Scientific Journal of Computer Science

futuristech
PT. Teknologi Futuristik Indonesia

Received: May 16, 2026; Revised: July 22, 2026; Accepted: July 23, 2026; Published: July 25, 2026; Current version: July 25, 2026. e-ISSN: 3110-3170
Digital Object Identifier 10.64539/sjcs.v2i2.2026.511

Article

A Lightweight Stacking Ensemble Intrusion Detection Framework for Software-Defined Networking Using the InSDN Dataset

Ubakaghinwa Paul Chigbu[1], Abdulrashid Abdulrauf[2,*], Ishaq Isa[3], Badamasi Usman Zuntu[4], Maryam Abubakar Sharif[2]

1 Department of Information and Communication Technology, National Office for Technology Acquisition and Promotion, FCT Abuja, 900281, Nigeria; e-mail: chigbuuba@gmail.com (U. P. Chigbu).
2 Department of Computer Science, Federal Polytechnic Kaltungo, Kaltungo 770101, Gombe State, Nigeria; e-mail: abdulrashid.abdulrauf2022@nda.edu.ng (A. Abdulrauf), maryamsharif@fedpolyklt.edu.ng (M. A. Sharif).
3 Department of Computer Science, Federal University of Applied Sciences Kachia, Kachia 802101, Kaduna State, Nigeria; e-mail: ishaq.isa@fuask.edu.ng (I. Isa).
4 Department of Computer Science, Kaduna Polytechnic, Kaduna 800262, Kaduna State, Nigeria; e-mail: badamasi.usman.13806@kadunapolytechnic.edu.ng (B. U. Zuntu).
* Correspondence

The authors received no financial support for the research, authorship, and/or publication of this article.

Abstract: Software-Defined Networking (SDN) has become a key enabler of next-generation communication infrastructures because of its centralized control, programmability, and global network visibility. However, the centralized architecture also introduces significant security vulnerabilities, making SDN environments highly susceptible to attacks such as DoS, DDoS, probing, brute-force, and botnet activities. Although deep learning-based intrusion detection systems have achieved high detection accuracy, many existing approaches suffer from high computational complexity, long training time, and limited suitability for real-time deployment. This study addresses this gap by developing a lightweight stacking ensemble intrusion detection framework for SDN using the InSDN dataset. The proposed framework employs XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees as base learners, with Logistic Regression serving as the meta-learner. Experiments were conducted using 48-feature, 6-feature, and 4-feature configurations derived from previous feature-reduction studies. The results demonstrate consistently high detection performance, achieving accuracies above 99% across all feature subsets, with only marginal degradation under reduced feature dimensions. The framework showed excellent detection capability for major attack categories while maintaining reliable performance for most minority classes. These findings demonstrate that stacking ensemble learning is a practical and computationally efficient alternative to complex deep learning architectures for SDN intrusion detection, with strong potential for scalable and real-time cybersecurity deployment in modern network environments.

Keywords: DDOS; Detection; Ensemble; Intrusion; InSDN dataset; Stacking.

1. Introduction

Network security has become one of the most critical concerns in modern communication systems due to the rapid growth of interconnected networks and the increasing sophistication of cyber threats [1]. The expansion of IoT services, cloud computing, and large-scale digital infrastructures has significantly increased the attack surface of modern networks, making the protection of network resources and sensitive information a fundamental requirement for sustainable digital operations [1], [2]. In particular, IoT environments are characterized by heterogeneous devices, dynamic traffic patterns, and resource constraints, which make the deployment of conventional security mechanisms increasingly challenging [3], [4].

Software-Defined Networking (SDN) has emerged as a key technology for next-generation communication

infrastructures by decoupling the control plane from the data forwarding plane and enabling centralized network control, programmability, and global network visibility [5]–[7]. These capabilities provide substantial advantages for traffic engineering, automated policy enforcement, and simplified network management, and have also supported the development of scalable SDN-IoT environments [8]. However, the centralized SDN controller becomes a high-value target for cyberattacks, making SDN environments vulnerable to attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), probing, brute-force attacks, botnet activities, spoofing, and API exploitation [9], [10]. Such attacks threaten the confidentiality, integrity, and availability of network resources and can lead to large-scale service disruptions, particularly in SDN-IoT deployments [11].

Intrusion Detection Systems (IDS) are therefore essential for protecting SDN infrastructures by continuously monitoring network traffic and identifying malicious activities in real time [12], [13]. Traditional signature-based IDS are effective for detecting known attacks but struggle against zero-day and evolving threats, while anomaly-based IDS can detect previously unseen attacks but often suffer from high false alarm rates [14]. To overcome these limitations, recent studies have increasingly adopted machine learning and deep learning techniques for SDN intrusion detection [15]. In particular, Ataa *et al*. [10] demonstrated that CNN-LSTM and Transformer-based IDS models can achieve detection accuracies exceeding 99% on the InSDN dataset.

Despite their impressive detection performance, existing deep learning-based IDS frameworks are often associated with high computational complexity, long training times, substantial hardware requirements, and limited interpretability. These limitations make their deployment in real-time or resource-constrained SDN environments challenging. Although ensemble learning methods have shown strong performance in various cybersecurity applications, their potential as lightweight and scalable alternatives to complex deep learning architectures has not been sufficiently explored for SDN intrusion detection. Furthermore, the effectiveness of stacking ensemble learning under reduced feature configurations remains largely underinvestigated.

The effectiveness of IDS models also depends heavily on the quality and representativeness of the datasets used during training and evaluation [16], [17]. Among recent SDN intrusion detection datasets, the InSDN dataset has emerged as one of the most comprehensive and realistic benchmark datasets for SDN security research [18]. The dataset captures realistic SDN traffic behavior and includes multiple attack categories such as DoS, DDoS, Probe, Botnet, brute-force, and Web attacks. Ataa *et al*. [10] further optimized the dataset by reducing the original feature space to 48 features and subsequently investigating reduced 6-feature and 4-feature subsets while maintaining high intrusion detection accuracy.

To address the identified research gap, this study proposes a lightweight stacking ensemble intrusion detection framework for SDN using the InSDN dataset. The proposed framework combines XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees as base learners, with Logistic Regression serving as the meta-learner. The model is evaluated using 48-feature, 6-feature, and 4-feature configurations derived from the feature-reduction strategy reported by Ataa *et al*. [10], enabling an investigation of the robustness of the ensemble under varying feature dimensionalities.

The main contributions of this study are as follows:

- A lightweight stacking ensemble framework for SDN intrusion detection is developed using five tree-based ensemble learners and a Logistic Regression meta-learner.
- The framework is evaluated on the InSDN benchmark dataset using 48-feature, 6-feature, and 4-feature configurations to assess robustness under feature reduction.
- A comprehensive experimental analysis is conducted using accuracy, precision, recall, F1-score, confusion matrices, correlation analysis, and feature importance analysis.
- The proposed approach is compared with state-of-the-art deep learning models, demonstrating competitive detection performance with lower computational complexity.

The remainder of this paper is organized as follows. Section 2 reviews related work on SDN intrusion detection and ensemble learning approaches. Section 3 describes the dataset, preprocessing steps, feature analysis, and the proposed stacking ensemble framework. Section 4 presents the experimental results, feature contribution analysis, and discussion of the findings. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Review of Related Works

Software-Defined Networking has gained significant attention as a flexible and programmable networking paradigm capable of improving traffic management, scalability, and centralized network control. However, the separation of the control and data planes introduces serious security vulnerabilities, particularly because the SDN controller becomes a critical target for cyberattacks. Consequently, researchers have increasingly explored the use of machine learning and deep learning techniques to develop efficient IDS capable of detecting evolving attacks in SDN environments. Among the recent contributions, Ataa *et al*. [10] proposed and compared hybrid CNN-LSTM and Transformer-based intrusion detection models using the InSDN dataset. Their study demonstrated state-of-the-art

performance, where the Transformer model achieved 99.02% accuracy while the CNN-LSTM attained 99.01% under a 48-feature configuration. Further experiments involving feature reduction to 6 and 4 features, as well as attack-class merging strategies, improved performance further, with CNN-LSTM achieving 99.19% accuracy. The study highlighted the effectiveness of deep learning architectures and feature engineering in improving SDN intrusion detection performance.

