ML & DL - based DDoS Detection in SDN: Implementation and Comparative Analysis

Aarti Jadav[1], Parul Sharma[2]

[1]Computer Engineering, Silver Oak University, Gujarat, India.
[2]Information Technology, Silver Oak University, Gujarat, India.

Abstract:
Software Defined Networking (SDN) is a modern networking approach that separates the control plane from the data plane, allowing centralized control and better network management. However, the centralized SDN controller is highly vulnerable to Distributed Denial of Service (DDoS) attacks, where attackers send a large volume of malicious traffic to overload the controller and disrupt network services. Traditional DDoS detection methods are not effective in dynamic network environments because they rely on predefined rules and signatures, which makes it difficult to detect new and unknown attacks.
To overcome this limitation, Machine Learning (ML) and Deep Learning (DL) techniques are used to detect DDoS attacks by analyzing network traffic patterns. In this research, various ML algorithms such as Random Forest, Decision Tree, Support Vector Machine, and K-Nearest Neighbor, as well as DL models such as Convolutional Neural Network, Long Short-Term Memory, and Multi-Layer Perceptron, are implemented and compared for DDoS detection in SDN. The performance of these models is evaluated using metrics such as accuracy, precision, recall, F1-score, training time, and prediction time to identify the most efficient model for real-time DDoS detection.

Keywords: Software Defined Networking (SDN), DDoS Detection, Machine Learning Algorithms, Deep Learning Techniques, Network Security, Intrusion Detection System, Network Traffic Analysis.

I. INTRODUCTION

Software Defined Networking (SDN) is a modern networking architecture that separates the control plane from the data plane and provides centralized network control and programmability, which improves network management, scalability, and flexibility [1]. In SDN architecture, the controller manages the entire network by installing flow rules in forwarding devices, which simplifies network configuration and monitoring [2]. However, the centralized nature of SDN also introduces security challenges because the SDN controller acts as a single point of failure and can be targeted by various cyber-attacks [3].

Among various network attacks, Distributed Denial of Service (DDoS) attacks are considered one of the most dangerous attacks in SDN environments because they can overload the controller with a large number of malicious packets and disrupt network services [4]. In a DDoS attack, multiple compromised devices send a large volume of fake traffic to the target controller or server, which consumes bandwidth, memory, and processing resources, making the service unavailable to legitimate users [5]. These attacks can occur in different forms such as TCP SYN flood, UDP flood, ICMP flood, and HTTP flood attacks, which makes detection more difficult in SDN networks [6].

Traditional DDoS detection techniques such as signature-based detection and rule-based intrusion detection systems are not effective in detecting new and unknown attacks because these methods rely on predefined attack signatures and cannot adapt to dynamic network traffic patterns [7]. To overcome these limitations, Machine Learning (ML) techniques are widely used for DDoS detection because they can learn traffic behavior from network data and classify traffic into normal and attack categories [8]. Machine Learning algorithms such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbor (KNN) are commonly used for network traffic classification and DDoS detection [9].

In recent years, Deep Learning (DL) techniques have been introduced for DDoS detection to improve detection accuracy and automatically extract important features from network traffic data [10]. Deep Learning models such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) are capable of detecting complex attack patterns and time-based traffic behavior in SDN networks [11]. These models perform better than traditional Machine Learning models when large-scale network traffic data is used for training and testing [12].

Therefore, this research focuses on the implementation and comparative analysis of Machine Learning and Deep Learning algorithms for DDoS detection in Software Defined Networking. The performance of different ML and DL models is evaluated using performance metrics such as accuracy, precision, recall, F1-score, training time, and prediction time to identify the most efficient model for real-time DDoS detection in SDN environments.

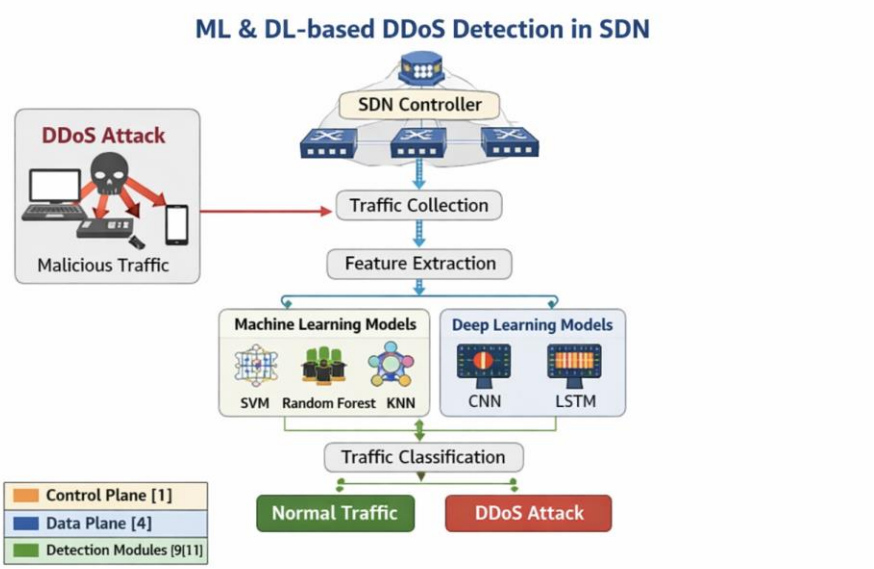

Fig 1: Architecture of ML and DL-based DDoS Detection in SDN [1]

