Feature Selection and 1DCNN-based DDOS Detection in Software-Defined Networking

Noor Almi'ani, Mohammed Anbar*, Shankar Karuppayah, Yousef Sanjalawe, Hamza Alrababah, Fadi Abu Zwayed, Iznan H. Hasbullah

***Abstract*—-Software-defined networking (SDN) revolutionizes network management by offering centralized control over complex infrastructures, but it also introduces significant security vulnerabilities. particularly Distributed Denial of Service (DDoS) attacks that significantly interrupt network services. The challenge of efficiently detecting DDoS attacks in SDNs is exacerbated by the computational overhead associated with analyzing numerous network features using conventional Machine Learning (ML) techniques. Addressing this gap, our research proposes a novel Intrusion Detection System (IDS) utilizing a 1D Convolutional Neural Network (1DCNN-IDS) model specifically designed to identify DDoS threats within SDN environments. To refine feature selection and enhance detection accuracy, we applied a hybrid objective function incorporating the Akaike Information Criterion (AIC), F-test (ANOVA), and T-test. The effectiveness of our model was validated using three diverse datasets: InSDN, CICIDS2017, and UNSW-NB15, achieving impressive accuracies of over 98%, 96%, and 92% respectively, alongside high precision, recall, and F1 scores. These findings highlight the substantial potential of incorporating ML and Deep Learning (DL) techniques for effective and efficient intrusion detection in SDNs, highlighting our methodology's contribution towards mitigating DDoS attack risks in these networks.**

***Index Terms*—Software-defined networking (SDN), IDS, Deep Learning (DL), 1DCNN, InSDN dataset, FS methods**

I. Introduction

THE conventional networks widely employed today have become increasingly complex, presenting significant challenges in terms of management and administration. Particularly in cases where IT operators need to establish sophisticated network policies, the existing network devices based on IP exhibit vertical integration. In a single network device, the control and data planes are closely integrated. The control plane is responsible for decision-making, while the data plane executes the appropriate actions for network traffic based on directives from the control plane. The process of establishing connections or relationships between different entities is commonly referred to as interlinking. Furthermore, the rapid expansion of networking has the potential to amplify maintenance expenses and create significant obstacles to innovation within conventional network infrastructures. Consequently, developing a novel routing algorithm may require a time frame of 5 to 10 years and incur substantial costs [1].

Moreover, the widespread adoption of devices throughout the network has led to an increase in the number of intermediary devices, such as firewalls, traffic distributors, detection systems, mitigation systems, and other similar components [2]. According to the findings of [1] and [3], a considerable 57% of network enterprises have observed a notable surge in the number of network appliances, now equaling the number of other obligatory network devices, such as routers.

The recently implemented network architecture, commonly called SDN, offers a potential solution to the conventional limitations of IP networks [4]. SDN provides faster-centralized network control as key features [5]. The core principle of SDN is to eliminate vertical integration by separating the control plane from the underlying infrastructure devices. Fig 1 illustrates the primary distinguishing characteristic between SDN and traditional networking.

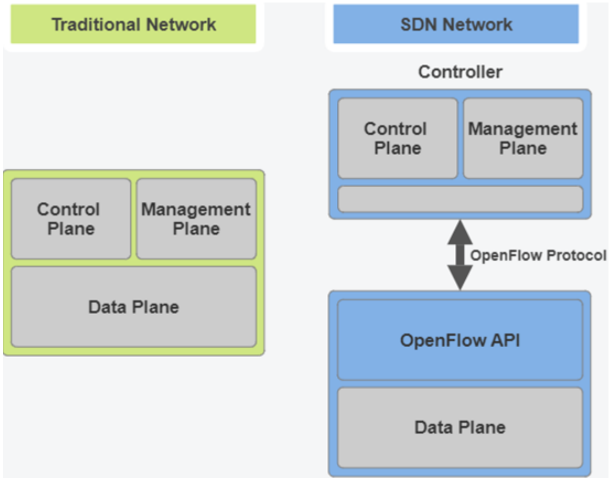

Fig. 1. Traditional Network versus SDN architecture.

On the left, a traditional network device is shown, compartmentalized into a control plane, management plane, and data plane, all enclosed within the device's boundaries, suggesting a tightly coupled architecture. In contrast, the right

Manuscript received August 4, 2023; revised May 13, 2024.