Beyond deep learning approaches, several studies have emphasized the importance of feature selection and conventional machine learning techniques for enhancing IDS efficiency in SDN environments. Nadeem *et al*. [19] investigated multiple feature selection methods for DDoS attack detection in SDN and demonstrated that Recursive Feature Elimination combined with RF achieved an accuracy of 99.97%, confirming the importance of optimal feature selection for improving classifier performance. Similarly, Logeswari *et al*. [20] proposed a Hybrid Feature Selection-LightGBM framework that combined Correlation-based Feature Selection and RF-RFE to identify informative features for SDN intrusion detection. Their results showed that the LightGBM classifier significantly improved accuracy, precision, recall, and F1-score, demonstrating the efficiency of gradient-boosting approaches in SDN security. In another study, Mahmud *et al*. [21] evaluated several machine learning classifiers using the UNSW-NB15 dataset, including Random Forest, Decision Tree, AdaBoost, and Gradient Boosting. Among the evaluated models, Gradient Boosting achieved the best performance with 99.87% accuracy and 100% recall, highlighting the robustness of ensemble learning methods for intrusion detection in SDN environments.

Several researchers have also explored diverse deep learning architectures for improving SDN and SDN-IoT intrusion detection systems. Hadi and Mohammed [22] proposed a deep learning-based Network Intrusion Detection System integrating CNN, DNN, RNN, LSTM, and GRU models with feature selection techniques on the NSL-KDD dataset. Their approach reduced the original 41 features to 12 optimal features and achieved accuracies ranging from 97.78% to 98.63%, with CNN producing the best performance. Likewise, Maddu and Rao [23] developed a multi-stage IDS framework incorporating Deep Convolutional Generative Adversarial Networks, CenterNet-based feature extraction, and ResNet152V2 optimized with the Slime Mold Algorithm for attack classification on the InSDN and Edge-IIoT datasets. Their framework effectively detected and mitigated attacks such as DDoS, spoofing, and API exploitation. Also, Rohith *et al*. [24] introduced LightIDS-SDN, an explainable and federated IDS architecture that integrates Transformer layers, Capsule Networks, and BiLSTM units with quantum-inspired feature selection. Using the InSDN dataset, the proposed framework achieved 98.73% accuracy while also incorporating explainable AI techniques such as SHAP and Grad-CAM to improve transparency and interpretability.

The growing integration of SDN with the IoT has further accelerated the development of intelligent IDS frameworks capable of handling heterogeneous and resource-constrained environments. Chaganti *et al*. [25] proposed an LSTM-based IDS framework for SDN-IoT environments, achieving an accuracy of 97.1% for multiclass attack classification while also employing visualization methods to improve feature interpretability. Similarly, Elsayed *et al*. [26] introduced the Secured Automatic Two-level Intrusion Detection System (SATIDS), which utilized an improved LSTM architecture for hierarchical attack detection in SDN-IoT networks. Their model achieved 99.73% accuracy on the InSDN dataset and demonstrated strong scalability for smart-city applications. In another related work, Wani and Khaliq [27] developed an SDN-based IDS that leverages deep learning to secure IoT environments without imposing computational overhead on IoT devices. These studies collectively demonstrate the growing reliance on intelligent IDS solutions to secure SDN-IoT infrastructures against sophisticated cyber threats.

Apart from model development, several studies have focused on improving datasets and real-time IDS deployment in SDN environments. Yasarathna *et al*. [28] introduced ASEADOS-SDN-IoT, a publicly available SDN-IoT intrusion detection dataset containing 457,044 labeled flow instances with 83 statistical features collected from a hybrid ONOS-controlled OpenFlow testbed. Their dataset captured realistic SDN-IoT traffic and attack scenarios, thereby providing a reliable benchmark for IDS research. Similarly, Alzahrani and Alenazi [29] developed two SDN datasets using Mininet and the Ryu controller and evaluated multiple machine learning classifiers for real-time intrusion detection. Their findings showed that the Decision Tree classifier achieved outstanding throughput and F1-score performance using only three selected features, emphasizing the significance of feature reduction for real-time IDS deployment. Also, Rui *et al*. [30] proposed a hierarchical SDN-IoT routing and intrusion detection framework integrating ensemble learning with SDN-based subnet management, demonstrating improved attack detection and routing efficiency in IoT environments.

Recent review studies have further emphasized the growing importance of intelligent IDS techniques in SDN security. Kumar and Alqahtani [31] conducted a comprehensive review of ML and DL-based IDS frameworks for SDN, highlighting the strengths and limitations of existing approaches. Their study identified several persistent challenges, including scalability, real-time processing constraints, feature redundancy, dataset limitations, computational complexity, and the lack of interpretability in deep learning models. Although hybrid deep learning architectures have achieved impressive detection performance,

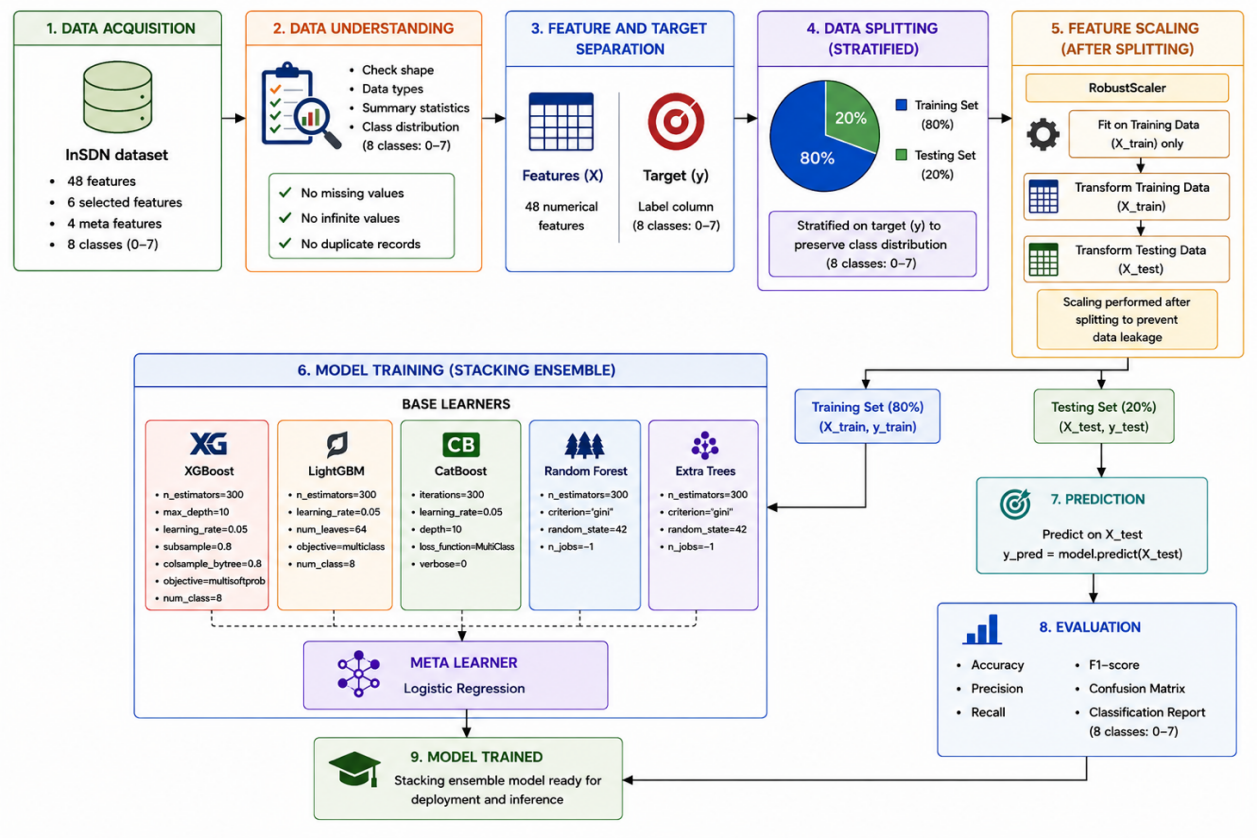

Figure 1. Proposed Process Framework.