II. LITERATURE REVIEW

Software Defined Networking (SDN) has emerged as a modern networking paradigm that separates the control plane from the data plane, providing centralized control, flexibility, and programmability in network management [1]. However, the centralized architecture of SDN makes the controller a primary target for Distributed Denial of Service (DDoS) attacks, which can overwhelm the controller by generating a large volume of malicious traffic and disrupt network services [2]. To address this issue, many researchers have proposed Machine Learning (ML)-based intrusion detection systems that analyze network traffic patterns and classify traffic as normal or attack traffic based on statistical features such as packet count, byte count, and flow duration [3], [4].

Traditional Machine Learning algorithms such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbor (KNN) have been widely used for DDoS detection in SDN environments due to their high detection accuracy and low false positive rate [5]. These algorithms work by training classification models using labeled network traffic datasets and then predicting whether incoming traffic is normal or malicious [6]. Some researchers have also proposed hybrid Machine Learning approaches that combine multiple classifiers to improve detection performance and reduce classification errors [7].

In recent years, Deep Learning (DL) techniques have been widely used for DDoS detection because they can automatically extract important features from large-scale network traffic data and detect complex attack patterns [8]. Deep Learning models such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Autoencoders have shown better performance compared to traditional Machine Learning algorithms in detecting DDoS attacks [9]. CNN models are mainly used for

spatial feature extraction, while LSTM models are effective in analyzing time-series traffic data and detecting sequential attack behavior [10].

Several datasets are used for training and testing DDoS detection models in SDN, such as CICIDS2017, CICDDoS2019, NSL-KDD, and CAIDA datasets, which contain both normal and attack traffic data [11]. Feature selection plays an important role in improving detection accuracy and reducing computational complexity, and commonly used features include flow duration, packet rate, byte rate, and source-destination traffic patterns [12]. From the literature, it is observed that Machine Learning models provide faster detection with lower computational cost, while Deep Learning models provide higher accuracy for large and complex datasets. Therefore, a comparative analysis of ML and DL models is necessary to identify the most efficient model for DDoS detection in SDN environments.

III. METHODOLOGY

This research proposes a Machine Learning (ML) and Deep Learning (DL)-based Distributed Denial of Service (DDoS) detection system in Software Defined Networking (SDN). The proposed system is designed to detect DDoS attacks by analyzing network traffic collected from the SDN controller and classifying it as normal or attack traffic using ML and DL algorithms [1]. The SDN architecture separates the control plane from the data plane, where the SDN controller manages the network traffic and flow rules, making it easier to monitor and analyze traffic behavior for attack detection [2]. However, the centralized nature of SDN also makes it vulnerable to DDoS attacks, where attackers send a large amount of malicious traffic to overload the controller and disrupt network services [3].

In the first stage of the proposed methodology, network traffic is collected from the SDN environment through OpenFlow switches and forwarded to the SDN controller [4]. The controller collects flow statistics such as source IP address, destination IP address, flow duration, packet count, and byte count, which are used for traffic analysis and attack detection [5]. The dataset used in this research is CICIDS2017/CICDDoS2019, which contains both normal and DDoS attack traffic and is widely used for intrusion detection research [6]. After traffic collection, data preprocessing is performed to remove missing values, duplicate records, and irrelevant features from the dataset to improve data quality and model performance [7]. Data normalization is also applied to scale feature values into a standard range so that the ML and DL models can be trained efficiently [8].

In the next stage, feature extraction and feature selection techniques are applied to select the most relevant features for DDoS detection [9]. Important features such as flow duration, total packets, total bytes, packets per second, and bytes per second are selected because DDoS attacks generate abnormal traffic patterns compared to normal traffic [9]. Feature selection helps reduce computational complexity and improves model accuracy by removing unnecessary features [10]. After feature selection, the dataset is divided into training and testing datasets, where 70% of the data is used for training the model and 30% is used for testing the model performance [10].

In the model training phase, Machine Learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), and Random Forest (RF) are used to classify network traffic [11]. These ML algorithms classify traffic based on statistical features and provide fast detection with good accuracy [11]. In addition, Deep Learning models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are used to detect complex and time-based attack patterns in network traffic [12]. CNN is mainly used for feature learning and pattern recognition, while LSTM is used for time-series analysis and sequential traffic behavior detection [12].