Noor Almi'ani is a Ph.D. student at National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia 11800 USM, Penang, Malaysia (e-mail: nyswe1991@gmail.com).

Mohammed Anbar is a senior lecturer at National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia 11800 USM, Penang, Malaysia (corresponding author to provide phone: +04-6534633; fax: +04-6533888; e-mail: anbar@usm.my).

Shankar Karuppayah is a senior lecturer at National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia 11800 USM, Penang, Malaysia (e-mail: kshankar@usm.my).

Yousef Sanjalawe is an assistant professor at the Department of Cybersecurity, School of Information Technology, American University of Madaba (AUM), Amman, 11821, Jordan (e-mail: jossephhfs@hotmail.com).

Hamza Alrababah is a lecturer at School of Computing, Skyline University College, University City of Sharjah – P.O. Box 1797 - Sharjah, United Arab Emirates (e-mail: hamza.alrababah@skylineuniversity.ac.ae)

Fadi Abu Zwayed is a Ph.D. student at National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia 11800 USM, Penang, Malaysia (e-mail: f.abuzwayed@gmail.com).

Iznan H. Hasbullah is a research officer at National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia 11800 USM, Penang, Malaysia (e-mail: iznan@usm.my).

side depicts the SDN framework with a central controller that governs the control and management planes. Below the controller is the open flow protocol, represented as a bridge between the controller and the data plane, which is further connected to an open flow switch. This layout indicates a decoupled, flexible approach where the control plane can dynamically manage the network via the open flow API. While SDN offers numerous advantages, security poses a significant challenge that can hinder its extensive adoption and implementation across diverse networks [6]. The centralized controller functions as the network's central hub. In the event of successful exploitation of the controller system by an attacker, they can effectively disrupt or manipulate the entire network by their malicious objectives [7]. This vulnerability poses a significant obstacle to the robust implementation of SDN. DDoS attacks represent a major and serious threat to SDN networks.

IDSs are conventional security measures that monitor and identify unauthorized activities occurring within an organizational network [8]. Their primary purpose is to identify and flag illicit actions within the network environment. These systems scrutinize incoming and outgoing network traffic, comparing it against known patterns of suspicious behavior. If a match is found, an alarm is triggered, indicating the detection of a potential attack. IDS plays a crucial role in maintaining network security by promptly identifying and alerting against potential threats [9].

On the flip side, Feature Selection (FS) methods, FS methods types illustrated in Fig 2, constitute a critical preprocessing phase to ensure the success of anomaly detection models. These methods are crucial for selecting and retaining the most relevant features from the dataset. By meticulously curating the feature set, anomaly detection models can be fine-tuned to achieve optimal performance and accuracy. Therefore, feature selection methods are a vital step in the overall process of developing effective anomaly detection models [10].

These methods can be broadly categorized into filter wrapper, and embedded methods. Filter methods, such as correlation-based selection, information gain, and chi-square tests, assess the relevance of features independently of the learning algorithm. They rely on statistics measures to rank and select features based on their inherent properties. On the other hand, wrapper methods, including forward selection, backward elimination, and stepwise selection, evaluate feature subsets by training and testing a specific ML model. These methods search for the optimal feature subset that maximizes the model's performance. Lastly, embedded methods, such as tree algorithms and regularization techniques, integrate feature selection into the model training process itself. They simultaneously learn the model parameters and perform feature selection, considering the specific characteristics of the learning algorithm. By leveraging these diverse feature selection methods, anomaly detection models can effectively identify and prioritize the most relevant features, leading to improved detection accuracy and computational efficiency [11].

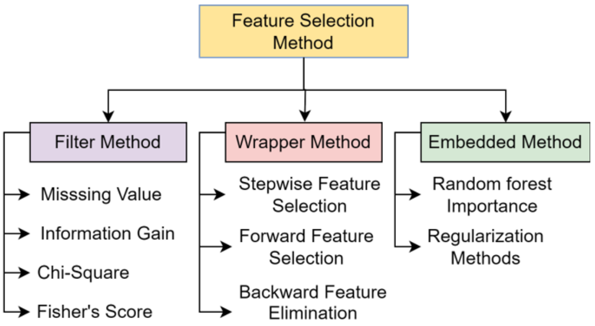

Fig. 2. Taxonomy of FS Methods (redrawing based on [12]).