Table 1. Classes distribution.

Label	Attack Category	Number of Samples
0	Normal	68,424
1	DoS	53,616
2	DDoS	121,942
3	Probe	98,129
4	Brute-force Attack	1,405
5	Exploitation (R2L)	17
6	Web Attack	192
7	Botnet Activity	164

many of these methods require high computational resources and complex training procedures. Consequently, recent studies increasingly recommend the exploration of ensemble learning approaches that combine multiple robust classifiers to improve generalization, reduce overfitting, and enhance computational efficiency.

Despite the limitations of the existing studies, and the high accuracy achieved by Ataa *et al*. [10], the study primarily focused on deep learning architectures and did not explore advanced ensemble learning approaches that could achieve comparable performance with lower computational overhead. Although effective feature reduction was demonstrated on the InSDN dataset, a research gap still exists in developing an efficient ensemble stacking-based IDS using reduced feature subsets and robust classifiers such as XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees to enhance scalability, efficiency, and generalization in SDN intrusion detection.

3. Method and Materials

This section presents the overview of the dataset, data preprocessing, and model development. Figure 1 gives a clear diagrammatic description of the proposed model, and the steps involved are discussed in subsequent sections.

3.1. Dataset

This study employed the InSDN dataset, originally introduced and further refined in the work of Ataa *et al*. [10] due to its relevance and suitability for Software-Defined Networking (SDN) intrusion detection research. The InSDN dataset represents realistic SDN network traffic and provides detailed packet-level information that captures diverse network activities, including normal communications, flow behaviors, and multiple cyberattack scenarios. The dataset effectively reflects the complexity of SDN infrastructures, thereby making it a reliable benchmark for evaluating intrusion detection models.

The dataset contains both benign and malicious traffic records with a total of 343,889 instances. Initially, each traffic record consisted of 77 raw features after removing socket-related information such as IP addresses, flow identifiers, and similar attributes. Subsequent studies further optimized the dataset by reducing the feature dimensionality to improve computational efficiency and reduce redundancy. Building upon the feature engineering process proposed by Ataa *et al*. [10], this study employed the reduced feature subsets consisting of 48, 6, and 4 features

Table 2. Feature Sets Used in the Study.

48-Feature Model	6-Feature Model	4-Feature Model
Protocol	Flow Duration	Flow Duration
Flow Duration	Flow Byts/s	Flow Byts/s
Tot Fwd Pkts	Fwd Header Len	Bwd Header Len
Tot Bwd Pkts	Bwd Header Len	Pkt Len Std
TotLen Fwd Pkts	Pkt Len Std	
TotLen Bwd Pkts	Pkt Size Avg	
Fwd Pkt Len Max		
Fwd Pkt Len Min		
Fwd Pkt Len Mean		
Fwd Pkt Len Std		
Bwd Pkt Len Max		
Bwd Pkt Len Min		
Bwd Pkt Len Mean		
Bwd Pkt Len Std		
Flow Byts/s		
Flow Pkts/s		
Flow IAT Mean		
Flow IAT Std		
Flow IAT Max		
Flow IAT Min		
Fwd IAT Tot		
Fwd IAT Mean		
Fwd IAT Std		
Fwd IAT Max		
Fwd IAT Min		
Bwd IAT Tot		
Bwd IAT Mean		
Bwd IAT Std		
Bwd IAT Max		
Bwd IAT Min		
Fwd Header Len		
Bwd Header Len		
Fwd Pkts/s		
Bwd Pkts/s		
Pkt Len Min		
Pkt Len Max		
Pkt Len Mean		
Pkt Len Std		
Pkt Len Var		
Pkt Size Avg		
Active Mean		
Active Std		
Active Max		
Active Min		
Idle Mean		
Idle Std		
Idle Max		
Idle Min		

in order to evaluate the effectiveness of the proposed ensemble stacking intrusion detection framework under varying feature dimensionalities.

3.1.1. Dataset Classes

The dataset contains the following eight classes and distribution are shown in Table 1. To improve the clarity and reproducibility of the study, Table 2 summarizes the complete set of 48 traffic-flow features used in the full model and explicitly identifies the features selected for the reduced 6-feature and 4-feature configurations adopted from the feature-reduction strategy of Ataa *et al*. [10].

The 6-feature configuration consisted of Flow Duration, Flow Byts/s, Fwd Header Len, Bwd Header Len, Pkt Len Std, and Pkt Size Avg, while the 4-feature configuration retained Flow Duration, Flow Byts/s, Bwd Header Len, and Pkt Len Std. These reduced feature sets were adopted from the feature-reduction strategy reported by Ataa *et al*. [10] and were used to evaluate the robustness of the proposed stacking ensemble under progressively lower-dimensional feature spaces.

3.2. Data Understanding and Exploration

Data understanding and exploratory analysis were conducted prior to model development in order to examine the structural characteristics, dimensionality, and statistical properties of the dataset. This phase was necessary to identify potential inconsistencies, redundant records, missing values, and abnormal feature distributions that could negatively affect model performance. The exploratory analysis involved examining the dataset dimensions, feature distributions, statistical summaries, and class imbalance characteristics.

The exploratory analysis revealed that all selected features in the dataset were numerical traffic-flow attributes, which simplified the preprocessing stage because categorical encoding was not required. The dataset was also examined for duplicate records, missing values, and infinite values that could negatively affect model performance. The inspection confirmed that the original InSDN dataset contained no duplicate records, no missing values, and no infinite values. Understanding these characteristics was essential for designing an effective preprocessing pipeline and ensuring stable model training.

3.2.1. Correlation Analysis of Features

To investigate the relationships among the input features, a Pearson correlation analysis was conducted on the 48 numerical features of the InSDN dataset. The resulting correlation matrix was visualized using a correlation heatmap, as shown in Figure 2. The analysis provides insight into the degree of linear association between pairs of features and helps identify redundant or highly correlated variables that may influence model learning.

The heatmap revealed that most feature pairs exhibited low to moderate correlation, indicating that the dataset contains diverse and complementary network traffic characteristics. However, a few feature groups showed strong positive correlations, particularly among flow-based traffic statistics such as packet rate, byte rate, and packet length features. This behavior is expected because these attributes are derived from related network flow