Finally, the trained ML and DL models are tested using the testing dataset, and performance metrics such as Accuracy, Precision, Recall, and F1-Score are calculated to evaluate the performance of the models [10]. The performance of ML and DL models is then compared to determine which model provides better accuracy and detection performance for DDoS detection in SDN environments [12]. The proposed methodology improves DDoS detection accuracy and reduces false positive rates, which helps in improving overall network security.

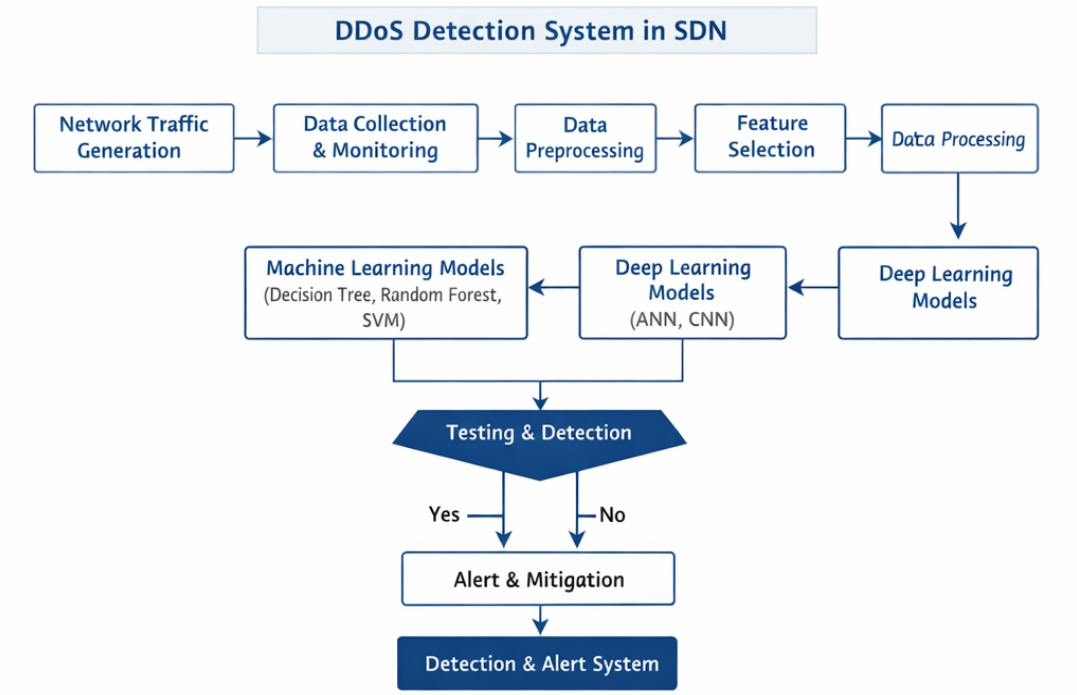

Fig 2: System Flow Diagram[1]

A. Dataset Overview and Description

The dataset used in this research includes both normal and DDoS attack traffic. Three datasets are used: Metasploitable-2 (metasploitable-2.csv), Normal_data.csv, and OVS.csv. The Metasploitable-2 dataset contains attack-related traffic, the Normal dataset includes only legitimate traffic, and the OVS dataset represents traffic from an SDN environment with both normal and attack activities. Each dataset consists of network flow features such as source and destination IP, ports, protocol type, flow duration, number of packets, total bytes, and traffic rate. These features help in identifying traffic behavior and detecting abnormal patterns related to DDoS attacks. Before training, preprocessing steps like data cleaning, normalization, encoding, and data balancing are applied. The dataset is then divided into training and testing sets, where training is used for building the model and testing is used to evaluate the performance of the DDoS detection system.

B. Data Analysis

Data analysis is performed to understand network traffic characteristics and distinguish between normal and DDoS attack traffic. Statistical and visualization techniques are used to examine feature distribution and traffic behavior. Key features such as flow duration, packet count, byte count, packet rate, and protocol type help identify abnormal patterns, as DDoS attacks typically generate high traffic volume and packet rates within a short time.

Exploratory Data Analysis (EDA) is performed to examine data distribution, feature relationships, and class imbalance in the dataset. Visualization techniques such as correlation matrices, histograms, box plots, and scatter plots are used to understand patterns in the data. This process helps in selecting relevant features for training, thereby improving the performance of Machine Learning and Deep Learning models and enabling accurate DDoS detection in the SDN environment.

