0.3]
	Max Depth: max_depth=3	lb=[2], lu=[10]
	Subsample: subsample=1.0	lb=[0.5], lu=[1.0]
RF (L0)	Number of Estimators: n_estimators=100	lb=[10], lu=[200]
	Max Depth: max_depth=10	lb=[2], lu=[50]
	Min Samples Split: min_samples_split=2	lb=[2], lu=[10]
LR (L1)	Regularization Strength: C=1.0	lb=[0.01], lu=[10]
	Solver: solver='lbfgs'	lb=[0], lu=[3]

An extensive set of experiments is conducted to implement the models and evaluate their performance using multiple performance metrics. The framework provides a dual-mode evaluation with two primary objectives: first, to identify the most effective model within the framework for traffic classification and DDoS threat detection in SDN environments, and second, to quantify the benefits introduced by the hybrid optimization approach. This is achieved by comparing model performance with untuned hyperparameters against results obtained using the PSO-GA optimization. The evaluation metrics include the confusion matrix in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), Accuracy, Precision, Recall, and F1-score.

4. Results and Discussions

This section presents the results from implementing the models within the proposed framework. It aims to evaluate the strengths and limitations of each model in detecting DDoS attacks and identify the most effective solutions for ensuring robust protection in SDN environments.

4.1 True Detections

TP and TN are essential indicators of the ability to correctly identify DDoS attack traffic and benign flows, respectively. A high TP count reflects the model's effectiveness in detecting actual threats, while a high TN count demonstrates the model's effectiveness in recognizing legitimate traffic without triggering false alarms. Collectively, these metrics provide insight into the model's operational reliability and its effectiveness in preserving network integrity under attack conditions. Fig. 3 illustrates the TP and TN outcomes for each model, representing correct identifications of traffic.

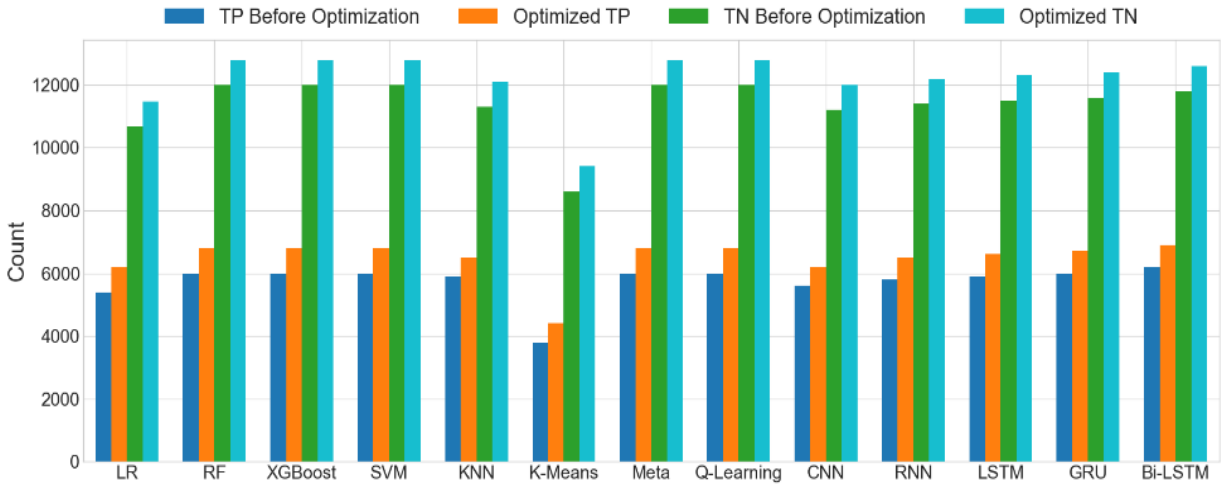

Fig. 3: Comparison of correctly classified attack and normal traffic for each model

The results show that PSO-GA optimization led to a consistent increase in both TP and TN values for all models. The ensemble-based models, including RF, XGBoost, SVM, Meta Learner, and Q-Learning, obtain the same improvement level, with their TP and TN values increasing accordingly. These consistent gains suggest that PSO-GA effectively refined decision boundaries, enabling more accurate identification of both malicious and legitimate traffic. KNN followed with clear improvements in both TP and TN, indicating better responsiveness to local data structures with fewer misclassifications in both directions. LR also enhances performance, showing noticeable TP and TN improvements; despite its simpler design, the model benefited from the optimized approach, which helped it better distinguish between classes. K-Means, despite its unsupervised nature, showed a notable increase in both TP and TN, suggesting improved clustering behavior and more accurate separation of traffic types even without labeled guidance. Among neural architectures, Bi-LSTM achieved the highest post-optimization TP and TN, reflecting its superior ability to capture bidirectional temporal patterns and make precise predictions. GRU followed with strong gains, demonstrating that its simplified architecture adapts well to tuning and enhances classification reliability. LSTM also improved in both metrics, with optimization helping it retain relevant features more effectively and reduce misclassifications. RNN showed clear TP and TN increases, indicating that even simpler recurrent structures respond well to learning rate and sequence length adjustments. CNN also exhibited TP and TN enhancements; although more focused on spatial features, its performance in traffic classification improved through filter and layer tuning.