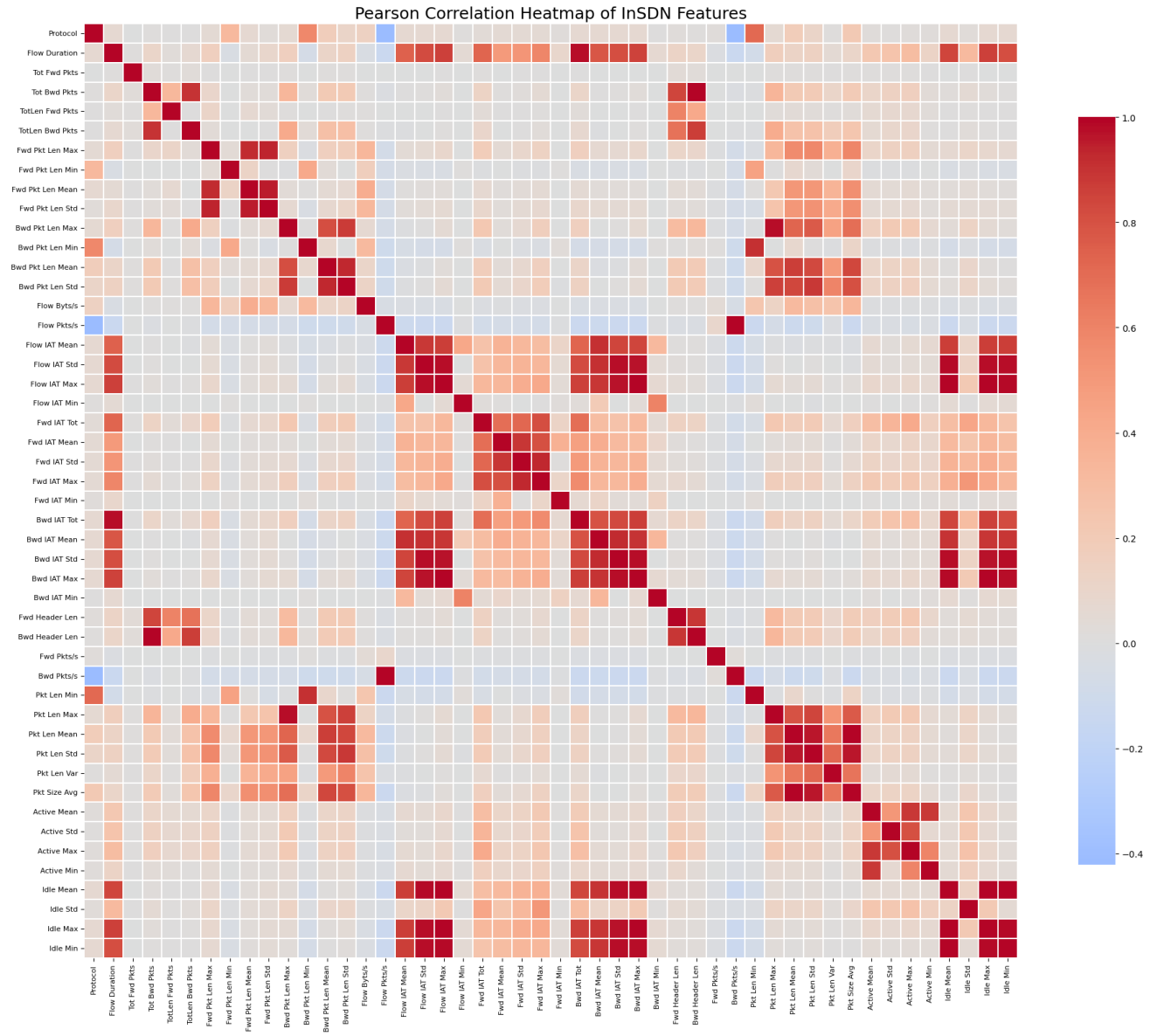

Figure 2. Pearson Correlation Heatmap of the Selected InSDN Features.

measurements and therefore capture similar traffic properties.

Despite the presence of some highly correlated features, all 48 features were retained in this study because tree-based ensemble models such as XGBoost, LightGBM, Random Forest, and Extra Trees are relatively robust to multicollinearity. The correlation analysis nevertheless enhances the interpretability of the dataset and provides additional evidence that the selected feature set captures multiple dimensions of network traffic behavior relevant for attack classification.

3.3. Data Pre-Processing

Data preprocessing was conducted to ensure the consistency and suitability of the dataset for model training. The exploratory analysis confirmed that the original InSDN dataset contained no duplicate records, no missing values, and no infinite values. Consequently, no data-cleaning operations, such as duplicate removal, missing-value imputation, or infinite-value handling, were required. The preprocessing stage therefore focused on separating the dataset into independent variables and target labels. The independent variables consisted of the selected traffic-flow features used for intrusion detection, while the target labels represented the eight attack categories contained in the dataset. The resulting feature matrix comprised numerical traffic-flow attributes describing packet behavior, flow statistics, timing characteristics, and protocol interactions associated with both benign and malicious network traffic.

3.4. Feature Scaling

Feature scaling was performed to normalize the numerical ranges of the traffic-flow features and improve the performance of the machine learning algorithms used in this study. Since network traffic features often exhibit varying scales and contain abnormal spikes caused by malicious traffic behavior, scaling was necessary to ensure that no individual feature disproportionately influenced the learning process.

Figure 3. Class distribution of the training and testing datasets after the 80:20 stratified split.

Table 3. Class Distribution After the 80:20 Stratified Split.

Class	Training set (80%)	Testing set (20%)
Normal	54,739	13,685
DoS	42,893	10,723
DDoS	97,553	24,389
Probe	78,503	19,626
Brute-force Attack	1,124	281
Exploitation (R2L)	14	3
Web Attack	154	38
Botnet Activity	131	33
Total	275,111	68,778

This study adopted the RobustScaler normalization technique because of its ability to effectively handle outliers commonly present in network traffic datasets. Unlike conventional normalization techniques such as StandardScaler, RobustScaler utilizes the median and interquartile range, making it less sensitive to extreme values and abnormal traffic fluctuations. The scaling process transformed all feature values into comparable numerical ranges while preserving important statistical properties relevant to intrusion detection.

3.5. Dataset Splitting

After preprocessing, the dataset was divided into training and testing subsets using 80:20 stratified split. Feature scaling was then performed by fitting the RobustScaler on the training data only, and the learned transformation was subsequently applied to the test data. This procedure was adopted to prevent data leakage and ensure an unbiased evaluation of the proposed model.

Stratified sampling was employed during the splitting process to preserve the original class distribution across both the training and testing datasets. This approach was particularly important because the dataset contained highly imbalanced attack classes. Maintaining consistent class proportions across both subsets ensured that minority attack categories were adequately represented during training and evaluation, thereby improving the reliability of the experimental results.

The class distributions in the training and testing subsets remained proportional to the original dataset distribution (see Figure 3 and Table 3), confirming that the stratified sampling strategy successfully preserved the relative frequency of each attack category. This ensured that both subsets contained representative samples of majority and minority classes, thereby reducing sampling bias and improving the reliability of the experimental evaluation.

3.6. Proposed Stacking Ensemble Framework

This study proposes a stacking ensemble-based intrusion detection framework for multi-class attack classification in Software-Defined Networking environments. Stacking ensemble learning is an advanced machine learning technique that combines the predictive strengths of multiple base learners through a secondary learning model known as a meta learner. The objective of stacking is to improve predictive accuracy, robustness, and generalization by leveraging the complementary capabilities of diverse classifiers.

The proposed framework integrates five powerful tree-based ensemble learning algorithms as base learners, including XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees classifiers. These algorithms were selected because of their proven effectiveness in handling high-dimensional tabular datasets, nonlinear feature interactions, imbalanced classes, and complex decision boundaries commonly observed in cybersecurity applications. Table 4 present the summary of the configuration used.

XGBoost was selected for its strong regularization capability and efficient gradient boosting mechanism, which enables accurate modeling of complex traffic behaviors. LightGBM was incorporated because of its fast-training speed and efficient leaf-wise tree growth strategy, making it suitable for large-scale intrusion detection tasks. CatBoost was included due to its ability to reduce prediction

Table 4. Proposed Model Configuration.

Model Component	Algorithm	Configuration/Parameters
Base Learner 1	XGBoost Classifier (XGBoost)	Number of estimators = 300, maximum tree depth = 10, learning rate = 0.05, subsampling enabled, column sampling enabled, multi-class objective function
Base Learner 2	Light Gradient Boosting Machine (LightGBM)	Number of estimators = 300, learning rate = 0.05, leaf-wise tree growth strategy with 64 leaves
Base Learner 3	CatBoost Classifier	Number of iterations = 300, learning rate = 0.05, tree depth = 10
Base Learner 4	Random Forest Classifier	Number of decision trees = 300, random feature selection, bagging-based ensemble learning
Base Learner 5	Extra Trees Classifier	Number of decision trees = 300, randomized split selection strategy
Meta Learner	Logistic Regression	Maximum iteration = 1000
Ensemble Framework	Stacking Ensemble Learning	Five-fold cross-validation, parallel model training enabled

bias and improve classification performance through ordered boosting mechanisms. Random Forest was utilized because of its robustness against overfitting and strong generalization capability through bagging and feature randomness. Extra Trees was incorporated to introduce additional randomness during tree construction, thereby improving model diversity and reducing variance.