While numerous ML-based FS techniques have been suggested for detecting DDoS attacks [13],[14],[15], the current preventive measures against DDoS attacks in SDNs are deemed inadequate. Nonetheless, a notable drawback of existing studies is the absence of intrusion datasets specifically designed for SDN networks. Many researchers rely on datasets generated from conventional networks, which may not accurately represent the intricacies of SDN architecture [16]. Consequently, the applicability and effectiveness of these adapted techniques for detecting intrusions in SDNs may be questionable [17], [18]. SDN has introduced unique security threats that differ from the typical risks faced by traditional legacy networks. Consequently, the relevant features characterizing DDoS attacks in traditional networks may not necessarily apply to DDoS attacks in SDN networks. Furthermore, utilizing weak FS algorithms may result in excluding crucial parameters, leading to the loss of significant data information [19],[12].

Given the successful application of DL in various domains, integrating SDN and DL can potentially enhance the performance of IDS and network security. This paper aims to enhance the accuracy of detecting DDoS attacks on SDN networks by integrating a stepwise feature selection method with hybrid objective functions and a 1DCNN. The key contributions of this paper can be summarized as follows:

1) Hybrid Objective Function:
   The introduction of a hybrid objective function that incorporates multiple metrics, including the AIC, ANOVA, and T-test statistics. This comprehensive approach aims to refine the selection of pertinent features for detecting DDoS attacks.
2) Improved Stepwise Regression Feature Selection Method:
   Enhancement of the stepwise selection method based on the hybrid objective functions. This methodology is designed to identify the most pertinent DDoS attack features within the dataset, leveraging the InSDN dataset for validation and refinement.
3) Utilization of 1DCNN for IDSs:
   Implement a 1DCNN in IDSs to efficiently detect DDoS attacks in SDN environments. The 1D CNN processes the features selected through the enhanced stepwise selection method, improving the model's ability to detect and respond to DDoS incidents.

The remaining sections of this paper are structured as follows: Section II briefly explains the background of SDN architecture and some of the new paradigm's security problems. Section III discusses related works and other DDoS attack detection methods in SDNs. Techniques, datasets, and DL models were employed. The methodologies utilized in this paper are outlined in Section IV, whereas the experimental setup is detailed in Section V. The experimental result is in Section VI. The conclusions drawn from this paper are revealed in Section V11.

II. RESEARCH BACKGROUND

This section succinctly introduces the principles underpinning the operation and architecture of SDN and illuminates the security obstacles accompanying this nascent paradigm. This section delves deeper into DDoS attacks in SDN, thoroughly examining their implications and exploring potential defense mechanisms.

A. SDN Architecture

The architecture of the SDN network is divided into three unique operational layers: the application layer, the control layer, and the data layer [20],[21]. The application layer is a platform for executing various applications and services, leveraging northbound programming interfaces. The control layer, on the other hand, is pivotal in its central oversight of the network. It operates separately from the underlying network infrastructure, the data plane, which comprises forwarding network devices like OpenFlow switches [22].

B. DDoS Attack on SDN

Despite the numerous advantages that SDNs offer in various application areas, there are still unresolved security concerns within SDNs. In particular, the centralized control plane presents a prime target for DDoS attacks aiming to overwhelm and disable the controller. The controller's centralized position can enhance network security by enabling the deployment of innovative security tools through northbound APIs. By providing a global view of the network, the controller can also efficiently detect anomalies and attacks. However, the flip side is that the controller also introduces a central point of failure. Flooding the controller with excessive bogus flow requests from compromised SDN switches can disrupt its ability to handle legitimate requests and manage flows. Additionally, the separation of the data and control planes enables indirect DDoS attacks from the data plane to target the control plane. As most SDN controllers lack scalability and resiliency capabilities, even short-lived data plane attacks can significantly degrade the controller's performance. This then indirectly causes denial of service for new flow requests. Moreover, the standardized protocols between the control and data planes like OpenFlow provide a defined interface for attackers to exploit. Vulnerabilities in OpenFlow protocol implementations can be leveraged to trigger flooding or resource depletion DDoS attacks on the controller. The centralized controller, communications between separated data and control planes, and use of standardized SDN protocols introduce avenues for DDoS attacks that can disable the controller and server coordination between the planes. Developing capabilities for DDoS detection and mitigation specifically tailored to the SDN architecture is crucial to realizing the benefits of SDNs across use cases [16],[22].