IV. RESULTS AND DISCUSSION

1. Confusion Matrix of Random Forest Classifier for DDoS Attack Detection

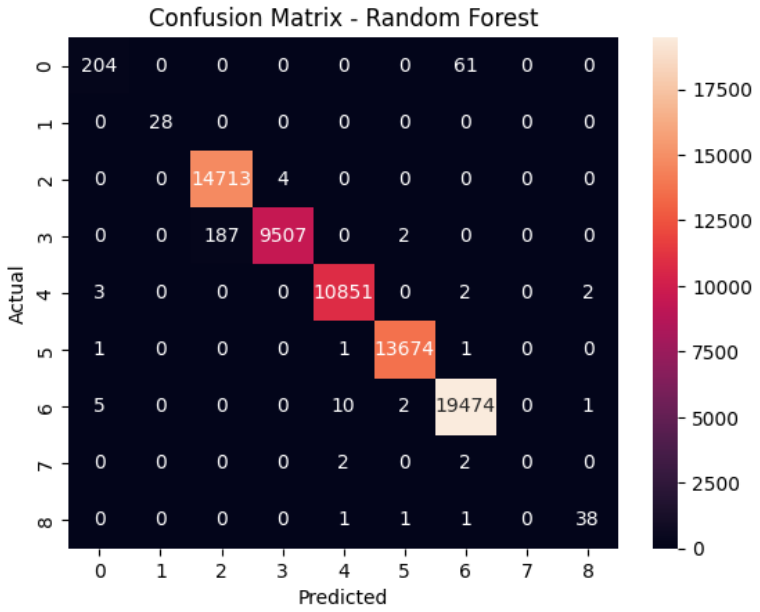

Fig 3: Random Forest Confusion Matrix for DDoS Detection

The confusion matrix shown in the figure represents the performance of the Random Forest classifier used for DDoS attack detection in the Software Defined Networking (SDN) environment. The matrix compares the actual class labels with the predicted class labels generated by the Random Forest model. Each row of the matrix represents the actual class, while each column represents the predicted class. The diagonal values in the matrix indicate correctly classified instances, whereas the off-diagonal values represent misclassified instances.

From the confusion matrix, it can be observed that the Random Forest model correctly classifies a large number of instances for most classes, as indicated by the high values along the diagonal. Only a small number of instances are misclassified into other classes, which indicates that the model has high accuracy and good classification performance. This confusion matrix demonstrates that the Random Forest algorithm is effective for detecting DDoS attacks and distinguishing between normal and attack traffic in the SDN-based network environment.

2. Accuracy Comparison of Machine Learning Models for DDoS Detection

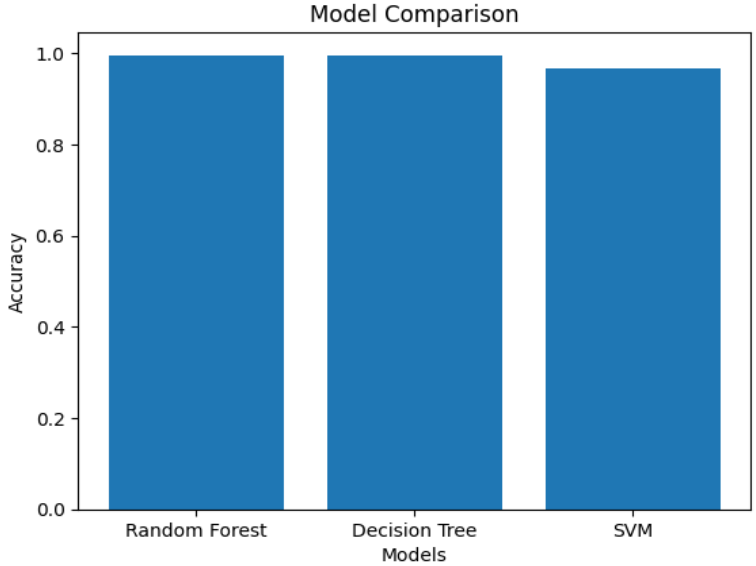

Fig 4: Model Accuracy Comparison for DDoS Detection

The bar plot illustrates the comparison of accuracy among the top three Machine Learning models used for DDoS attack detection, with the x-axis representing the models and the y-axis showing their accuracy values. The bars represent the performance of Random Forest, Decision Tree, and Support Vector Machine (SVM) models.
This visualization helps in identifying which model performs best in terms of accuracy for detecting DDoS attacks in the SDN environment. From the plot, it can be observed that the Random Forest and Decision Tree models achieve higher accuracy compared to the SVM model. This comparison helps in selecting the most suitable model for accurate and efficient DDoS detection, which can improve network security and reduce the impact of DDoS attacks.