The outputs generated by the base learners were subsequently combined using a Logistic Regression meta learner. The meta learner was responsible for learning the optimal combination of predictions generated by the base models in order to produce the final classification output. Logistic Regression was selected as the meta learner because of its simplicity, computational efficiency, and ability to effectively combine probabilistic outputs from multiple classifiers.

The stacking ensemble architecture was implemented using a five-fold cross-validation strategy during training. Cross-validation enabled the meta learner to learn from out-of-fold predictions generated by the base learners, thereby reducing overfitting and improving generalization capability. The proposed ensemble framework therefore combines the strengths of multiple robust classifiers to achieve improved attack detection performance, reduced false-positive rates, and enhanced classification stability in SDN intrusion detection tasks.

3.7. Model Evaluation

Model performance was evaluated using standard classification metrics, including Accuracy, Precision, Recall, and F1-score. These metrics were selected because they provide complementary information about the classification capability of the model, particularly under imbalanced class distributions where accuracy alone may be insufficient.

For a confusion matrix consisting of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), the evaluation metrics are defined as follows:

1. **Accuracy** measures the proportion of correctly classified samples among all evaluated samples and is defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

2. **Precision** measures the proportion of predicted positive samples that are actually positive and is expressed as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. **Recall**, also known as sensitivity or true positive rate, measures the proportion of actual positive samples that are correctly identified and is expressed as:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4. The **F1-score** is the harmonic mean of Precision and Recall and provides a balanced measure of classification performance and is defined as:

$$F1 - Score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Particular emphasis was placed on Recall and F1-score, as these metrics are especially important in intrusion detection scenarios where minimizing false negatives is critical for ensuring that malicious activities are not overlooked.

4. Results and Discussion

This section presents the experimental results obtained from the proposed stacking ensemble model using the InSDN dataset under different feature configurations, including 48 features, 6 features, and 4 features. The performance of the proposed model is comprehensively evaluated and discussed using standard classification metrics, including accuracy, precision, recall, and F1-score.

Table 5. Proposed Model Performance.

Num of Features	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
48 features	99.1916	99.1823	99.1916	99.1847
6 features	99.1567	99.1482	99.1567	99.1491
4 features	99.0389	99.0240	99.0389	99.0167

Table 6. Classification Report for 48-Feature Model.

Class	Precision	Recall	F1-Score	Support
Normal	0.98	0.98	0.98	13685
DDoS	1.00	1.00	1.00	24389
DoS	1.00	1.00	1.00	10723
Brute-force Attack	0.90	0.78	0.84	281
Web Attack	0.69	0.63	0.66	38
Probe	0.99	0.99	0.99	19626
Botnet Activity	0.91	0.97	0.94	33
Exploitation (R2L)	0.00	0.00	0.00	3
Accuracy			0.99	68778
Macro Avg	0.81	0.79	0.80	68778
Weighted Avg	0.99	0.99	0.99	68778

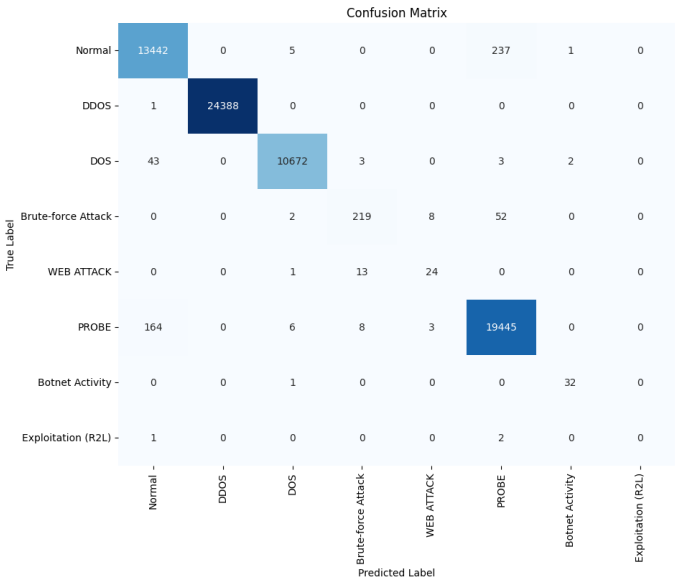

Figure 4. Confusion Matrix of the Proposed Model Using 48 Features.

4.1. Performance Evaluation and Confusion Matrix analysis Across Feature Subsets

The experimental results obtained from the proposed stacking ensemble intrusion detection framework demonstrate outstanding classification performance across all evaluated feature configurations of the InSDN dataset. The performance of the proposed is presented in Table 5.

Using the 48-feature subset, the proposed model achieved the highest overall performance with an accuracy of 99.1916%, precision of 99.1823%, recall of 99.1916%, and F1-score of 99.1847%. These results indicate that the ensemble framework effectively captured the complex traffic-flow characteristics associated with both benign and malicious network activities within the SDN environment. The high precision value demonstrates the model's ability to minimize false-positive predictions, while the high recall value indicates effective detection of malicious traffic with very few false negatives. Furthermore, the balanced F1-score confirms the consistency and reliability of the proposed framework in handling multi-class intrusion detection tasks.

The confusion matrix presented in Figure 4 and the classification report in Table 6 further validate the strong classification capability of the proposed model using the 48-feature subset. The results show that the majority of traffic and attack classes were correctly classified with minimal misclassification errors across the evaluated categories. The Normal, DoS, DDoS, and Probe classes achieved exceptionally high detection performance, indicating that the proposed model effectively learned the distinguishing characteristics of the major traffic patterns and attack behaviors. The model demonstrated particularly strong discrimination capability for the dominant attack categories, while the minority classes exhibited comparatively lower performance due to the limited number of available samples and the severe class imbalance. Brute-force and Web Attack categories experienced some degree of misclassification, whereas Botnet Activity was detected with high reliability. In contrast, the Exploitation (R2L) class was not correctly identified by the model, which can be attributed to the extremely small number of samples available for that category in the test set.

Using the reduced 6-feature subset, the proposed model maintained remarkably high performance, achieving an accuracy of 99.1567%, precision of 99.1482%, recall of 99.1567%, and F1-score of 99.1491%. Although a slight reduction in performance was observed compared to the 48-feature configuration, the difference was marginal, indicating that the selected six features retained sufficient discriminative information for accurate attack classifica-

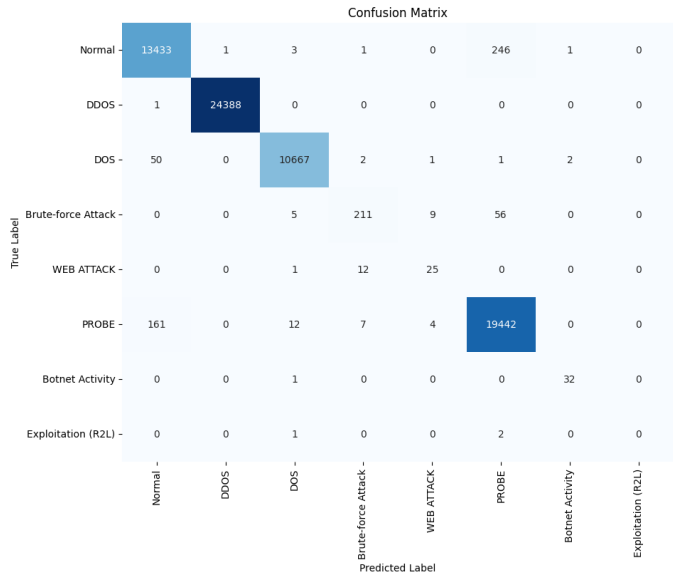

Figure 5. Confusion Matrix of the Proposed Model Using 6 Features.