4.2 False Detections

FP and FN are critical indicators of a model's error behavior. FPs occur when benign traffic is incorrectly flagged as malicious, potentially leading to unnecessary disruptions in legitimate network activity. FNs, on the other hand, represent missed detections of actual attacks, which can leave the system vulnerable. Minimizing both types of errors is essential for maintaining service continuity in SDN environments. FP and FN results are illustrated in Fig. 4, showing misclassification rates of each model. The results show that the PSO-GA approach reduced FP and FN errors across all models. Among the ML models, Logistic Regression showed a substantial drop in both error types, indicating that optimization helped it better align its linear decision boundary. KNN also achieved notable reductions in FP and FN, reflecting improved neighborhood tuning despite its sensitivity to local noise. The RF, XGBoost, SVM, Meta-Learner, and Q-Learning models all converged to the same post-optimization error level, demonstrating strong generalization and balanced decision-making under PSO-GA.

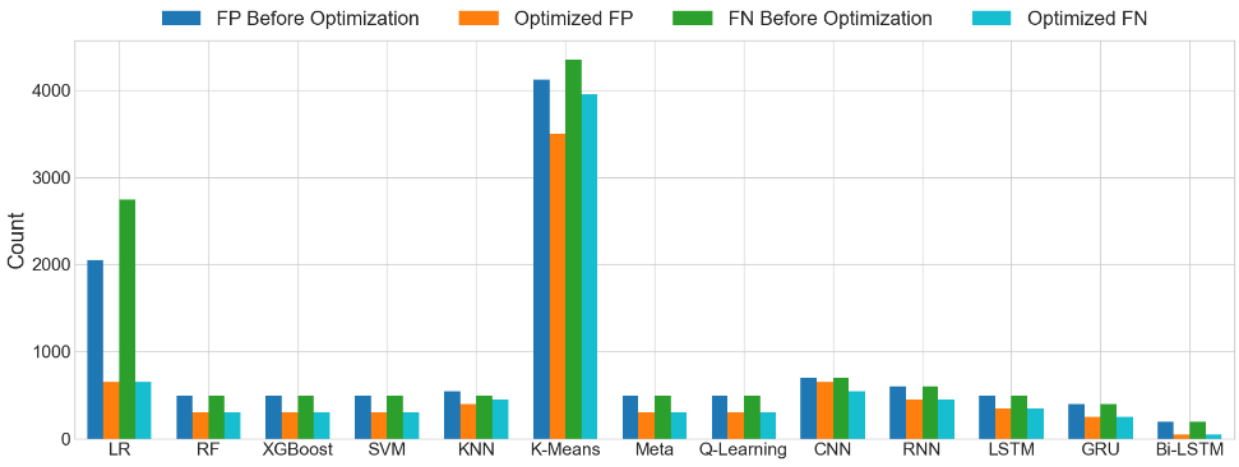

Fig. 4: Misclassification rates of attack and normal traffic in the framework

The K-Means clustering model retained the highest error count overall, but still showed meaningful reductions, indicating that even unsupervised clustering benefits from careful parameter tuning. In the deep learning group, Bi-LSTM achieved the lowest FP and FN after optimization, highlighting its strong ability to distinguish between normal and malicious traffic. GRU followed with clear improvements, supported by its efficient gating mechanism for temporal learning. LSTM also reduced both error types, with optimization helping it retain relevant patterns more effectively. RNN showed noticeable reductions as well, benefiting from adjustments to learning rate and sequence length. CNN likewise lowered its FP and FN values, with filter and layer tuning enhancing its spatial feature extraction.

4.3 Accuracy

Accuracy reflects the proportion of correctly classified malicious and benign traffic instances out of the total samples. It serves as a general indicator of a model's overall effectiveness in distinguishing between attack and normal flows. It doesn't account for class imbalance or the severity of misclassifications, but it is a useful benchmark for comparing model performance under consistent conditions. Accuracy results for all the models are displayed in Fig. 5. The experimental results clearly demonstrate that the hybrid PSO-GA approach enhances model performance across all categories. Beyond the improvement, the comparative analysis shows that among the DL models, Bi-LSTM reached the highest accuracy, highlighting its strength in capturing bidirectional temporal dependencies. The optimization process enabled it to fully leverage its architecture to distinguish subtle traffic patterns. GRU followed closely, with its efficient gating structure allowing it to maintain high accuracy with fewer parameters. LSTM also showed substantial improvement, benefiting from enhanced memory retention and dropout tuning that reduced overfitting. RNN increased notably as well, while CNN provided the lowest accuracy among the DL models, though it still showed meaningful gains after optimization. In the ML category, RF, XGBoost, SVM, the Meta-Learner, and Q-Learning all achieved top-level accuracy after optimization, demonstrating PSO-GA's ability to fine-tune ensemble depth, margin constraints, and learning strategies for consistently correct predictions across all classes. K-Means, although obtaining the lowest performance among all models, still demonstrated measurable improvement after optimization, showing that even unsupervised clustering benefits from refined parameter tuning.

