

Enhancing DDoS Detection in SDN-Based Autonomous Vehicles using Transformer-based Feature Transformation

Romold Perera
Faculty of Computing
Sri Lanka Institute of Information
Technology
Malabe, Sri Lanka
romoldperera58@gmail.com

Tharindu Sahan Senarathne
Faculty of Computing
Sri Lanka Institute of Information
Technology
Malabe, Sri Lanka
sahasenarathne@gmail.com

Tharin Ransika
Faculty of Computing
Sri Lanka Institute of Information
Technology
Malabe, Sri Lanka
tharinransika@gmail.com

Pasindu Wijewardhana
Faculty of Computing
Sri Lanka Institute of Information
Technology
Malabe, Sri Lanka
pasindu4888@gmail.com

Samadhi Chathuranga Rathnayake
Faculty of Computing
Sri Lanka Institute of Information
Technology
Malabe, Sri Lanka
samadhi.r@slit.lk

Nelum Chathuranga Amarasena
Faculty of Computing
Sri Lanka Institute of Information
Technology
Malabe, Sri Lanka
nelum.a@slit.lk

Abstract—Autonomous vehicles (AVs) require fast, reliable, and secure communication to function safely in dynamic environments. To meet these demands, Software-Defined Networking (SDN) is increasingly used in AV systems due to its centralized control and ability to manage network traffic efficiently. However, this centralization also makes SDN vulnerable to Distributed Denial-of-Service (DDoS) attacks, which can disrupt critical vehicle operations. While earlier studies have used Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) for DDoS detection, this work introduces a Transformer-based feature selection method that leverages self-attention to focus on the most relevant network traffic features. This improves both detection accuracy and response time. To further enhance performance, we apply Bayesian Optimization with HyperNet (BOHN) for tuning model parameters. For interpretability, SHapley Additive exPlanations (SHAP) is used to highlight the impact of each feature on the model's predictions. The model is trained on the InSDN dataset, which simulates real-world SDN attack scenarios, and tested on the UNSW-NB15 dataset, a comprehensive benchmark containing a mix of normal and malicious traffic. Experimental results show significant improvements in accuracy, false positive reduction, and adaptability to new types of threats, demonstrating the value of self-attention-based approaches for securing SDN-enabled AV networks.

Keywords—Autonomous Vehicles (AVs), Software-Defined Networking (SDN), DDoS Detection, Transformer Architecture, Bayesian Optimization, SHAP Interpretability

I. INTRODUCTION

The concept of Software Defined Networking (SDN) comes from earlier concepts which focused on the programmability of network and the concept of control data plane separation [1]. The architecture which contains the SDN controller centralizes control in this said SDN controller, which manages network traffic via protocols like OpenFlow. [1]. While SDN offers significant advantages in network

flexibility and management, its centralized nature introduces security vulnerabilities. Unlike traditional networks, where security breaches typically affect isolated nodes, a compromised SDN controller can disrupt the entire network. Attackers can exploit SDN's architecture to launch Distributed Denial-of-Service (DDoS) and other cyberattacks, leading to bandwidth saturation and resource exhaustion [1]. Intrusion Detection Systems (IDS) are commonly employed to mitigate these threats; however, the dynamic nature of SDN can introduce false positives, making effective detection a challenge [1].

Feature selection is a critical step in intrusion detection, enabling the removal of irrelevant or redundant features to enhance classification accuracy. There are two ways to enable feature selection methods either using filter based methods such as chi square, ANOVA or to use wrapper based methods recursive selection forward elimination etc [2]. Although wrapper methods provide highly relevant feature subsets, they are computationally expensive [3]. The increasing complexity of network traffic necessitates robust intrusion detection mechanisms, and deep learning (DL) has emerged as a powerful solution [3]. However, most researches solely have relied on datasets such as NSL-KDD and KDD CUP, which are mostly general purpose and these datasets do not fully capture the SDN characteristics present within the environment [1].