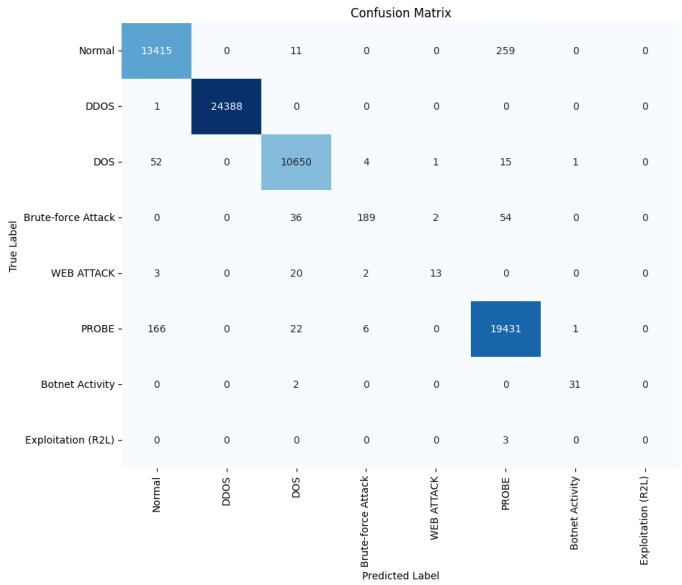

Figure 6. Confusion Matrix of the Proposed Model Using 4 Features.

Table 7. Classification Report for 6-Feature Model.

Class	Precision	Recall	F1-Score	Support
Normal	0.98	0.98	0.98	13685
DDOS	1.00	1.00	1.00	24389
DOS	1.00	0.99	1.00	10723
Brute-force Attack	0.91	0.75	0.82	281
WEB ATTACK	0.64	0.66	0.65	38
PROBE	0.98	0.99	0.99	19626
Botnet Activity	0.91	0.97	0.94	33
Exploitation (R2L)	0.00	0.00	0.00	3
Accuracy			0.99	68778
Macro Avg	0.80	0.79	0.80	68778
Weighted Avg	0.99	0.99	0.99	68778

Table 8. Classification Report for 4-Feature Model.

Class	Precision	Recall	F1-Score	Support
Normal	0.98	0.98	0.98	13685
DDOS	1.00	1.00	1.00	24389
DOS	0.99	0.99	0.99	10723
Brute-force Attack	0.94	0.67	0.78	281
WEB ATTACK	0.81	0.34	0.48	38
PROBE	0.98	0.99	0.99	19626
Botnet Activity	0.94	0.94	0.94	33
Exploitation (R2L)	0.00	0.00	0.00	3
Accuracy			0.99	68778
Macro Avg	0.83	0.74	0.77	68778
Weighted Avg	0.99	0.99	0.99	68778

tion. This result demonstrates the effectiveness of the feature reduction process in decreasing dimensionality and computational complexity without significantly affecting model performance.

As illustrated in Figure 5 and summarized in Table 7, the 6-feature model shows classification behavior similar to that achieved with the 48-feature configuration. Most traffic samples were correctly classified, with only minor increases in misclassification rates across a few classes. The DoS, DDoS, and Probe attack classes continued to achieve near-perfect classification performance, while slight confusion persisted between WEB ATTACK and Brute-force categories. Importantly, the model maintained strong detection capability for minority attack categories despite severe class imbalance.

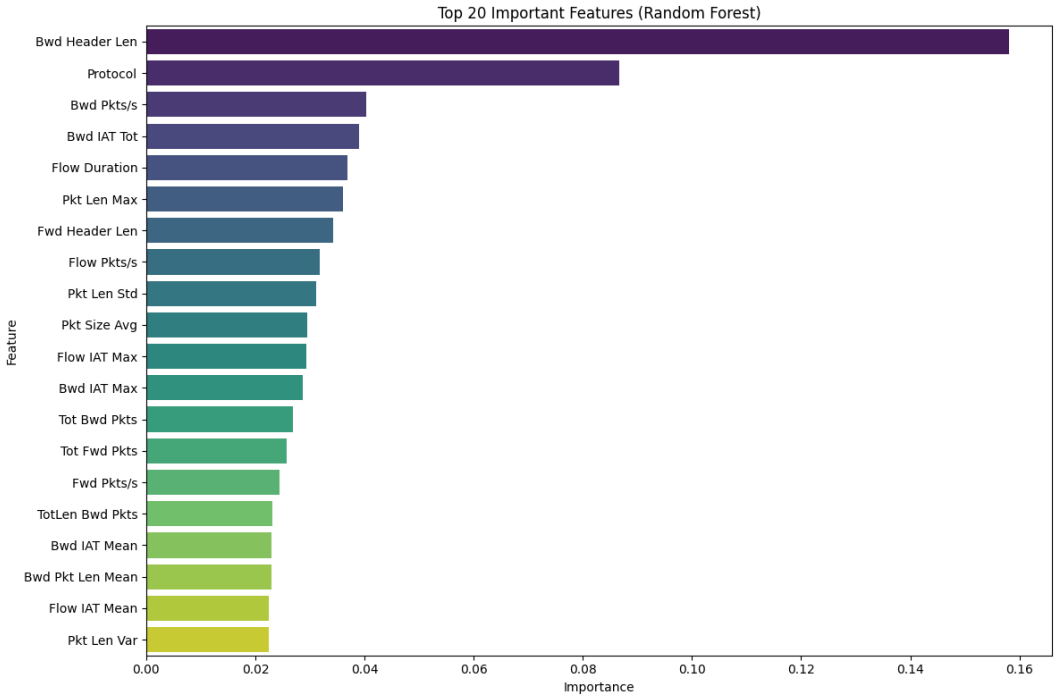

Figure 7. Top 20 Feature Importance of the Proposed Model Using 48 Features.

Similarly, the 4-feature subset also produced highly competitive results, recording an accuracy of 99.0389%, precision of 99.0240%, recall of 99.0389%, and F1-score of 99.0167%. Although performance slightly decreased compared to the other feature subsets, the results remain exceptionally strong given the drastic reduction in feature dimensionality.

The confusion matrix shown in Figure 6 and the classification report presented in Table 8 reveal that the proposed model maintained effective classification performance even with only four selected features. While most Normal, DoS, and DDoS traffic instances were correctly identified, a higher degree of misclassification was observed in WEB ATTACK and Brute-force categories. Despite this, the model still demonstrated strong generalization capability.

4.2. Analysis of Feature Contribution

To identify the most influential traffic-flow attributes contributing to intrusion detection performance, a feature importance analysis was conducted using a Random Forest model trained on the 48-feature InSDN configuration. The importance scores were obtained from the feature importances attribute of the trained Random Forest model, which computes impurity-based feature importance based on the mean decrease in impurity across the ensemble of decision trees. Figure 7 presents the top 20 ranked features according to these importance scores. As shown in Figure 7, the Random Forest-based analysis indicates that Bwd Header Length is the most dominant feature with an importance score of 0.158052, significantly outperforming all other attributes. This suggests that backward traffic header characteristics play a critical role in distinguishing between normal and malicious network behaviors in Software-Defined Networking (SDN) environments. Similarly, the Protocol type feature (0.086718) is identified as the second most important variable, indicating that variations in communication protocols strongly influence attack detection capability.

Other highly ranked features such as Backward Packets per Second, Backward Inter-Arrival Time Total, Flow Duration, and Maximum Packet Length also contribute substantially to the model's decision-making process. These features capture essential traffic-flow dynamics such as timing behavior, packet distribution, and flow intensity, which are commonly exploited by different types of network attacks. Furthermore, features related to packet length statistics (mean, variance, and standard deviation) and flow inter-arrival times demonstrate moderate importance, highlighting the relevance of both temporal and statistical traffic characteristics in intrusion detection. It should be noted that the reported importance values reflect the behavior of the Random Forest model and are used primarily to provide interpretability regarding influential traffic attributes. They do not represent a unified feature-importance measure for the entire stacking ensemble framework. The feature importance analysis was conducted using the 48-feature configuration, which represents the full feature set employed in this study. Although feature importance scores were also generated for the reduced 6-feature and 4-feature subsets during experimentation, they are not reported here because these subsets contain only a small number of preselected features, and the resulting rankings provide limited additional interpretive value.