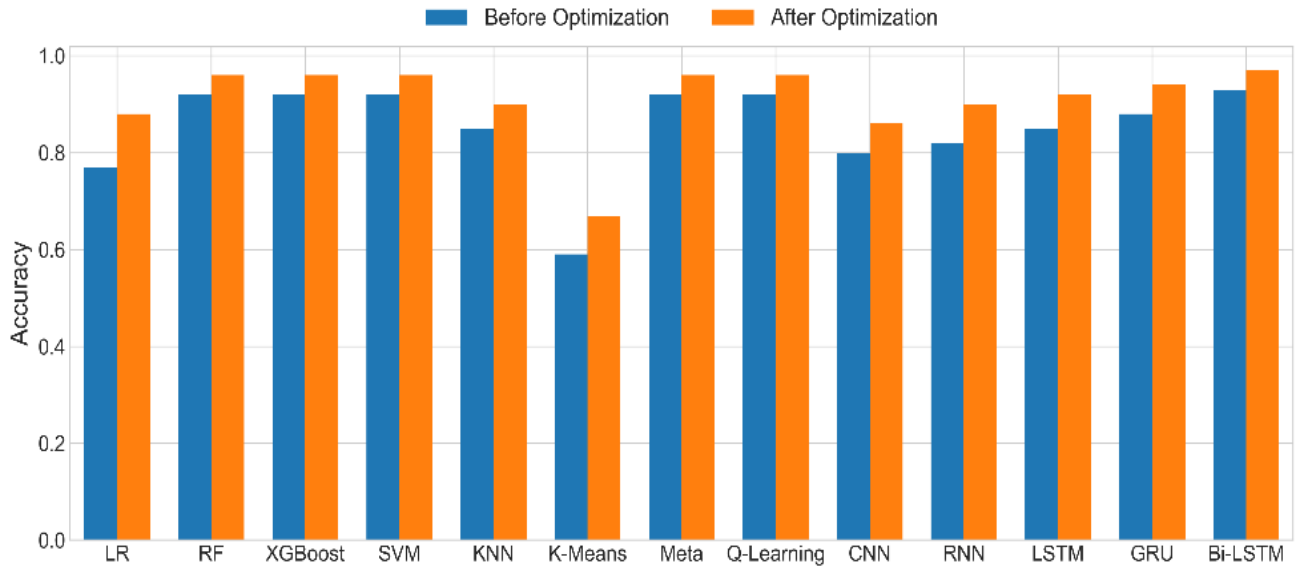

Fig. 5: Accuracy comparison for the detection models

4.4 Precision

Precision is important in evaluating how well a model identifies malicious traffic without misclassifying benign flows. High precision indicates that the model produces fewer false positives, which is crucial in network security to avoid unnecessary mitigation actions that could disrupt legitimate services. This is especially important when the cost of misclassifying normal traffic as an attack is high, as it directly affects the reliability and trust of the detection system. Precision scores are displayed in Fig. 6. The results show that the PSO-GA optimization consistently improves precision across all model categories. Among the ML models, RF, XGBoost, SVM, the Meta-Learner, and Q-Learning all reached perfect precision after optimization. These gains stem from the optimizer's ability to tune parameters such as tree depth, learning rate, kernel settings, ensemble weighting, and exploration strategies, allowing each model to sharpen its decision boundaries and eliminate false positives. KNN also improved its precision through optimized neighbor selection and distance weighting, though it remains sensitive to local noise. Logistic Regression showed clear improvement as well, benefiting from better regularization and solver adjustments, though its linear nature still limits its ability to capture complex traffic patterns. K-Means demonstrated a noticeable precision increase due to improved cluster initialization, despite the inherent limitations of unsupervised learning. In the DL category, Bi-LSTM achieved perfect precision, outperforming other architectures due to its ability to process sequences bidirectionally and capture richer temporal dependencies. GRU also showed strong improvement, supported by its efficient gating mechanism. LSTM followed with substantial gains through enhanced memory-cell tuning. RNN improved through optimized learning rate and dropout settings, though its lack of gating still restricts its performance. CNN exhibited the smallest improvement in this group, as its spatial-feature focus makes it less effective for sequential traffic patterns compared to recurrent models.

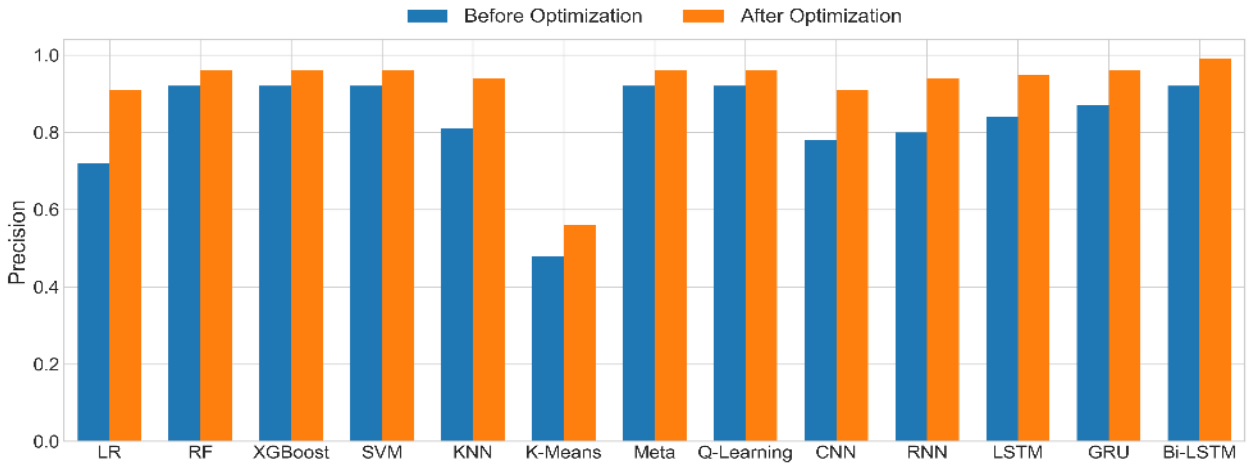

Fig. 6: Precision scores as the model's confidence in identifying attack traffic

4.5 Recall

Recall measures the models' ability to correctly identify all actual attack instances. High recall indicates that the model is effective at minimizing false negatives, an essential quality in security, where failing to detect an attack can cause severe consequences. Especially in an SDN system, where traffic patterns are dynamic and threats can evolve rapidly, a model with strong recall ensures that malicious flows are not overlooked. Recall values are illustrated in Fig. 7. The results across the evaluated models show a clear effect of the hybrid PSO-GA approach on recall enhancement, while also revealing meaningful differences in each model's detection capacity. The RF, XGBoost, SVM, Meta-Learner, and Q-Learning achieved flawless recall after optimization.