Our approach enhances Network Intrusion Detection Systems (NIDS) for SDN-based autonomous vehicles by utilizing a CNN-BiLSTM model with transformer based feature transformation. Unlike previous studies that detected DDoS using CNN-LSTM models [4], Blockchain based Machine learning models [5], BiLSTM processes sequences bidirectionally, preserving contextual information. To address limitations related to data redundancy and feature

selection, we introduce a Transformer-based feature selection mechanism that leverages self-attention to prioritize important traffic features dynamically. Additionally, we employ SHapley Additive exPlanations (SHAP) for model interpretability. Our model is trained on the InSDN dataset, specifically designed for SDN security, and evaluated on UNSW-NB15 to assess generalizability.

II. RELATED WORK

Deep learning, which is a branch of machine learning, has been effectively utilized across multiple fields, such as image analysis recognition in various industries and natural language understanding. Lately, deep learning methods have attracted interest within the research of network intrusion detection systems (NIDS), especially CNN and LSTM models. This is due to their ability to capture the spatial features of data as well as the temporal dependencies that are very complex to detect otherwise.

Elsayed et al. paper discusses about their approach where they combined the use of CNNs and LSTMs to develop a mechanism for detecting intrusions, utilizing their capabilities to identify both spatial and sequential patterns from network traffic data [6]. In order to avoid overfitting, the authors used L2 regularization and dropout methods [6]. Their research employed the InSDN dataset, a newly released benchmark for network intrusion detection systems in software-defined networking environments [7]. The model introduced in this literature also showed improved performance against unknown attacks also known as zero day vulnerabilities, achieving an accuracy of 96.32% [6].

A CNN and BiLSTM model was presented by Kaiyuan et al. which uses both the hybrid deep learning as well as hybrid feature selection methods. They used the Synthetic Minority Oversampling Technique (SMOTE) to increase the amount of minority class samples after initially applying One-Side Selection (OSS) to remove noise from the majority class [8]. By balancing the dataset, our method shortened training time and allowed the machine to efficiently learn characteristics from minority samples. In the second phase, they used a BiLSTM to identify temporal correlations in the data and a CNN to extract spatial characteristics [1]. The approach implemented by them showed an accuracy of 93.51 % on UNSW-NB15 dataset, 97.77% on the NSL KDD dataset and 97.77% on InSDN three most popular datasets [1].

Another hybrid approach was presented by Ben Said et al. where they combined CNN-BiLSTM with an attention mechanism [3]. Their approach leverages CNN for local feature extraction from network traffic, BiLSTM for capturing temporal dependencies, and an attention layer to prioritize relevant features. Evaluated on the InSDN dataset, their model achieved an accuracy of 98.03%, outperforming traditional intrusion detection models like AlexNet (75.82%), LeNet5 (84.89%), CNN (91.05%), and CNN-BiLSTM without attention (97.26%) [3]. The method they proposed was very effective in detecting intrusions in SDN environments [3].

A machine learning based approach was also introduced by Ashwin et al. where they integrated Particle Swarm Optimization (PSO) and Generalized Normal Distribution Optimization (GNDO) for feature selection, optimizing the detection process [9]. They also tested out Support Vector Machine (SVM) and Decision Tree (DT) classifier methods as

well on the SDN_DDoS_2020 dataset, focusing on key network traffic features. However from the models they evaluated the highest performing model was the PSO-Decision Tree method which achieved a high accuracy of 98.87%. They have also mentioned that the False Alarm Rate (FAR) was 0.7702. This combined with the achieved high accuracy solidifies this method as a very good model to detect DDoS attacks in SDN environments [9].

The paper by Mengxue Li et al. introduced another novel approach where they used a hybrid CNN and LSTM networks for DDoS attack detection in SDN environments [10]. The CNN was used to extract spatial features and the LSTM was used to capture the time based components. This in turn enhances the detection accuracy of various attack patterns [10]. The model was trained on UNB CICIDS2018, achieving an accuracy of 96.13% in detecting DDoS attacks. Through comparative analysis, their method demonstrated improved performance over traditional techniques, showcasing its potential for real-time intrusion detection in SDN networks [10].

Table 1 below provides a summarized overview of the various methods employed by different researchers in this domain.