3. Epoch-wise Accuracy and Loss of Proposed Model

The model was trained for 20 epochs, and the performance was evaluated using accuracy and loss metrics for both training and validation datasets.
From the results, it can be observed that the training accuracy steadily increased from approximately 98.20% in the initial epochs to 99.68% in the final epoch, indicating that the model effectively learned the underlying patterns in the dataset. Similarly, the validation accuracy remained consistently high, reaching around 99.66%, which shows strong generalization capability.

```
Epoch 1/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 30s 3ms/step - accuracy: 0.9820 - loss: 0.0734 - val_accuracy: 0.9905 - val_loss: 0.0786
Epoch 2/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9931 - loss: 0.0319 - val_accuracy: 0.9939 - val_loss: 0.0745
Epoch 3/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 28s 3ms/step - accuracy: 0.9949 - loss: 0.0250 - val_accuracy: 0.9952 - val_loss: 0.0722
Epoch 4/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 29s 3ms/step - accuracy: 0.9956 - loss: 0.0215 - val_accuracy: 0.9961 - val_loss: 0.0828
Epoch 5/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 38s 4ms/step - accuracy: 0.9959 - loss: 0.0213 - val_accuracy: 0.9960 - val_loss: 0.0655
Epoch 6/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9961 - loss: 0.0195 - val_accuracy: 0.9963 - val_loss: 0.0797
Epoch 7/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 30s 3ms/step - accuracy: 0.9962 - loss: 0.0192 - val_accuracy: 0.9957 - val_loss: 0.1069
Epoch 8/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 28s 3ms/step - accuracy: 0.9963 - loss: 0.0185 - val_accuracy: 0.9962 - val_loss: 0.1205
Epoch 9/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9963 - loss: 0.0188 - val_accuracy: 0.9960 - val_loss: 0.1351
Epoch 10/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 29s 3ms/step - accuracy: 0.9965 - loss: 0.0177 - val_accuracy: 0.9959 - val_loss: 0.1760
Epoch 11/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 44s 5ms/step - accuracy: 0.9965 - loss: 0.0179 - val_accuracy: 0.9962 - val_loss: 0.1728
Epoch 12/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9965 - loss: 0.0172 - val_accuracy: 0.9961 - val_loss: 0.2223
Epoch 13/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 41s 3ms/step - accuracy: 0.9966 - loss: 0.0173 - val_accuracy: 0.9960 - val_loss: 0.2498
Epoch 14/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 28s 3ms/step - accuracy: 0.9966 - loss: 0.0177 - val_accuracy: 0.9963 - val_loss: 0.2915
Epoch 15/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9966 - loss: 0.0169 - val_accuracy: 0.9964 - val_loss: 0.3426
Epoch 16/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 29s 3ms/step - accuracy: 0.9966 - loss: 0.0174 - val_accuracy: 0.9962 - val_loss: 0.3774
Epoch 17/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9967 - loss: 0.0175 - val_accuracy: 0.9966 - val_loss: 0.3873
Epoch 18/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 28s 3ms/step - accuracy: 0.9967 - loss: 0.0170 - val_accuracy: 0.9964 - val_loss: 0.4139
Epoch 19/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 28s 3ms/step - accuracy: 0.9967 - loss: 0.0165 - val_accuracy: 0.9965 - val_loss: 0.3771
Epoch 20/20
8598/8598 ━━━━━━━━━━━━━━━━━━━━ 27s 3ms/step - accuracy: 0.9968 - loss: 0.0163 - val_accuracy: 0.9966 - val_loss: 0.3969
```

Fig 5: Training and Validation Performance across Epochs

The training loss decreased significantly from 0.0734 to 0.0163, demonstrating that the model minimized errors effectively during training. Although the validation loss fluctuates slightly in later epochs, this is a common behavior and does not indicate severe overfitting, as validation accuracy remains stable.
Additionally, after around 8–10 epochs, the model performance starts to stabilize, with only marginal improvements in accuracy. This indicates that the model has converged and further training provides minimal gain.

4. Training and Validation Accuracy Curve of Deep Learning Model for DDoS Detection

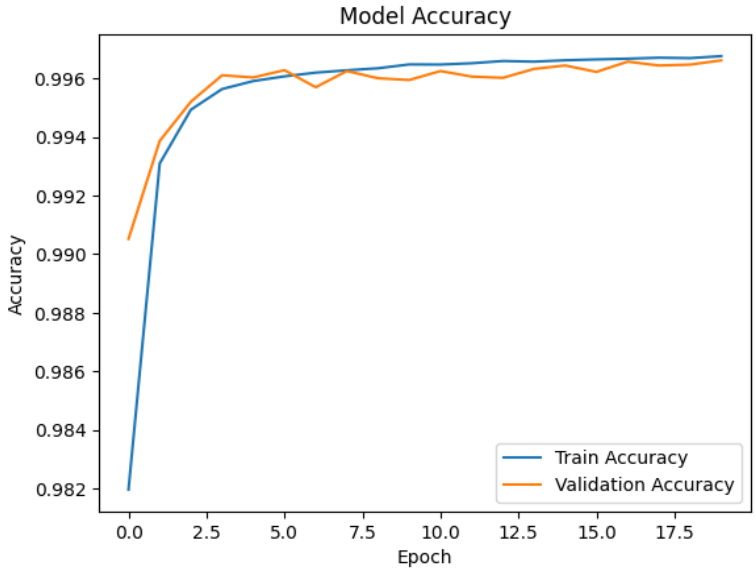

Fig 6: Training & Validation Accuracy Curve for DDoS Detection

The line graph illustrates the training and validation accuracy of the proposed Deep Learning model over multiple epochs, with the x-axis representing the number of epochs and the y-axis showing the accuracy values. Two curves are plotted, where one represents training accuracy and the other represents validation accuracy. The graph shows a rapid increase in accuracy during the initial epochs, followed by a gradual stabilization as the model continues to learn.