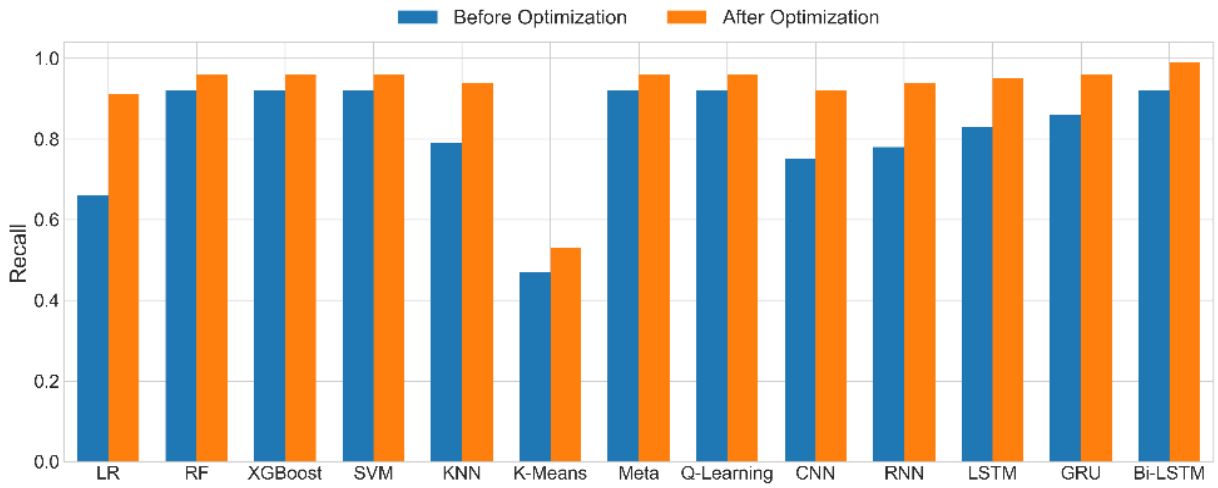

Fig. 7: Recall performance showing sensitivity of the models to malicious flows

These algorithms, known for their robustness and adaptability, responded well to refined hyperparameters that strengthened their ability to detect subtle and complex attack patterns without missing relevant instances. KNN also showed a noticeable improvement, indicating better responsiveness to local data variations, although its performance remains influenced by noise and density in the feature space. Logistic Regression demonstrated a clear recall increase as well; despite its simpler linear structure, optimization helped it capture more relevant patterns through improved regularization and solver adjustments. K-Means likewise improved its recall, showing that even without labeled data, optimization enhanced its ability to group anomalous traffic more effectively and reduce false negatives. Among deep learning architectures, Bi-LSTM stood out with perfect recall. Its bidirectional structure enables comprehensive sequence understanding, and optimization further amplifies its ability to detect relevant temporal patterns. GRU and LSTM followed with strong improvements, benefiting from fine-tuned gating mechanisms and memory configurations that enhanced their ability to retain and utilize long-term dependencies. RNN also achieved a substantial recall gain, reflecting improved learning dynamics despite its simpler architecture. CNN, although not inherently designed for sequential data, still showed a meaningful recall increase due to improved filter and learning-rate settings, though its recall remains lower than that of recurrent models because of its limited temporal modeling capability.

4.6 F1-score

F1-Score shows a balanced view of threat detection and false alarm control. A high F1-score shows the model is ideal where both missed threats and excessive alerts carry serious consequences. A low F1-score suggests the model is either missing attacks or generating too many false positives, undermining its practical utility. The F1-score results are provided in Fig. 8. The F1-score improvements across all models underscore the versatility and effectiveness of PSO-GA as a hyperparameter optimization approach. The RF, XGBoost, SVM, Meta-Learner, and Q-Learning models all achieved perfect F1-scores after optimization, highlighting how PSO-GA's adaptive tuning complements the structural strengths of each model and enables precise DDoS detection. KNN also showed a clear improvement, reflecting better neighborhood selection and weighting, though it remains sensitive to local noise. Logistic Regression demonstrated noticeable gains as well; despite its simple linear form, optimization enhanced its ability to separate classes in high dimensional space. K-Means clustering likewise improved, indicating that refined cluster initialization and boundary adjustments strengthened its ability to distinguish between normal and anomalous traffic. In the DL group, Bi-LSTM again achieved a perfect result, confirming its strong capability in modeling bidirectional sequences and capturing nuanced temporal dependencies.