TABLE I SUMMARY OF PAST RESEARCH

Ref	Method	Number of selected features	Binary class classification n highest accuracy achieved	Type of network
[10]	CNN-LSTM	Not Reported	96.13% on UNB CICIDS 2017	SDN
[3]	Attention Based CNN-BiLSTM	Not reported	98.13% on InSDN, for multiclass classification	SDN
[1]	CNN-BiLSTM	10	93.51% on UNSW-NB15, 95.96% on NSL-KDD, 97.77% on InSDN	SDN
[8]	RNN	Not reported	97.39% NSL-KDD	SDN
[9]	SVM, PSO_DT	Not Reported	98.87% on SDN_DDoS 2020	SDN
[15]	Hybrid ANN, SVM and Random Forest	Not reported	Not reported	SDN

III. METHODOLOGY

A. Dataset

The dataset to train our model is the InSDN dataset. This is recognized as one of the earliest and most comprehensive datasets designed to evaluate Intrusion Detection Systems (IDSs) in Software-Defined Networking (SDN) environments. Using a dataset which is based on SDN environment is crucial for accurately evaluating threat detection techniques tailored to SDN. Datasets not designed for SDN may introduce compatibility issues, as attack implementations must consider the unique architecture of SDN networks [12]. Compared to traditional attacks SDN specific attacks has different behaviors involved. “IPsweep”, “Portscan” are two popular methods used to flood controller of the SDN environment data packets. This is enough to cause a DDoS attack most of the time [12]. Since these are not classified as DDoS attacks most of the time, they can still generate a significant volume of packets within an SDN environment. The InSDN dataset captures a wide range of attack patterns, providing valuable insights into these scenarios [12]. Figure 1 illustrates the distribution of the dataset, highlighting the proportion of DDoS versus benign traffic.

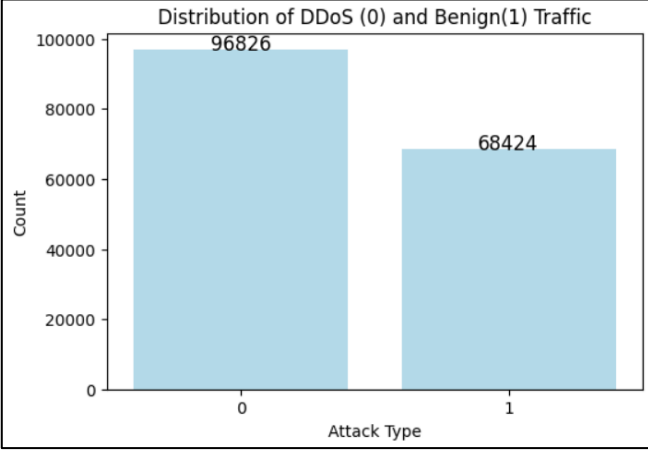


Fig. 1: Distribution of Data for DDOS and BENIGN data in the InSDN dataset

To generalize our model we also trained and tested it on the UNSW-NB15 dataset. This dataset was also used in various research based on Intrusion detection and is vastly regarded as one of the best datasets for this purpose. Additionally, past studies have also frequently utilized this publicly available intrusion detection dataset, to ensure the generalizability of models. In our case, while we used the InSDN dataset for performance evaluation, we focused on the UNSW-NB15 dataset to validate our model further solidifying the model architecture introduced in our research.

B. Data preprocessing

Data preprocessing, also known as data engineering, plays an important role to ensure the effectiveness of the learning process. It involves multiple steps, including cleaning rows and columns, encoding categorical features, and normalizing data. These preprocessing techniques were applied to all datasets to ensure they were adequately prepared for analysis. The key steps are detailed below.

1) Handling Missing Values and Dropping Unnecessary Columns

To maintain data integrity, all rows were carefully examined for missing values. While this is one of the standard practices in data preprocessing this is also a crucial step to make a very good performing model. Additionally, columns that do not contribute meaningful information, such as the "id" column, were removed to retain only essential features for analysis.

2) Encoding Categorical Features with LabelEncoder

Transforming categorical features into numerical values is crucial for improving the learning capability of classifiers that only process numerical data. There were some categorical attributes as well in the InSDN dataset such as "proto," "service" and "state". These were converted into numerical representations to facilitate analysis.