This visualization helps in evaluating the performance and learning behavior of the model. It can be observed that both training and validation accuracy remain very close to each other, indicating that the model is well-fitted and does not suffer from overfitting or underfitting.

The consistently high accuracy achieved across epochs demonstrates the effectiveness of the proposed model in accurately detecting DDoS attacks in the SDN environment.

Additionally, the smooth and consistent trend of both curves indicates stable learning without sudden fluctuations, which reflects the robustness of the training process. The minimal gap between training and validation accuracy suggests that the model generalizes well to unseen data.

This stability also implies that the selected hyperparameters and training configuration are appropriate, contributing to reliable and consistent performance in DDoS attack detection within the SDN environment.

5. Training and Validation Loss Curve of Deep Learning Model for DDoS Detection

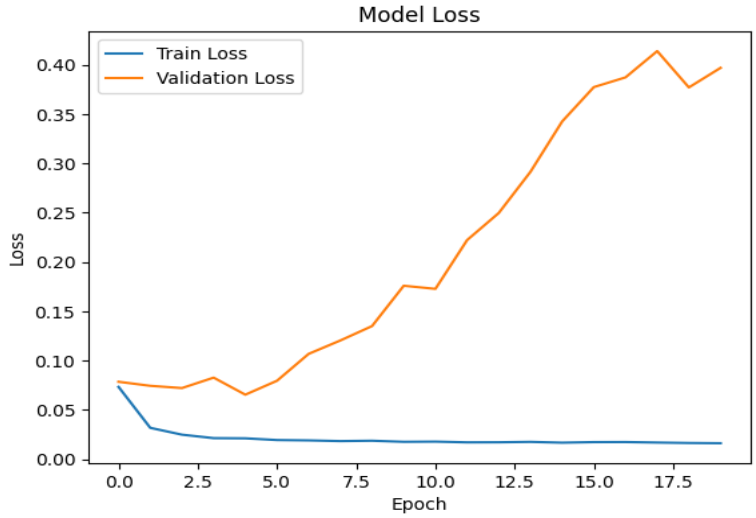

Fig 7: Training & Validation Loss Curve for DDoS Detection

The line graph represents the training and validation loss of the Deep Learning model across multiple epochs, where the x-axis indicates the number of epochs and the y-axis shows the loss values. Two curves are plotted to compare the training loss and validation loss during the learning process.
Initially, both losses start at a relatively higher value, but the training loss decreases rapidly and stabilizes at a very low value as the number of epochs increases.
In contrast, the validation loss shows a gradual increasing trend after a certain number of epochs, indicating that the model starts to overfit the training data. This divergence between training and validation loss suggests that while the model performs well on training data, its generalization performance on unseen data decreases over time.
This visualization helps in identifying overfitting and emphasizes the need for techniques such as regularization or early stopping to improve model performance.

6. Multiclass ROC Curve for Random Forest Classifier

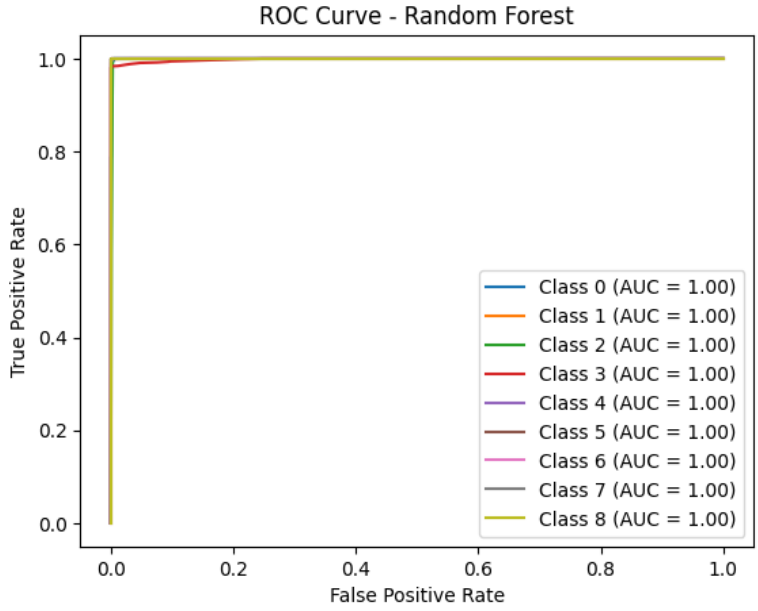

Fig 8: Multiclass ROC Curve (Random Forest)