4.3. Discussion of the findings

The experimental results obtained from the proposed stacking ensemble intrusion detection framework demonstrate consistently high and stable performance across all evaluated feature configurations of the InSDN dataset. The findings indicate that the proposed model is highly effective in capturing both global and local traffic-flow patterns associated with normal and malicious network behaviors within Software-Defined Networking environments. Across all experimental settings, the model-maintained accuracy values above 99%, even under significant feature reduction from 48 to 4 features, demonstrating strong robustness, generalization capability, and resilience to dimensionality reduction.

The results further show that reducing the feature space had only a marginal impact on performance. While the 48-feature configuration achieved the highest overall results, the 6-feature and 4-feature configurations still preserved strong predictive capability with only slight performance degradation. This indicates that the most informative traffic characteristics are concentrated within a relatively small subset of features, and that redundant or less informative attributes can be removed without significantly affecting classification quality. This is particularly important for real-time SDN intrusion detection systems, where computational efficiency and low latency are critical requirements.

From a class-wise perspective, the proposed model demonstrates excellent discrimination capability for the dominant DoS and DDoS attack classes, which achieved exceptionally high detection performance across all evaluated feature subsets. The Normal traffic class was also identified with high reliability, indicating that the model effectively learned the distinguishing characteristics of legitimate network behavior. In contrast, the Probe class achieved comparatively lower detection performance, suggesting that Probe traffic remains more challenging to distinguish from other attack categories under the current feature configuration. The minority classes exhibited varying levels of performance due to the severe class imbalance and the limited number of available samples. Brute-force and Web Attack categories experienced some degree of misclassification, whereas Botnet Activity was detected with high reliability. The Exploitation (R2L) class was not correctly identified by the model, which can be attributed to the extremely small number of samples available for that category in the test set. Despite these limitations, the overall classification performance remained exceptionally strong, highlighting the robustness and generalization capability of the proposed stacking ensemble framework under highly imbalanced SDN intrusion detection conditions.

It is important to note that no explicit class imbalance handling technique, such as oversampling, undersampling, class weighting, or cost-sensitive learning, was employed in this study. This decision was made to preserve the original class distribution of the InSDN dataset and to enable a fair comparison with the baseline deep learning models reported by Ataa *et al*. [10]. However, the severe imbalance of the dataset, particularly the extremely small number of Exploitation (R2L) samples and the limited Web Attack and Brute-force samples, contributed to the comparatively lower detection performance observed for these minority classes. The results suggest that incorporating dedicated imbalance-handling strategies could further improve minority-class detection performance and represents an important direction for future research.

The feature importance analysis further provides insight into the decision-making behavior of the proposed model. Traffic-flow characteristics related to backward communication patterns, protocol behavior, packet timing, and flow duration were identified as the most influential factors in intrusion detection. In particular, backward header-related attributes and protocol information play a dominant role in distinguishing malicious traffic from legitimate flows. This confirms that SDN attack behaviors are strongly reflected in asymmetric traffic patterns and temporal flow irregularities, which are effectively captured by tree-based ensemble learning models.

As summarized in Table 9, When compared with state-of-the-art deep learning-based approaches, particularly the work of Ataa *et al*. [10] the proposed stacking ensemble framework demonstrates competitive and, in some cases, improved performance while offering significantly reduced computational complexity. Unlike deep learning architectures such as CNN-LSTM and Transformer models that require high computational resources and longer training times, the proposed ensemble approach achieves comparable or superior results using efficient tree-based models combined through stacking.

The comparative results clearly demonstrate that the proposed stacking ensemble model consistently outperforms the best-performing deep learning models reported by Ataa *et al*. [10] across all feature configurations. Notably, the performance gap becomes more pronounced as the feature dimensionality decreases, indicating that the proposed model is more resilient to feature reduction and better suited for lightweight intrusion detection scenarios. This makes it particularly appropriate for real-world SDN deployments where computational efficiency, scalability, and real-time response are critical constraints.

5. Limitations

Although the proposed stacking ensemble framework achieved high overall intrusion detection performance, several limitations should be acknowledged. First, the InSDN dataset exhibits severe class imbalance, particularly for the Exploitation (R2L) category, which contains only a very small number of samples. This imbalance negatively affected the detection performance of extremely

Table 9. Comparative Performance Analysis with State-of-the-Art.

Number of Features	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
48 Features	CNN-LSTM [10]	99.01	99.00	99.02	99.02
48 Features	Transformer [10]	99.02	99.03	99.05	99.05
48 Features	Proposed Stacking Ensemble	99.19	99.18	99.19	99.18
6 Features	CNN-LSTM [10]	98.80	98.80	98.90	98.90
6 Features	Transformer [10]	98.90	98.85	98.85	98.91
6 Features	Proposed Stacking Ensemble	99.16	99.15	99.16	99.15
4 Features	CNN-LSTM [10]	98.20	98.20	98.20	98.20
4 Features	Transformer [10]	98.50	98.47	98.52	98.49
4 Features	Proposed Stacking Ensemble	99.04	99.02	99.04	99.01

rare attack classes. Second, the study did not incorporate dedicated class imbalance mitigation techniques such as oversampling, undersampling, class-weighted learning, or cost-sensitive ensemble methods. Consequently, the reported performance reflects the behavior of the model under the original dataset distribution. Third, the experiments were conducted on a single benchmark dataset, and the generalization capability of the proposed framework across other SDN or SDN-IoT datasets was not evaluated. Finally, although the proposed ensemble demonstrated competitive performance compared with existing deep learning approaches, the study did not include repeated experimental runs or statistical significance testing to determine whether the observed performance differences are statistically significant.

6. Future Work

Future research will focus on improving the detection performance of minority attack categories by incorporating advanced class imbalance handling techniques such as SMOTE, adaptive synthetic sampling, class-weighted learning, and cost-sensitive ensemble strategies. Additional studies will evaluate the proposed framework on multiple benchmark datasets and real-world SDN traffic traces in order to assess its generalization capability and robustness under diverse network conditions. Another important direction is the development of real-time deployment and online learning mechanisms that can adapt to evolving attack patterns in dynamic SDN environments. Also, future work will investigate the integration of explainable artificial intelligence techniques, including SHAP and LIME, to provide more transparent explanations of ensemble predictions and improve the interpretability of intrusion detection decisions.

Finally, repeated experimental runs and statistical significance analysis will be conducted to provide stronger evidence regarding the comparative performance of the proposed framework against existing state-of-the-art approaches.

7. Conclusion

This study presented a stacking ensemble-based intrusion detection framework for Software-Defined Networking environments using the InSDN dataset under multiple feature configurations (48, 6, and 4 features). The proposed model integrated XGBoost, LightGBM, CatBoost, Random Forest, and Extra Trees as base learners, with Logistic Regression as the meta-learner, to enhance classification accuracy, robustness, and generalization while maintaining computational efficiency. Experimental results demonstrated consistently high performance across all feature subsets, achieving accuracies above 99% even under significant feature reduction, thereby confirming the effectiveness of the selected informative features and the resilience of the ensemble approach. The model exhibited excellent detection capability for the dominant DoS and DDoS attack categories and maintained reliable performance for Normal traffic and several minority attack classes, although Probe traffic proved comparatively more challenging to distinguish under the current feature configuration. The results also highlighted the impact of severe class imbalance on extremely rare attack categories, particularly Exploitation (R2L), which contained only a few samples in the test set.

Comparative analysis with state-of-the-art deep learning models, including CNN-LSTM and Transformer architectures **reported by** Ataa *et al*. [10], revealed that the proposed stacking ensemble achieved competitive detection performance and, in some feature configurations, slightly higher performance, while offering lower computational complexity and improved suitability for real-time deployment scenarios. A limitation of the current study is that no dedicated class imbalance mitigation strategy was incorporated, which affected the detection performance of extremely rare attack categories, particularly Exploitation (R2L). Future work will investigate the integration of oversampling, class-weighted learning, and cost-sensitive ensemble techniques to improve minority-class detection performance. These findings demonstrate that ensemble learning is a practical and computationally efficient alternative to hybrid deep learning approaches for SDN intrusion detection, although additional repeated experiments and statistical significance analysis would be valuable for confirming the consistency of the observed performance differences.