C. Stepwise Regression Method

Stepwise regression is a method of fitting regression models in which an automatic procedure carries out the choice of predictive variables. It involves adding or removing potential explanatory variables in sequence, and testing for statistical significance after each iteration. This method is particularly useful when dealing with multiple variables and aims to find a suitable model by including variables that have a significant impact on the dependent variable. Stepwise regression can follow two approaches: forward selection, which starts with no variables and adds them one by one, and backward elimination, which starts with all candidate variables and removes the least significant one at each step. The process is guided by specific criteria like the AIC, Bayesian information criterion (BIC), or the adjusted R-squared. The technique simplifies the model-building process but is subject to criticism for its potential to overfit data and its reliance on arbitrary significance levels. Despite this, it remains a popular exploratory tool in statistical analysis and predictive modeling [23].

III. RELATED WORKS

In recent years, numerous protective measures have been proposed to address the issue of DDoS attacks in SDNs. Interestingly, the centralized control feature of SDNs can be leveraged to facilitate the detection of such attacks. This section delves into the most all-encompassing methodology for analyzing DDoS attacks within the framework of SDNs. There have been several research initiatives aimed at resolving this problem, including the implementation of entropy-based approaches (Section III-A), ML-based solutions (Section III-B), and DL-based solutions (Section III-C).

A. Entropy-Based Approaches

In the context of detecting DDoS attacks, entropy has been utilized as a measure of randomness in evaluating traffic patterns [24]. The primary goal of these investigations is to differentiate between legitimate and malevolent traffic by examining the entropy of packet header fields. Through experimental evaluations in simulated environments, the efficacy of entropy-based methods in identifying both low and high-intensity DDoS attacks has been demonstrated. This approach provides a high degree of detection accuracy while minimizing false positives, enhancing its overall performance. Furthermore, the computational overhead imposed by this method has been considered tolerable. For an in-depth analysis of network traffic, future research could explore incorporating additional security layers at the infrastructure stage.

To detect and counteract DDoS attacks in SDN environments, Shannon entropy was employed. [25]. This approach yielded noteworthy outcomes, achieving an accuracy rate exceeding 98.2% and an extraordinarily low false-positive rate of 0.04%. Despite these favorable results, the authors emphasized the imperative need for robust and reliable security solutions that can effectively protect networks against fraudulent traffic. They also outlined potential avenues for future research, including tackling the challenges presented by slow DDoS attacks that emulate legitimate traffic and consume fewer resources, as well as exploring the use of multiple controllers for managing SDN control.

Renyi joint entropy was used to find the connection between different parts of network traffic [26]. The threshold was changed dynamically based on the rate of traffic. This methodology demonstrated enhanced detection accuracy for DDoS attacks across various traffic rates, surpassing existing techniques in terms of true positive rates and false positive rates. Recommendations for future research directions include integrating the dynamic threshold with other information-theory-based algorithms and rule generation using ML techniques. The primary objective of these strategies is to improve the performance of DDoS attack detection by accurately identifying positive instances and reducing the number of false alarms.

To identify and mitigate DDoS attacks in SDNs, researchers incorporated entropy methodology into ensemble learning techniques [27]. of their method, and the Random Forest (RF) algorithm was used to find more problems when they thought things were not going as planned. However, it’s important to note that their use of open-flow switches for detection purposes appears to contravene the core principles of SDNs, which aim to abstract decision-making from the underlying forwarding devices.

Using these different approaches and entropy and statistical measures to find and stop DDoS attacks in SDNs, problems like false positives, the need for a lot of training data, and high computational resource usage can be dealt with. Other limitations include potential missteps in parameter selection, the ability of attackers to adapt and evade detection, and the risk of missing low-rate but damaging attacks. Delays in detection due to computational time and difficulty dealing with dynamic networks can also present challenges [28].

B. ML-Based Approaches

Recently, ML-based anomaly detection techniques have demonstrated remarkable success in identifying DDoS attacks in SDN networks. By harnessing the potential of ML, these methods autonomously learn from training data and reveal underlying patterns. Their superiority over signature-based approaches lies in their ability to detect abnormal network behavior, effectively identifying and mitigating DDoS attacks in SDNs.