The figure presents the Receiver Operating Characteristic (ROC) curves for a Random Forest model applied to a multiclass classification problem. Each curve corresponds to a specific class (Class 0 through Class 8), illustrating the relationship between the False Positive Rate (FPR) on the x-axis and the True Positive Rate (TPR) on the y-axis. All classes achieve an Area Under the Curve (AUC) score of 1.00, indicating perfect classification performance. The curves are tightly aligned along the top-left boundary of the plot, demonstrating that the model effectively distinguishes between all classes with minimal misclassification. This visualization highlights the robustness and high predictive accuracy of the Random Forest model across all categories.

7. Training and Validation Performance of the Model

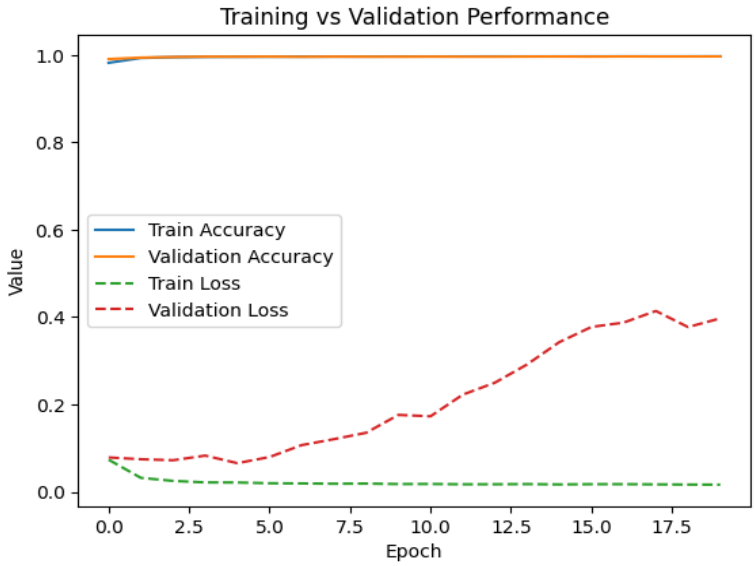

Fig 9: Model Training & Validation Performance

This figure shows how the model performs during training over multiple epochs. It compares training accuracy, validation accuracy, training loss, and validation loss. The training and validation accuracy both quickly reach values close to 1.0, indicating that the model is learning effectively. The training loss decreases steadily and stays very low, which means the model fits the training data well. However, the validation loss gradually increases over time, suggesting that the model may be starting to overfit the data. This means that while the model performs very well on training data, its performance on new or unseen data may not improve as much.

The consistently high training accuracy demonstrates that the model has the ability to learn complex patterns from the dataset. However, the increasing validation loss suggests that the model becomes overconfident in its predictions on unseen data. This means that even though accuracy remains high, the model's reliability on new data may not be optimal.
To improve performance, techniques such as dropout, early stopping, or regularization can be applied to reduce overfitting. Additionally, using more diverse data or improving data preprocessing can help the model generalize better. Monitoring both accuracy and loss is important, as accuracy alone may not fully reflect the true performance of the model.

8. Top 10 Important Features in the Model

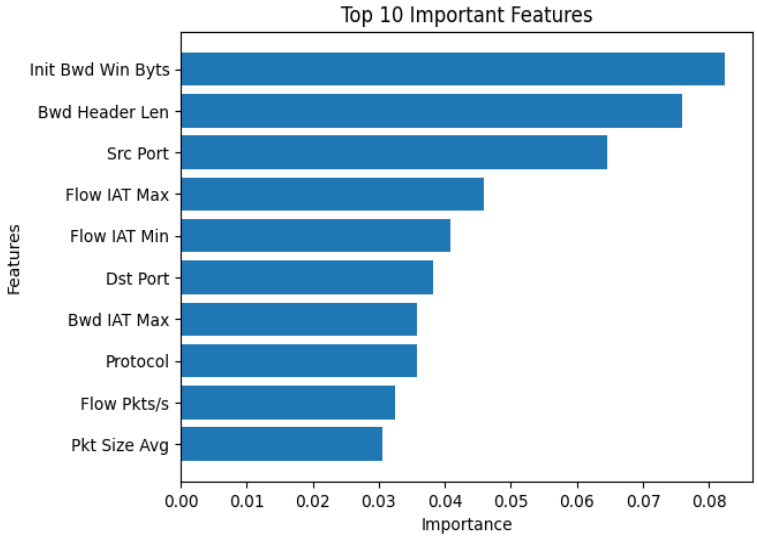

Fig 10: Top 10 Important Features

This chart highlights the ten most important features influencing the model's predictions, ranked by their contribution. The feature "Init Bad Win Byts" has the highest importance, followed by "Bwd Header Len" and "Src Port," indicating that these factors have the strongest impact on the model's decisions. Features such as "Flow IAT Max," "Flow IAT Min," and "Dst Port" show moderate influence, while the remaining features contribute slightly less but are still relevant.
Overall, the distribution shows that a few key features play a major role in prediction, while others provide additional support. This helps in understanding which variables are most important and can be useful for feature selection and model optimization.