To transform the values into numerical representations we have used LabelEncoder technique. This converts categorical variables into a numerical form. This method was also utilized by previous researchers when they used the InSDN dataset. There were other methods such as One-Hot Encoding but that method only converts to binary columns. We decided it is not suitable for this research due to that. Also Label Encoding preserves the categorical structure while efficiently representing the data.

C. Novel Feature Selection and Feature Transformation

Our model presents a novel hybrid architecture that integrates CNN-BiLSTM layers with a Transformer encoder module, optimized for feature transformation in network intrusion detection. While CNNs capture local spatial patterns and BiLSTMs model temporal dependencies, the inclusion of the Transformer encoder—prior to the BiLSTM—provides a unique mechanism for early-stage feature refinement. Specifically, the Multi-Head Self-Attention (MHSA) in the Transformer dynamically weighs the relationships between features across the entire input sequence, thereby enhancing the quality of information passed into the BiLSTM layer.

In contrast to conventional CNN-LSTM or CNN-BiLSTM models with attention mechanisms applied at the output stage, our Transformer encoder operates at the feature extraction phase, allowing redundancy and noise reduction before temporal modeling occurs. This differentiates our method by addressing feature relevance earlier in the pipeline, which we empirically observe to improve generalization and reduce overfitting. The architecture proceeds with a second BiLSTM layer following an attention mechanism, which further focuses on the most relevant time steps from the sequence. Finally, a dense layer and sigmoid output produce a binary classification output suitable for intrusion detection.

Compared to prior hybrid deep learning approaches, our contribution lies in the deliberate use of MHSA for feature selection, rather than time-step attention alone, enabling the model to make better use of its temporal encoding by being fed a cleaner, contextually weighted feature representation. This results in enhanced performance, both in accuracy and interpretability, which we demonstrate through empirical evaluation and SHAP-based feature importance analysis. Given an input feature sequence X , the attention mechanism computes query (Q), key (K), and value (V) matrices as follows:

$$Q = XW_Q, K = XW_K, V = XW_V \quad (1)$$

where W_Q, W_K, W_V are learnable weight matrices. The dot product attention formula is then used to calculate the attention scores.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \quad (2)$$

Where d_K is the dimensionality of the key vectors. The above mechanism allows the model to focus on the most significant features while suppressing noisy or less relevant ones.

After computing the attention scores, a feed-forward network (FFN) is applied to refine the feature representation:

$$MultiHead(X) = Concat(head_1, \dots, head_h)W_O$$



Fig. 2: Multi head self attention Visualization

where h is the number of attention heads, and W_O is a learnable output weight matrix.

After the attention layer, a feed-forward network (FFN) is applied to refine the feature selection:

$$FFN(x) = \sigma(W_1x + b_1)W_2 + b_2 \quad (3)$$

where:

- W_1, W_2 are learnable weight matrices,
- b_1, b_2 are biases,
- σ is a ReLU activation function.

This allows the model to focus on the most important features while maintaining gradient stability.

Figure 2 illustrates the Multi-Head Self-Attention (MHSA) visualization from the Transformer encoder component of our proposed CNN-BiLSTM-Transformer model for DDoS detection in SDN-based autonomous vehicles.

In this heatmap, the x-axis represents the different attention heads, and the y-axis corresponds to time steps in the input sequence. The color intensity reflects the attention weight distributions across these heads and time steps. This visualization reveals how each attention head focuses on different temporal patterns and relationships in the feature-transformed network traffic data. The diverse activation patterns across heads demonstrate the model's ability to capture both short-term and long-term dependencies, which is crucial for identifying complex DDoS attack behaviors that may evolve over time.

D. Integration with CNN-BiLSTM

The CNN block consists of a 1D convolution layer, followed by Batch Normalization and MaxPooling, which reduces the dimensionality of the feature space while preserving critical patterns. By extracting meaningful spatial representations, the CNN helps enhance the quality of the input data before passing it to the next stage.