A recent study introduced a novel system that employs machine learning techniques to identify DDoS attacks in SDN networks [29]. The adopted approach leverages a hierarchical multi-class (HMC) framework to address the issue of imbalanced datasets effectively and boost the effectiveness of underrepresented classes. The system was evaluated using a dataset of genuine instances of DDoS attacks. The results showed a high level of precision in detecting DDoS attacks within the system, indicating the potential of this approach for real-world applications.

An optimized ensemble model was presented in [30] that uses weighted voting for detecting and mitigating DDoS attacks in Software-Defined Network (SDN) environments. This model combines six fundamental classifiers, including two SVMs, two Random Forests (RFs), and two gradientboosted machines. A novel hybrid metaheuristic optimization algorithm called Binary Harmony Search (BHS) was employed to determine the optimal weights for the ensemble model. The proposed model’s effectiveness was evaluated using the CIC-DDoS2019 dataset, demonstrating a high detection rate of 99.41% while maintaining a low false positive rate of 0.6409%.

Several FS techniques were used on the CICIDS2017 dataset to find important features contributing to the detection of DDoS attacks that were proposed. These included ”SelectPercentile,” ”SelectFromModel,” and Principal Component Analysis (PCA). Most FS methods find a feature size between 12 and 15 to yield optimal results. The researchers evaluated six different machine-learning techniques for classification. Both the RF and the k-nearest neighbors algorithm (KNN) demonstrated superior performance, while Logistic Regression (LR) and Naive Bayes (NB) were comparatively less reliable.

While ML approaches promise effective detection of DDoS attacks in SDNs, they face challenges such as high computational demands and the need for substantial, representative training data. Overfitting can also be an issue, leading to models that don’t generalize well to new data. False positives and negatives may compromise system reliability, and the ’black box’ nature of some ML models complicates troubleshooting and exposes potential weaknesses to adversaries. Finally, ML models might struggle to adapt quickly to rapidly changing network conditions or previously unseen attack types.

C. DL-Based Approach

Currently, DL approaches are significant as they can capture intrinsic patterns within input data automatically without requiring manual intervention. However, there is a scarcity of research that has utilized the DL approach to detect and mitigate DDoS attacks in SDN environments.

In the realm of detecting DDoS attacks in SDNs, [31] conducted a comparative analysis of the efficacy of Artificial Neural Networks (ANN) in contrast to several classical ML algorithms. A dataset about DDoS was generated within a simulated setting, utilizing the Mininet software and Ryu controller. The findings indicate that ANN exhibits promise in detecting attacks, achieving a high level of accuracy at 98.2%.

To enhance the detection of DDoS attacks targeting SDNbased SCADA systems, [32] suggests employing a Recurrent Neural Network (RNN) classifier model that integrates two separate parallel DL approaches: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). The proposed parallel structure was meticulously refined using the training and validation datasets. Experimental studies’ accuracy in detecting DDoS attacks was impressive at 97.62%, and using transfer learning techniques increased it by about 5%. Researchers have shown that the suggested RNN DL classifier model can be used to find DDoS attacks that are aimed at SDN-based SCADA systems.

Despite the inherent advantages of DL approaches over traditional ML approaches, many previous studies have primarily evaluated their models on datasets derived from conventional IP networks rather than on SDN platforms. This oversight hinders the full potential of DL in overcoming the inherent challenges of traditional ML techniques. However, SDNs differ significantly from conventional networks regarding their properties and behavior during operation. SDNs also employ novel protocols, like OpenFlow (OF), that are distinct from legacy networks. IDSs in the SDN can become confused when new vulnerabilities are discovered in the OF protocol, potentially driving attackers to build new attacks.

Much research also relies on out-of-date datasets like KDDCup-'99 and NSL-KDD, compounding the problem. These benchmark datasets weren't just created using data from two decades ago; they also don't include any recent Internet traffic. However, the sophistication and number of new intrusion attack types constantly increase, making them harder to recognize. On the other hand, previous studies that evaluated the effectiveness of DL techniques on SDN networks often utilized custom datasets that simulated the SDN environment. However, these datasets were limited in scope, primarily focusing on a subset of DDoS attacks and neglecting comprehensive coverage of attacks targeting all network layers.

IV. Research METHODOLOGY

In this paper, we propose a two-stage IDS-based 1DCNN model for DDoS attack detection in SDN using an FS method. In the first stage, an objective function that combines AIC, ANOVA, and T-test guides an improved stepwise regression method to find a subset of features that can effectively detect DDoS attacks.