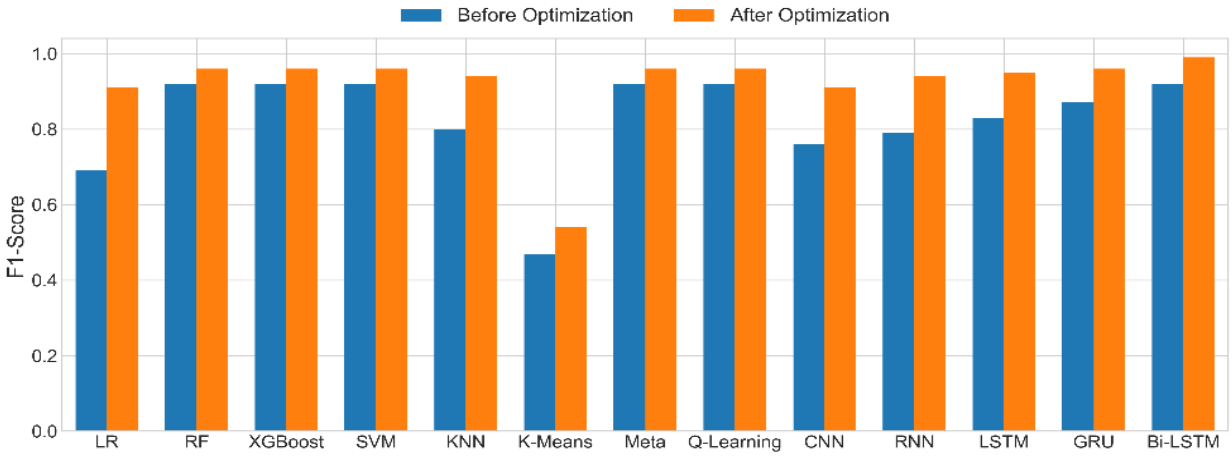

Fig. 8: F1-score distribution

GRU followed with a substantial improvement, benefiting from its streamlined gating mechanism and responsiveness to hyperparameter adjustments. LSTM also improved meaningfully through enhanced memory cell dynamics and dropout calibration, helping it retain relevant patterns while reducing overfitting. RNN showed a solid gain as well, responding effectively to learning rate and batch size tuning despite its simpler architecture. CNN, while yielding the lowest performance among the DL models, still demonstrated a clear F1-score improvement after PSO-GA optimization, with enhanced spatial feature extraction through tuned convolutional filters and network depth.

4.7 Standard Deviation of Results

To evaluate the statistical reliability and stability of the models, the mean and standard deviation (SD) of all performance metrics were computed across the 10-fold cross validation procedure. Standard deviation quantifies the variability of each metric across folds and reflects how consistently a model performs under different data partitions. Low deviation indicates stable generalization, whereas higher values suggest sensitivity to specific folds or potential overfitting. Fig. 9 and Fig. 10 present the mean ± standard deviation for the confusion-matrix components (TP, TN, FP, FN) and the performance metrics (Accuracy, Precision, Recall, and F1-score), respectively. The 10-fold cross validation results show that all models exhibit highly stable behavior, with consistently small standard deviations relative to their mean values. This stability indicates reliable generalization across folds and confirms that the dataset partitions do not introduce instability or fold specific bias. The uniformly low deviations reflect the adequacy and balance of the dataset, as well as the consistency of feature distributions across partitions. Consequently, the learned decision boundaries remain largely unchanged, and the low SD values represent genuine model robustness rather than artifacts of the evaluation procedure. These findings demonstrate that the observed performance improvements are not driven by isolated fold anomalies but reflect consistent behavior across the entire dataset. Within the ML group, ensemble based methods and K-Means show the lowest variability, indicating highly stable decision boundaries across folds. LR also maintains low deviation, whereas margin-based and instance-based models exhibit comparatively higher fluctuation due to their sensitivity to local data characteristics. The Q-Learning model displays moderate variability, suggesting a generally stable policy with some dependence on fold specific state action distributions. The meta learning model shows higher deviation than the most stable ML and DL models, reflecting the influence of fold level variation on its aggregated predictions. Among deep learning architectures, recurrent models, particularly the simpler recurrent structure, demonstrate the lowest standard deviation overall, highlighting strong consistency in temporal pattern learning. Gated architectures such as LSTM and Bi-LSTM also maintain low variability, while other models show greater sensitivity to fold-level differences.