Following the CNN block, the extracted features are input into a Bidirectional Long Short-Term Memory (BiLSTM) network, which captures long-term dependencies across both forward and backward directions, making it well-suited for sequential network traffic data. To further refine the temporal features, an attention mechanism is applied after the BiLSTM layer, dynamically assigning weights to different time steps and emphasizing the most informative parts of the sequence. Once spatial and temporal features are extracted and refined, they are passed through a fully connected dense layer with a sigmoid activation function for binary classification. Multiple dropout layers with a probability of 0.3 are incorporated throughout the model to prevent overfitting, while Batch Normalization is used to stabilize training and improve generalization. Early stopping is also implemented to halt training once validation loss ceases to improve, further enhancing model robustness.

In addition to the architectural design, Bayesian Optimization with HyperNet (BOHN) is integrated into the training process to optimize hyperparameter selection. The HyperNetwork is employed to rapidly generate architectural hyperparameters (such as filter sizes, dropout rates, and LSTM units) from a latent representation proposed by the Bayesian Optimization algorithm. This indirect encoding allows flexible and efficient exploration of the hyperparameter space. The search process is guided by a surrogate probabilistic model, typically a Gaussian Process, and an acquisition function to balance exploration and

exploitation when suggesting new hyperparameter candidates. During this search, the validation loss is used as the primary criterion for hyperparameter selection, with additional monitoring of accuracy and F1-score to ensure generalizability and balanced performance.

E. Implementaion Details

The model was implemented and trained using the Kaggle notebook environment, equipped with NVIDIA GPU P100 which facilitated efficient computation. The training process spanned approximately 5 minutes, employing a batch size of 32 was used for training over 60 epochs. The learning rate was set to 0.001, and optimization was performed using the Adam optimizer. Using the accuracy as the primary performance metric we have also used the binary cross entropy loss function as well.

IV. RESULTS AND DISCUSSION

The below Table 2 represents the other methods we have tested for binary classification. The effectiveness of the proposed feature selection method was evaluated against conventional feature selection techniques, as well as several hybrid feature selection approaches. These included M1 scores combined with Recursive Feature Elimination (RFE), Principal Component Analysis (PCA) integrated with K-means clustering, M1 scores utilized alongside Least Absolute Shrinkage and Selection Operator (LASSO), and L1-regularized logistic regression. The comparative study highlights the relative performance of these techniques in enhancing classification accuracy and feature optimization.

TABLE II COMPARATIVE ANALYSIS WITH OTHER METHODS TESTED FOR HYBRID FEATURE SELECTION

Method	Model Accuracy
Previous study which used Random Forest Classifier with Recursive feature elimination	97.77% on InSDN , 93.51% on UNSW-NB15
PCA-K Means	93% on InSDN , 85.44% on UNSW-NB15
M1 Scores with Lasso	97.15% on InSDN , 90% on UNSW-NB15
Proposed study Transformer based attention mechanism	99.36% on InSDN , 95.43% on UNSW-NB15

For the evaluation of our proposed model CNN-BiLSTM with Transformer Encoder we employed standard performance metrics, namely Precision, Recall, and F1 Score, to ensure a comprehensive assessment of the model's effectiveness. Our approach achieved outstanding results,

with a Recall of 99.38%, Precision of 99.30%, and an F1 Score of 99.34%. Additionally, the model demonstrated high accuracy, achieving 99.36% on the InSDN dataset and 95.43% on the UNSW-NB15 dataset.

In our research, we employed SHapley Additive exPlanations (SHAP) to interpret and validate the feature importance within our deep learning-based intrusion detection model. SHAP assigns each feature an importance value for a particular prediction, offering both global and local interpretability of complex models. Given the black-box nature of architectures such as CNN-BiLSTM combined with Multi-Head Self-Attention (MHSA), SHAP provides a critical bridge between model accuracy and transparency particularly vital in security-sensitive environments like Software-Defined Networks (SDNs) in autonomous vehicles.

Figure 3 displays the mean SHAP values across all samples, offering a global perspective on feature relevance. Features such as Protocol, SrcPort, and Timestamp consistently ranked as the most impactful, supporting their integration into lightweight, real-time detection pipelines. These insights not only validate the feature engineering phase but also inform the development of rule-based heuristics and resource efficient deployment strategies essential for low-latency environments found in autonomous vehicular networks.