The second stage uses a 1DCNN-IDS model with 64 Conv1D layers that handle 40 features each. These layers capture local dependencies and patterns in the feature sequence so that DDoS attack traits in SDN data can be recognized better. The 1DCNN-IDS model is trained based on the features selected in Stage 1. CCC

A. Data Pre-processing

To construct an accurate IDS, data preprocessing is a pivotal initial step before model training. The original input data may not be in a suitable format for building and training DL models. Several essential steps are undertaken to transform the input dataset into a comprehensible and usable format, including the following steps:

1) Data cleansing:
Data cleansing is a process applied to an existing dataset to eliminate anomalies, ensuring an accurate and unique representation of the dataset's domain. This operation helps reduce the cost and complexity of the model. [33], [34]. The InSDN dataset contains features like source IP, destination IP, and flow ID. To mitigate the risk of overfitting, all socket-specific characteristics have been removed. [16], as these features can vary from network to network. Additionally, non-numeric columns are removed.
2) Encoding the Labeled Data:
Machines understand numbers, not text [35]. It is necessary to convert each text category into numbers for the machine to process using mathematical equations [36]. The labeled string is transformed into a distinctive numerical equivalent using the one-hot encoding method, which converts label classes into unique integer forms[37]. This model specifically focuses on binary classification, distinguishing input data into two categories: malicious and non-malicious [16]. Consequently, a binary value of 0 is assigned to normal strings, while all malicious DDoS attack traffic is represented by a value of 1. The significance of the labeling symbols is depicted in Table I.

TABLE I
ENCODED THE LABELLED DATA

Symbol	Description
0	Normal
1	DDOS attack

3) Scaling:
Data-scaling methodologies include standardization and normalization. Standardization, often called Z-score normalization, adjusts the values of a specific feature in a dataset to achieve a mean of 0 and a standard deviation of 1. This scaling technique assumes that the data follows a Gaussian distribution (also known as a normal distribution or a bell curve). On the other hand, normalization customizes these values to fit within a specified range [38].
The features in the InSDN dataset exhibit varying ranges [39]. To address this variability, the data were rescaled using the standardization method, or Z-score, as per Equation 2 [37]. This process transforms the data scale, resulting in standardized data with a mean value of 0 and a standard deviation of 1.

$$x(i) = \frac{x(i) - \mu(x(i))}{\sigma(x(i))} \tag{1}$$

The feature value is denoted by *i*, while the value after normalization is represented by *x(i)* and has a zero mean (μ) and a standard deviation (σ) of 1.

B. The Hybrid-Objective-Based FS Method

FS plays a vital role in handling high-dimensional datasets in the realm of big data analytics. The primary objective of FS is to identify the most significant features for accurate prediction while discarding irrelevant or extraneous features [40]. The model of this paper focuses on pinpointing the most informative attributes from the input dataset, bridging the gap between the fundamental concept of FS and its practical application. This tailored model is being employed to enhance the efficiency and effectiveness of FS in big data analytics, enable more accurate predictions, and streamline data analysis processes.

1) Hybrid Objective Function

The core purpose of a model based on the FS method is a stepwise regression method combined with an objective function that combines several metrics: AIC, ANOVA, and T-test statistics of a subset of features. The objective score is minimized during the forward selection process and maximized during the backward elimination process. Thus, the FS will favor the selection of features that both improve the log-likelihood of the model (thus lowering the AIC) and have significant relationships with the target variable (thus having large ANOVA and T-Test Statistic)[41],[42]. The formula can define the objective score:

$$ObjectiveScore = \text{Anova} - \text{AIC} + \text{T\_Test} \tag{2}$$

Where:
AIC: a measure of a statistical model's goodness of fit.
Anova: represents the between-group variability relative to the within-group variability, calculated using the ANOVA.