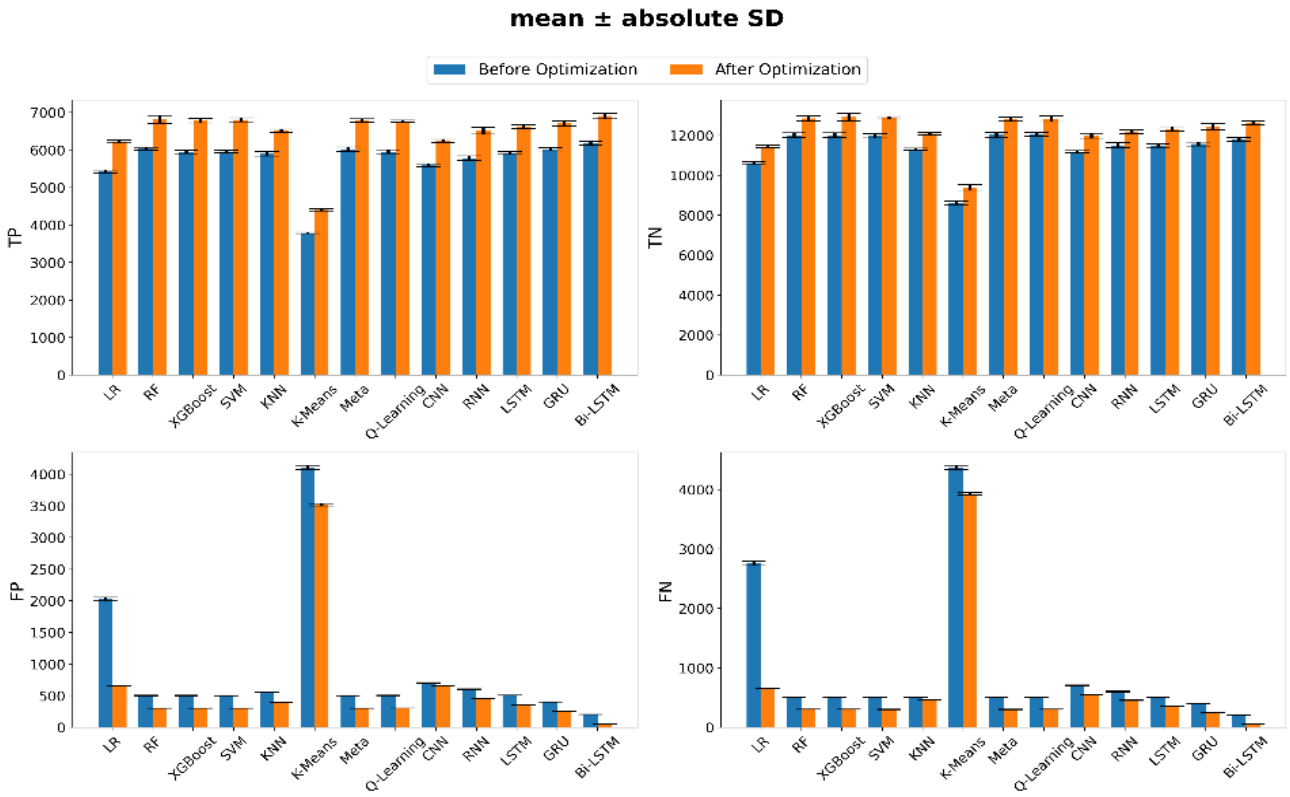

Fig. 9: Confusion matrix before and after optimization, with error bars showing 10-fold variability

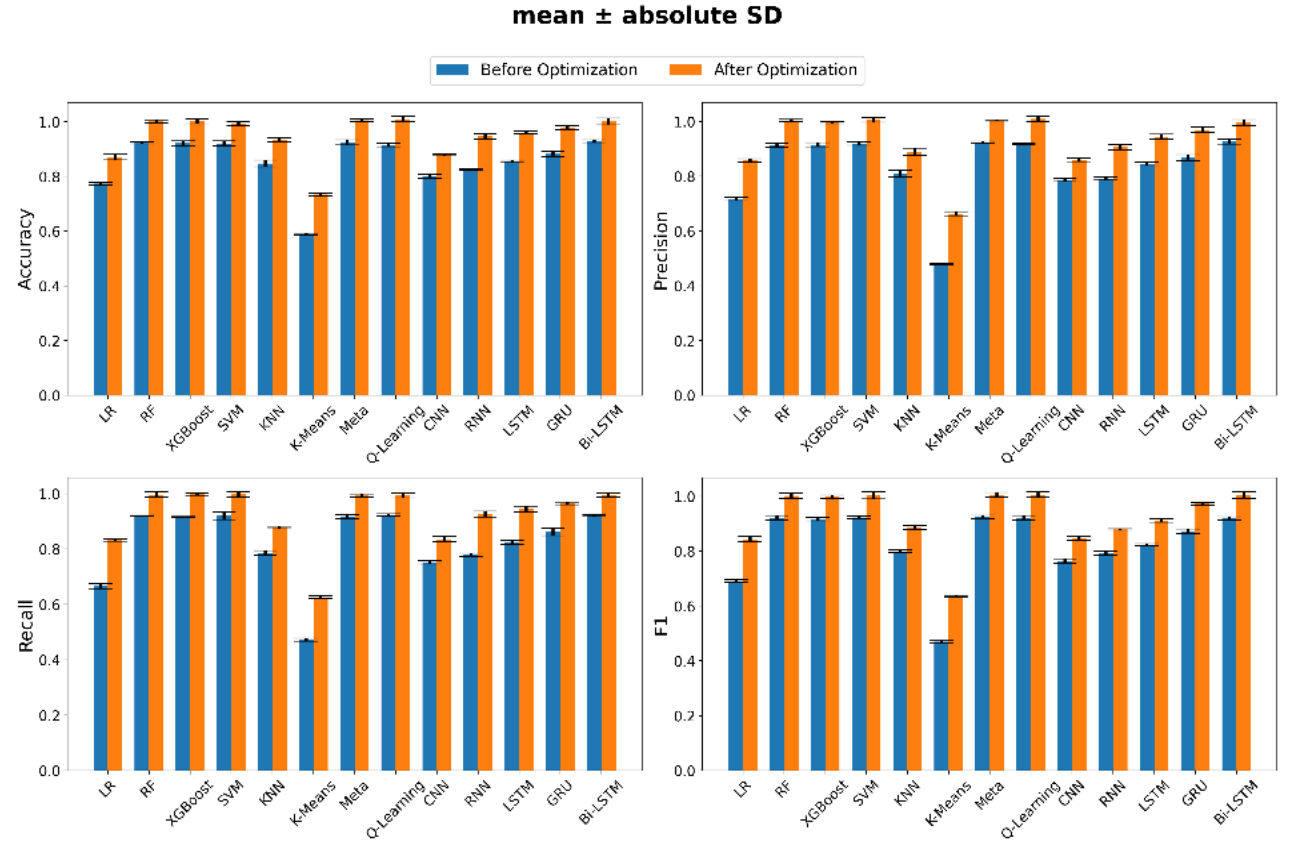

Fig. 10: Model performance with 10-fold cross-validation error bars

Collectively, these SD trends indicate that models benefiting from ensemble averaging or recurrent temporal modeling achieve the highest stability, whereas architectures with more complex or data dependent optimization dynamics exhibit greater variability across folds.