8. Declarations

8.1. Author Contributions

Ubakaghinwa Paul Chigbu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft; **Abdulrashid Abdulrauf:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision, Project administration, Funding acquisition; **Ishaq Isa:** Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft; **Badamasi Usman Zuntu:** Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft; **Maryam Abubakar Sharif:** Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft.

8.2. Institutional Review Board Statement

Not applicable.

8.3. Informed Consent Statement

Not applicable.

8.4. Data Availability Statement

The datasets that support the findings of this research are available in "InSDN dataset files" at https://drive.google.com/drive/folders/16bRX1uo6zyKlk-MgKqZDyc4DeYuBzfOxx?usp=sharing, reference number [10].

8.5. Acknowledgment

Not applicable.

8.6. Conflicts of Interest

The authors declare no conflicts of interest.

9. References

[1] S. Pandey, M. Chaudhary, and Z. Tóth, "An investigation on real-time insights: enhancing process control with IoT-enabled sensor networks," *Discover Internet of Things*, vol. 5, no. 1, p. 29, 2025. https://doi.org/10.1007/s43926-025-00124-6.

[2] A. Salam, "Internet of things for sustainability: perspectives in privacy, cybersecurity, and future trends," in *Internet of Things for Sustainable Community Development: Wireless Communications, Sensing, and Systems*, Cham: Springer International Publishing, 2019, pp. 299–326. https://doi.org/10.1007/978-3-030-35291-2_10.

[3] T. Magara and Y. Zhou, "Internet of things (IoT) of smart homes: privacy and security," *Journal of Electrical and Computer Engineering*, vol. 2024, no. 1, p. 7716956, 2024. https://doi.org/10.1155/2024/7716956.

[4] J. M. Kizza, "Internet of things (IoT): growth, challenges, and security," in *Guide to Computer Network Security*, Cham: Springer International Publishing, 2024, pp. 557–573. https://doi.org/10.1007/978-3-031-47549-8_25.

[5] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014. https://doi.org/10.1109/JPROC.2014.2371999.

[6] S. Singh and R. K. Jha, "A survey on software defined networking: architecture for next generation network," *J. Netw. Syst. Manage.*, vol. 25, no. 2, pp. 321–374, 2017. https://doi.org/10.1007/s10922-016-9393-9.

[7] N. Anand *et al.*, "Securing Software Defined Networks: A Comprehensive Analysis of approaches, applications, and Future Strategies against DoS Attacks," *IEEE Access*, vol. 13, pp. 64473 – 64515, 2024. https://doi.org/10.1109/ACCESS.2024.3520478.

[8] N. Indrason and G. Saha, "Exploring Blockchain-driven security in SDN-based IoT networks," *Journal of Network and Computer Applications*, vol. 224, p. 103838, 2024. https://doi.org/10.1016/j.jnca.2024.103838.

[9] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *2015 International Conference on Computing, Networking and Communications (ICNC)*, 2015, pp. 77–81. https://doi.org/10.1109/ICCNC.2015.7069319.

[10] M. S. Ataa, E. E. Sanad, and R. A. El-khoribi, "Intrusion detection in software defined network using deep learning approaches," *Sci. Rep.*, vol. 14, p. 29159, 2024. https://doi.org/10.1038/s41598-024-79001-1.

[11] S. Layeghy, M. Gallagher, and M. Portmann, "Benchmarking the benchmark—Comparing synthetic and real-world Network IDS datasets," *Journal of Information Security and Applications*, vol. 80, p. 103689, 2024. https://doi.org/10.1016/j.jisa.2023.103689.

[12] C. Singh and A. K. Jain, "A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 8, p. 100543, 2024. https://doi.org/10.1016/j.prime.2024.100543.

[13] A. Abderrahmane, H. Drid, and A. Behaz, "A survey of controller placement problem in SDN-IoT network," *International Journal of Networked and Distributed Computing*, vol. 12, no. 2, pp. 170–184, 2024. https://doi.org/10.1007/s44227-024-00035-y.

[14] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 258–263. https://doi.org/10.1109/WINCOM.2016.7777224.

[15] Q. Yan, W. Huang, X. Luo, Q. Gong, and F. R. Yu, "A multi-level DDoS mitigation framework for the industrial internet of things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 30–36, 2018. https://doi.org/10.1109/MCOM.2018.1700621.

[16] H. Dhirar and A. Hamad, "Comparative evaluation of a novel IDS dataset for SDN-IoT using deep learning models against InSDN, BoT-IoT, and ToN-IoT," *Measurement: Digitalization*, p. 100015, 2025. https://doi.org/10.1016/j.meadig.2025.100015.

[17] Sarika and R. Dass, "Design of a Novel Network Intrusion Detection Technique for SDN-based IoT Network Using Machine Learning," *Optoelectronics, Instrumentation and Data Processing*, vol. 61, no. 3, pp. 396–407, 2025. https://doi.org/10.3103/S8756699025700451.

[18] M. S. Elsayed, N. A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, pp. 165263–165284, 2020. https://doi.org/10.1109/ACCESS.2020.3022633.

[19] 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. http://dx.doi.org/10.32604/cmc.2022.021669.

[20] G. Logeswari, S. Bose, and T. J. Anitha, "An intrusion detection system for sdn using machine learning," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 867–880, 2023. http://dx.doi.org/10.32604/iasc.2023.026769.

[21] M. Z. Mahmud, S. R. Alve, S. Islam, and M. M. Khan, "Sdn intrusion detection using machine learning method," *arXiv preprint arXiv:2411.05888*, 2024. https://doi.org/10.48550/arXiv.2411.05888.

[22] M. R. Hadi and A. S. Mohammed, "A novel approach to network intrusion detection system using deep learning for Sdn: Futuristic approach," *arXiv preprint arXiv:2208.02094*, 2022. https://doi.org/10.48550/arXiv.2208.02094.

[23] M. Maddu and Y. N. Rao, "Network intrusion detection and mitigation in SDN using deep learning models," *International Journal of Information Security*, vol. 23, no. 2, pp. 849–862, 2024. https://doi.org/10.1007/s10207-023-00771-2.

[24] S. Rohith, G. Logeswari, K. Tamilarasi, and G. Sudhakaran, "A federated deep learning approach for SDN security with quantum optimized feature selection and hybrid MSDC net architecture," *Sci. Rep.*, vol. 16, p. 8038, 2026. https://doi.org/10.1038/s41598-026-37289-1.

[25] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep learning approach for SDN-enabled intrusion detection system in IoT networks," *Information*, vol. 14, no. 1, p. 41, 2023. https://doi.org/10.3390/info14010041.

[26] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Engineering Journal*, vol. 14, no. 10, p. 102211, 2023. https://doi.org/10.1016/j.asej.2023.102211.

[27] A. Wani, R. S, and R. Khaliq, "SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL)," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 3, pp. 281–290, 2021. https://doi.org/10.1049/cit2.12003.

[28] T. L. Yasarathna and N. A. Le-Khac, "ASEADOS-SDN-IoT: A novel SDN-IoT network intrusion detection dataset and framework," *Internet of Things*, p. 101891, 2026. https://doi.org/10.1016/j.iot.2026.101891.

[29] A. O. Alzahrani and M. J. Alenazi, "ML-IDSDN: Machine learning based intrusion detection system for software-defined network," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 1, p. e7438, 2023. https://doi.org/10.1002/cpe.7438.

[30] K. Rui, H. Pan, and S. Shu, "Secure routing in the Internet of Things (IoT) with intrusion detection capability based on software-defined networking (SDN) and Machine Learning techniques," *Sci. Rep.*, vol. 13, no. 1, p. 18003, 2023. https://doi.org/10.1038/s41598-023-44764-6.

[31] G. Kumar and H. Alqahtani, "Machine Learning Techniques for Intrusion Detection Systems in SDN-Recent Advances, Challenges and Future Directions," *Computer Modeling in Engineering & Sciences (CMES)*, vol. 134, no. 1, 2023. https://doi.org/10.32604/cmes.2022.020724.