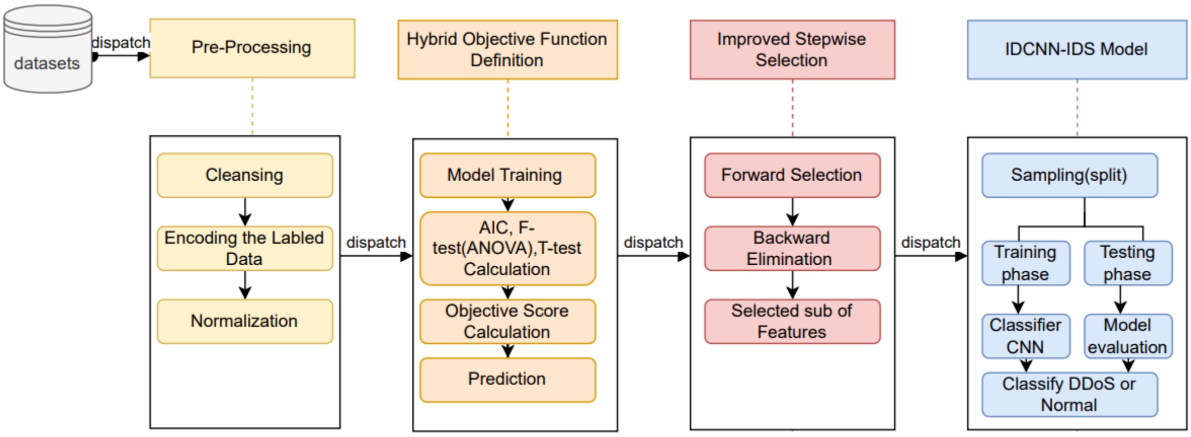

Fig. 3. Architecture of the proposed 1DCNN-IDS model with the FS method

T-test statistic: is the measure of the difference between the means of the selected features and the target variable, computed using the T-test.

2) AIC

An error metric that gives the best predictive accuracy of the model. The AIC penalizes increasing the number of parameters in the model (to discourage overfitting), so it's often used for FS. The goal is to minimize the AIC. The AIC is calculated using the formula [43].

$$AIC = 2k - 2ln(L) \tag{3}$$

Where:
K: is the number of parameters in the model. L: is the likelihood of the model.

3) ANOVA Statistic

The ANOVA measures the ratio of between-group variability to within-group variability in the chosen feature subset[44]. The ANOVA is calculated using the formula[45]:

$$ANOVA = \frac{MSR}{MSE} \tag{4}$$

Where:
MSR=Mean Square Regression.
MSE = Mean Square Error.

$$MSR = \frac{Sum_o f_S quares_R egression}{Degrees_o f_F reedom_R egression} \tag{5}$$

$$MSE = \frac{Sum_o f_S quares_E rror}{Degrees_o f_F reedom_E rror} \tag{6}$$

Where:
Sum_of_Squares_Regression: is the sum of squares explained by the regression model. Degrees_of_Freedom_Regression: is the number of predictor variables in the model. Sum_of_Squares_Error: is the sum of squares not accounted for by the model, also known as the residual sum of squares. Degrees_of_Freedom_Error: is the number of observations minus the number of predictor variables minus 1.

4) T-test Statistic

Ttest can be employed as a part of the process to evaluate the statistical significance of individual features[46]. For each step in the selection process (either forward or backward), a regression model is fitted on the selected features. Then, a t-test is performed for each regression coefficient to test the null hypothesis that the coefficient (and, consequently, the corresponding feature) is not significantly different from zero[47]. In mathematical terms, the t-statistic for a given regression coefficient β can be calculated using the formula[48]: Let's break down the components of this formula:

$$T = \frac{\beta}{\text{SE}(\beta)} \tag{7}$$

Where: β: is the estimated coefficient for the feature. SE(β) is the standard error of the coefficient.

C. Improved stepwise regression method

In traditional stepwise feature selection, the primary metric used is the AIC, which measures the fit of a model to the data while penalizing models with higher complexity to avoid overfitting. The default process typically involves two primary strategies: forward selection, which starts with no variables and adds those that most enhance the AIC, and backward elimination, which begins with all variables and removes those whose exclusion improves the AIC. Relying solely on one metric may degrade the performance of feature selection in terms of selecting appropriate features. Therefore, this approach has been enhanced by adapting the hybrid function depicted in Equation 2.

This hybrid approach modifies the traditional forward and backward selection steps. In the forward selection phase, features are added based on their contribution to minimizing the combined objective score, ensuring the inclusion of features that fit the model well and have strong statistical relationships with the target variable. In backward elimination, features are removed to maximize the objective score, focusing on eliminating features that are statistically insignificant or do not contribute meaningfully to model fit and prediction accuracy.