5. Conclusion

This work presents a multi-layer framework for DDoS detection in SDN environments, integrating deep learning, machine learning, meta-learning, reinforcement learning, and unsupervised clustering models. To further enhance performance, a hybrid PSO-GA optimization strategy is applied. The evaluation follows a dual-mode design, before and after optimization, to identify the most effective classifiers and quantify the gains introduced by PSO-GA. Across the ML models, ensemble-based approaches such as RF, XGBoost, and SVM consistently delivered strong performance, demonstrating reliable classification across all metrics. Meta-Learning and Q-Learning matched these results, showing strong adaptability to dynamic SDN traffic patterns. Among neural architectures, Bi-LSTM emerged as the most effective model, maintaining the lowest error rates and strongest predictive behavior. GRU and LSTM also performed well, benefiting from their ability to capture temporal dependencies in network flows. The impact of PSO-GA was evident across all model groups. Fine tuned hyperparameters significantly improved classification accuracy, reduced misclassifications, and strengthened predictive confidence. Notably, models with moderate baseline performance, such as Logistic Regression, KNN, and K-Means, also experienced substantial gains, demonstrating the optimizer's ability to elevate simpler or unsupervised methods. These improvements highlight the value of combining architectural strengths with adaptive optimization to build reliable and intelligent SDN-based DDoS detection systems. Overall, the most effective solutions were those that paired robust model architectures with PSO-GA's adaptive tuning. Rather than favoring a single model type, this work emphasizes model-specific evaluation and optimization. Future research may extend the PSO-GA hybrid approach to a broader family of reinforcement learning classifiers beyond Q-Learning.

Funding

This research received no external funding.

Conflict of Interest

The authors declared "No Conflict of Interest".

Data Availability Statement

The dataset is available at the following link: SDN-DDoS traffic dataset.
https://data.mendeley.com/datasets/b7vw628825/1

CRediT authorship contribution statement

Conceptualization, ASMA and MM; methodology, ASMA; software, ASMA; validation, ASMA and MM; formal analysis, ASMA; investigation, ASMA; resources, ASMA; data curation, ASMA; writing—original draft preparation, ASMA; writing—review and editing, MM; visualization, MM; supervision, MM; project administration, MM; funding acquisition, MM. All authors have read and agreed to the published version of the manuscript.

References

[1] H. Yu, H. Qi, and K. Li, "WECAN: an efficient west-east control associated network for large-scale SDN systems", *Mobile Networks and Applications*, Vol. 25, pp. 114-124, 2020. https://doi.org/10.1007/s11036-018-1194-9

[2] Z. Zhang and C. Wang, "Service function chain migration: A survey", *Computers*, Vol. 14, No. 6, art. no. 203, 2025. https://doi.org/10.3390/computers14060203

[3] Q. I. Ali and S.R. Awad, "Enhancing SDN Performance: Machine learning integration with the POX controller for dynamic routing and congestion management", *International Transactions on Electrical Engineering and Computer Science*, Vol. 4, No. 3, pp. 152-160, 2025. https://doi.org/10.62760/iteecs.4.3.2025.132

[4] R. Chaudhary, G. S. Aujla, N. Kumar, P. K. Chouhan, "A comprehensive survey on software defined networking for smart communities", *International Journal of*

Communication Systems, Vol. 38, No. 1, art. no. e5296, 2022.
https://doi.org/10.1002/dac.5296

[5] A. H. Abdi, L. Audah, A. Salh, M. A. Alhartomi, H. Rasheed, and S. Ahmed, "Security control and data planes of SDN: A comprehensive review of traditional, AI, and MTD approaches to security solutions", *IEEE Access*, Vol. 12, pp. 69941 – 69980, 2024.
https://doi.org/10.1109/ACCESS.2024.3393548

[6] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I.H. Hasbullah, A. A. Bahashwan, T. A. Al-Amiedy, and D.R. Ibrahim, "Effectiveness of an entropy based approach for detecting low and high rate DDoS attacks against the SDN controller: Experimental analysis", *Applied Sciences*, Vol. 13, No. 2, art. no. 775, 2023.
https://doi.org/10.3390/app13020775

[7] S. Batewela, M. Liyanage, E. Zeydan, M. Ylianttila, and P. Ranaweera, "Security orchestration in 5G and beyond smart network technologies", *IEEE Open Journal of the Computer Society*, Vol. 6, pp. 554-573, 2025.
https://doi.org/10.1109/OJCS.2025.3563619

[8] W. Hill, Y. T. Acquaah, J. Mason, D. Limbrick, S. Teixeira-Poit, C. Coates, and K. Roy, "DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies", *Discover Applied Sciences*, Vol. 6, art. no. 472, 2024.
https://doi.org/10.1007/s42452-024-06172-x

[9] A. Hirsi, M. A. Alhartomi, L. Audah, A. Salh, N. M. Sahar, and S. Ahmed, "Comprehensive analysis of DDoS anomaly detection in software defined networks", *IEEE Access*, Vol. 13, pp. 23013 – 23071, 2025.
https://doi.org/10.1109/ACCESS.2025.3535943

[10] A. A. Wabi, I. Idris, O. M. Olaniyi and J. A. Ojeniyi, "DDoS attack detection in SDN: Method of attacks, detection techniques, challenges and research gaps", *Computers & Security*, Vol. 139, art. no. 103652, 2024.
https://doi.org/10.1016/j.cose.2023.103652

[11] S. Soltani, A. Amanlou, M. Shojafar, and R. Tafazolli, "Security of topology discovery service in SDN: Vulnerabilities and countermeasures", *IEEE Open Journal of the Communications Society*, Vol. 5, pp. 3410 – 3450, 2024.
https://doi.org/10.1109/OJCOMS.2024.3406489