Figure 4 presents the SHAP summary plot, which captures the distribution and directionality of feature impact on the model's output. Notably, features such as Protocol, SrcPort, and Timestamp demonstrate significant contributions toward DDoS detection. For instance, higher SHAP values for Protocol associated with UDP traffic align with common DDoS strategies like UDP flooding. Similarly, abnormal variations in SrcPort reflect spoofing behavior typical of volumetric attacks. The Down/UpRatio further highlights asymmetric traffic patterns, often indicative of reflective or amplification attacks. These patterns provide actionable insights that can guide threshold tuning, feature prioritization, and real-time anomaly detection in production environments.

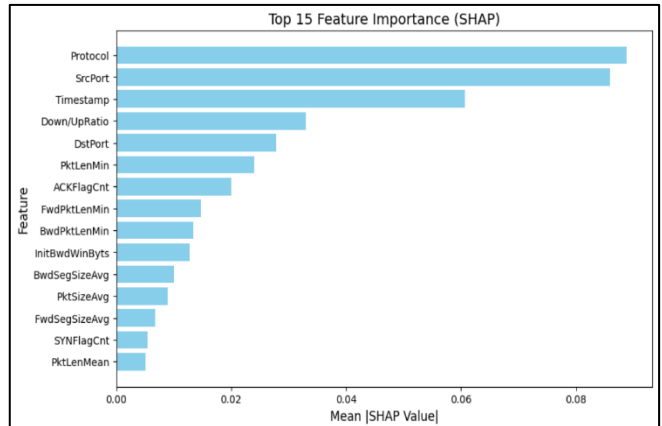


Fig. 3: Mean SHAP Values to visualize feature importance

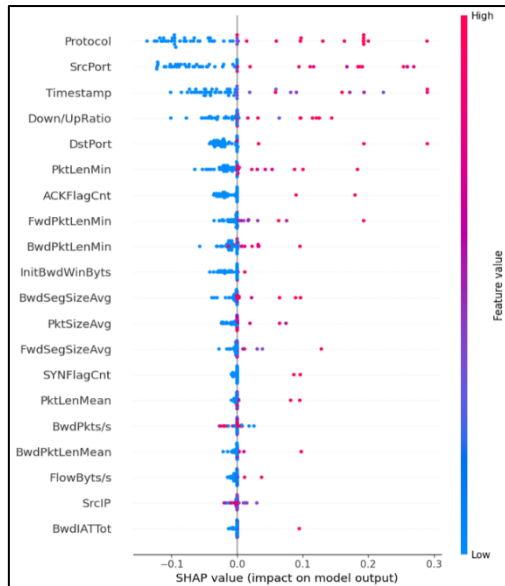


Fig. 4: SHAP summary plot

We also plotted the mean SHAP values to visualize feature importance. SHAP values ensure that each feature is appropriately considered when determining its impact on the model's predictions

V. CONCLUSION

This study presents a Transformer-based feature selection mechanism to enhance DDoS detection in SDN-based autonomous vehicles. By leveraging self-attention mechanisms, the model effectively prioritizes important features, reducing false positives. The integration of CNN, BiLSTM, and Transformer-based feature transformation enables the architecture to outperform traditional approaches in terms of accuracy and generalization. These findings highlight the value of attention mechanisms in capturing complex temporal and spatial dependencies in network traffic data. However, the model's high computational complexity results in training times exceeding one hour in some cases, particularly due to the overhead introduced by Bayesian Optimization with Hypernet. This poses a challenge for scalability and real-time applications. Future research should explore model compression, lightweight architectures, or faster tuning methods to reduce computational cost while maintaining the high performance achieved by the current model.

VI. ACKNOWLEDGEMENT

Our sincere gratitude goes to our supervisor, Mr. Samadhi Rathnayake, for his support throughout our research project. Additionally, we would like to thank Mr. Nelum Amarasena, our co-supervisor, for guiding us with new methodologies and pointing out our mistakes.