V. Comparison of Base Paper and Proposed Method

"DDoS Detection in Software Defined Networking Using Machine Learning and Deep Learning Techniques" [5]

Algorithm	Base Paper Accuracy (%)	Proposed Model Accuracy (%)	Improvement (%)
Random Forest	95.10%	99.57%	+4.47
SVM	93.46%	96.71%	+3.25
Decision Tree	90.20%	99.55%	+9.35

Table 1: Accuracy Comparison of Base Paper [5] and Proposed ML-DL Model

IV. CONCLUSION

The integration of **Machine Learning (ML)** and **Deep Learning (DL)** techniques with **Software Defined Networking (SDN)** has significantly improved the detection and mitigation of Distributed Denial of Service (DDoS) attacks. Traditional security mechanisms struggle to handle the dynamic and large-scale nature of modern network attacks, whereas ML/DL-based approaches provide adaptive and intelligent solutions.
From the comparative analysis, it is observed that ML algorithms such as Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) offer efficient detection with lower computational complexity, making them suitable for real-time deployment in SDN environments [1][2]. On the other hand, DL models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Autoencoders achieve higher detection accuracy by learning complex traffic patterns, although they require more computational resources [3][4].

Furthermore, SDN architecture enhances DDoS detection by providing centralized control and global network visibility, enabling faster response and better traffic analysis [5]. Hybrid approaches combining ML and DL techniques have shown improved performance in terms of accuracy, precision, and false positive rate [6].

REFERENCES:

1. Gao, X., et al. (2025). *Distributed denial of service attack detection in software-defined networking using CNN-MLP*. PLOS ONE. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0312425
2. Tang, F., et al. (2025). *Optimizing DDoS detection in SDNs through machine learning techniques*. arXiv. https://arxiv.org/pdf/2505.13493
3. Zhang, Q., et al. (2025). *Collaborative P4-based DDoS detection using early-exit neural networks in SDN*. arXiv. https://arxiv.org/abs/2509.12291

4. Zhou, Y., et al. (2025). *Federated learning-based DDoS detection in SDN using GAN models*. arXiv. https://arxiv.org/abs/2503.14618
5. Ahmed, I., et al. (2024). *Analyzing machine learning and deep learning techniques for SDN-based DDoS attack detection: A comparative study*. DOAJ. https://doaj.org/article/46f3a8728f744ff3b6107aaa4fee4dda
6. Fatehi, A., & Montazerolghaem, A. (2024). *DDoS detection in SDN using deep learning*. ResearchGate. https://www.researchgate.net/publication/381819015
7. Kumar, S., et al. (2024). *Deep learning approaches for intrusion detection in SDN environments*. Springer. https://link.springer.com/article/10.1186/s42400-024-00219-7
8. Aladaileh, M., et al. (2023). *A comprehensive analysis of machine learning and deep learning-based solutions for DDoS attack detection in SDN*. ResearchGate. https://www.researchgate.net/publication/372302880
9. Khan, M. A., et al. (2023). *Comparison of machine learning and deep learning approaches for detecting DDoS attacks in SDN*. Applied Sciences (MDPI). https://www.mdpi.com/2076-3417/13/5/3033
10. Nayak, S., et al. (2023). *Detection of DDoS attacks in SDN using machine learning models*. arXiv. https://arxiv.org/abs/2303.06513
11. Patel, H., et al. (2022). *Machine learning techniques for DDoS detection in SDN: A survey*. ResearchGate. https://www.researchgate.net/publication/370442250
12. Singh, R., et al. (2022). *Hybrid machine learning approach for DDoS detection in SDN*. IJERT. https://www.ijert.org/research/hybrid-ml-ddos-detection
13. Aamir, M., & Zaidi, S. M. A. (2020). *Clustering-based semi-supervised machine learning for DDoS attack classification*. Journal of King Saud University. https://doi.org/10.1016/j.jksuci.2019.02.003
14. Alenezi, M., & Reed, M. (2020). *DDoSNet: A deep-learning model for detecting network attacks*. arXiv. https://arxiv.org/abs/2006.13981
15. Latah, M., & Toker, L. (2020). *Towards an efficient anomaly-based intrusion detection for software-defined networks*. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3019202