[12] M. A. Almedires, A. Elkhalil, and M. Amin, "Adversarial attack detection in industrial control systems using LSTM based intrusion detection and black-box defense strategies", *Journal of Cyber Security and Risk Auditing*, Vol. 2025, No. 3, 2025.
https://doi.org/10.63180/jcsra.thestap.2025.3.2

[13] A. Ali, "Adaptive and Context-Aware Authentication Framework Using Edge AI and Blockchain in Future Vehicular Networks", *STAP Journal of Security Risk Management*, Vol. 2024, No. 1, 2024.
https://doi.org/10.63180/jsrm.thestap.2024.1.3

[14] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software defined networks through feature selection methods and machine learning models", *Sustainability*, Vol. 12, No. 3, art. no. 1035, 2020.
https://doi.org/10.3390/su12031035

[15] M. W. Nadeem, H. G. Goh, V. Ponnusamy, and Y. Aun, "DDoS detection in SDN using machine learning techniques", *Computers, Materials & Continua*, Vol. 71, No. 1, pp. 771-789, 2022.
https://doi.org/10.32604/cmc.2022.021669

[16] K. M. Ko, J. M. Baek, B. S. Seo, and W. B. Lee, "Comparative study of AI enabled DDoS detection technologies in SDN", *Applied Sciences*, Vol. 13, No. 17, art. no. 9488, 2023.
https://doi.org/10.3390/app13179488

[17] H. M. Belachew, M. Y. Beyene, A. B. Desta, B. T. Alemu, S. S. Musa, and A. J. Muhammed, "Design a robust DDoS attack detection and mitigation scheme in SDN edge IoT by leveraging machine learning", *IEEE Access*, Vol. 13, pp. 10194 – 10214, 2025.
https://doi.org/10.1109/ACCESS.2025.3526692

[18] R. V. Mendonça, A. A. M. Teodoro, R. L. Rosa, M. Saadi, D. C. Melgarejo, and P. H. J. Nardelli, "Intrusion detection system based on fast hierarchical deep convolutional neural network", *IEEE Access*, Vol. 9, pp. 61024 – 61034, 2021.
https://doi.org/10.1109/ACCESS.2021.3074664

[19] N. Ahuja, D. Mukhopadhyay, and G. Singal, "DDoS attack traffic classification in SDN using deep learning", *Personal and Ubiquitous Computing*, Vol. 28, pp. 417-429, 2024.

https://doi.org/10.1007/s00779-023-01785-2

[20] J. A. P. Díaz, I. M. Valdovinos, K. K. R. Choo, and D. Zhu, "A flexible SDN based architecture for identifying and mitigating low-rate DDoS attacks using machine learning", *IEEE Access*, Vol. 8, pp. 155859-155872, 2020. https://doi.org/10.1109/ACCESS.2020.3019330

[21] M. P. Novaes, L. F. Carvalho, J. Lloret, M. L. P. Jr, "Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments", *Future Generation Computer Systems*, Vol. 125, pp. 156-167, 2021. https://doi.org/10.1016/j.future.2021.06.047

[22] S. Mehmood, R. Amin, J. Mustafa, M. Hussain, F. S. Alsubaei, and M. D. Zakaria, "Distributed denial of services (DDoS) attack detection in SDN using optimizer-equipped CNN-MLP", *PLOS One*, Vol. 20, No. 1, art. no. e0312425, 2025. https://doi.org/10.1371/journal.pone.0312425

[23] T. E. Ali, Y. W. Chong, and S. Manickam, "Comparison of ML/DL approaches for detecting DDoS attacks in SDN", *Applied Sciences*, Vol. 13, No. 5, art. no. 3033, 2023. https://doi.org/10.3390/app13053033

[24] N. M. Y. Naula, C. V. Rosales, and J. A. P. Diaz, "SDN based architecture for transport and application layer DDoS attack detection by using machine and deep learning", *IEEE Access*, Vol. 9, pp. 108495-108512, 2021. https://doi.org/10.1109/ACCESS.2021.3101650

[25] A. A. Najar and S. M. Naik, "A robust DDoS intrusion detection system using convolutional neural network", *Computers and Electrical Engineering*, Vol. 117, art. no. 109277, 2024. https://doi.org/10.1016/j.compeleceng.2024.109277

[26] Y. A. Dunainawi, B. R. A. Kaseem, and H.S.A. Raweshidy, "Optimized artificial intelligence model for DDoS detection in SDN environment", *IEEE Access*, Vol. 11, pp. 106733-106748, 2023. https://doi.org/10.1109/ACCESS.2023.3319214

[27] N. Aslam, S. Srivastava, and M. M. Gore, "A comprehensive analysis of machine learning- and deep learning based solutions for DDoS attack detection in SDN", *Arabian Journal for Science and Engineering*, Vol. 49, pp. 3533–3573, 2024. https://doi.org/10.1007/s13369-023-08075-2

[28] A. S. Zaidoun and Z. Lachiri, "A hybrid deep learning model for multi-class DDoS detection in SDN networks", *Annals of Telecommunications*, Vol. 80, pp. 459 - 472, 2025. https://doi.org/10.1007/s12243-025-01085-1

[29] A. K. Refai, "DDoS Attack Detection using Machine Learning Trained Models", *IEEE International Conference on Consumer Electronics*, pp. 1-6, 2026. https://doi.org/10.1109/ICCE67443.2026.11449845

[30] SDN-DDoS traffic dataset. https://data.mendeley.com/datasets/b7vw628825/1

***