REFERENCES

- [1] R. Ben Said, Z. Sabir and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," in *IEEE Access*, vol. 11, pp. 138732-138747, 2023, doi: 10.1109/ACCESS.2023.3340142
- [2] A. Ojha, A. Yadav and V. Singh, "SDN-DDoSNet: A Deep Learning Framework for DDoS Attack Detection in Software-Defined Networks," *2024 IEEE 8th International Conference on Information and Communication Technology (CICT)*, Prayagraj UP, India, 2024, pp. 1-6, doi: 10.1109/CICT64037.2024.10899627.
- [3] R. B. Said and I. Askerzade, "Attention-Based CNN-BiLSTM Deep Learning Approach for Network Intrusion Detection System in Software Defined Networks," *2023 5th International Conference on Problems of Cybernetics and Informatics (PCI)*, Baku, Azerbaijan, 2023, pp. 1-5, doi: 10.1109/PCI60110.2023.10325985.
- [4] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147-167, 2019.
- [5] D. V. V. S. Manikumar and B. U. Maheswari, "Blockchain Based DDoS Mitigation Using Machine Learning Techniques," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020, pp. 794-800, doi: 10.1109/ICIRCA48905.2020.9183092.
- [6] M. Abdallah, N. A. Le Khac, H. Jahromi, and A. D. Jurcut, "A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs," in *Proc. 16th Int. Conf. Availability, Reliability and Security (ARES)*, Aug. 2021, pp. 1-12, doi: 10.1145/3465481.3469.
- [7] 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, doi: 10.1109/ACCESS.2020.3022633.
- [8] M. W. Nadeem, H. G. Goh, Y. Aun and V. Ponnusamy, "A Recurrent Neural Network based Method for Low-Rate DDoS Attack Detection in SDN," *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, IPOH, Malaysia, 2022, pp. 13-18, doi: 10.1109/AiDAS56890.2022.9918802.
- [9] A. Ashwin, V. Santhosh, R. Thirupathi Venkatesh, N. Gowthami and S. Tamilselvi, "Detection and Mitigation of DDoS Attack in SDN Using Feature Based SVM and Decision Tree Approach," *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, Bengaluru, India, 2024, pp. 325-330, doi: 10.1109/ICICNIS64247.2024.10823201.
- [10] M. Li, B. Zhang, G. Wang, B. ZhuGe, X. Jiang and L. Dong, "A DDoS attack detection method based on deep learning two-level model CNN-LSTM in SDN network," *2022 International Conference on Cloud Computing, Big Data Applications and Software Engineering (CBASE)*, Suzhou, China, 2022, pp. 282-287, doi: 10.1109/ICICNIS64247.2024.10823201.
- [11] N. Giri, R. Jaisinghani, R. Kriplani, T. Ramrakhyani and V. Bhatia, "Distributed Denial Of Service (DDoS) Mitigation in Software Defined Network using Blockchain," *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019, pp. 673-678, doi: 10.1109/I-SMAC47947.2019.9032690.
- [12] J. E. Varghese and B. Muniyal, "Trend in SDN Architecture for DDoS Detection- A Comparative Study," *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Nitte, India, 2021, pp. 170-174, doi: 10.1109/DISCOVER52564.2021.9663589.
- [13] B. Nugraha and R. N. Murthy, "Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks," *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Leganes, Spain, 2020, pp. 51-56, doi: 10.1109/NFV-SDN50289.2020.9289894.
- [14] L. Zhang, J. Huang, Y. Zhang, and G. Zhang, "Intrusion detection model of CNN-BiLSTM algorithm based on mean control," in *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*, 2020: IEEE, pp. 22-27.
- [15] G. Kaur and P. Gupta, "Hybrid Approach for detecting DDOS Attacks in Software Defined Networks," *2019 Twelfth International Conference on Contemporary Computing (IC3)*, Noida, India, 2019, pp. 1-6, doi: 10.1109/IC3.2019.8844944.
- [16] M. Abdallah, N. A. Le Khac, H. Jahromi, and A. D. Jurcut, "A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs," in *Proc. 16th Int. Conf. Availability, Reliability and Security (ARES)*, Aug. 2021, pp. 1-12, doi: 10.1145/3465481.3469