Algorithm 1 Pseudocode of Improved stepwise regression method

```
Input:
    Data: The dataset of features
    Number of Features to Select (k)
Output:
    Selected Feature Indices
Begin:
    Split Dataset into Predictor Variables (X) and Target Variable (y)
procedure ObjectiveFunction($Model, X, y,$ Feature Indices)
    Fit the Model using $X$ at Feature Indices and $y$
    Predict $y$ using fitted Model and $X$ at Feature Indices
    Calculate Log-Likelihood $(L)$
    Calculate AIC $= 2k - 2\ln(L)$
    Retrieve the best predictive accuracy of the model
    Calculate ANOVA $= \frac{MSR}{MSE}$ between $X$ at Feature Indices and $y$, retrieve F-Statistic
    Calculate $T = \frac{\beta}{SE(\beta)}$ between $X$ at Feature Indices and $y$, retrieve T-Test Statistic
    Calculate Objective Score $=$ ANOVA $-$ AIC $+$ T-Test Statistic
    Return Objective Score
end procedure
Initialize Logistic Regression Model
Apply algorithm 2 (Forward selection)
Apply algorithm 3 (Backward selection)
Return indices of selected Features
```

Algorithm 2 Forward Selection Strategy

```
Initialize SequentialFeatureSelector with Logistic Regression Model, Objective Function, $k$, and direction 'forward'
Apply SequentialFeatureSelector to standardized Predictor Variables and Target Variable
Initialize empty Selected Features set
for $i$ from 1 to $k$ do
    Initialize $best\_fitness$ as negative infinity and $best\_feature$ as None.
    Initialize min Objective Score to $-\infty$
    for each Feature in the Predictor Variables do
        if Feature not in Selected Features then
            Calculate the Objective Score for the Feature using Objective Function
            if Objective Score is less than min Objective Score then
                Update min Objective Score to current Objective Score
                Update $best\_feature$ to current Feature
            end if
        end if
    end for
    Add $best\_feature$ to Selected Features
end for
```

Algorithm 3 Backward Selection Strategy

```
Initialize SequentialFeatureSelector with Logistic Regression Model, Objective Function, $k$, and direction 'backward'
Apply SequentialFeatureSelector to standardized Predictor Variables and Target Variable
while number of Selected Features $>$ desired number of features $(k)$ do
    Initialize $worst\_fitness$ as negative infinity and $worst\_feature$ as None.
    Initialize max Objective Score to $-\infty$
    for each Feature in Selected Features do
        Temporarily remove Feature from Selected Features
        Calculate Objective Score without Feature using Objective Function
        if Objective Score is greater than max Objective Score then
            Update max Objective Score to current Objective Score
            Update $worst\_feature$ to current Feature
        end if
        Add Feature back to Selected Features
    end for
    Remove $worst\_feature$ from Selected Features
end while
```

The impact of this hybrid stepwise selection on the chosen features is substantial. It allows for a more comprehensive evaluation of features, considering their statistical significance and contribution to the model's fit. This leads to improved model accuracy and generalizability, as the model incorporates features that not only fit well according to AIC but are also relevant and distinct, as indicated by ANOVA and T-test scores. Additionally, this method helps reduce overfitting by balancing model complexity with fit and focusing on statistically significant features, which is crucial for predictive modeling. Algorithm 1 shows the pseudocode of the Improved Stepwise Regression method, while Algorithm 2 shows the Forward Selection Strategy, and Algorithm 3 shows the Backward Selection Strategy. After this stage, we obtain a set of significant features that are essential for detecting DDoS attacks on SDN.

D. 1DCNN-based detection model

The 1DCNN-IDS model is trained based on selected features in Stage 1. A 1DCNN operates by applying learned filters, or "kernels," across the sequential data, capturing localized patterns [49]. This feature allows the model to identify and learn from local dependencies in the data, thereby being robust to noise and distortions. Additionally, due to the shared-weight architecture of CNNs, 1DCNNs are relatively parameter-efficient compared to fully connected networks, enabling faster training [50]. Furthermore, the hierarchical structure of 1DCNNs allows them to learn complex patterns at varying scales, contributing to their wide applicability and power in various tasks involving sequential data[51].

Cov1D layers, which are 1D adaptations of CNN layers, are utilized. Each Cov1D layer comprises 64 convolution filters and 40 features, in conjunction with a kernel size 3. The 'RELU' activation function is applied to these layers, engineered to transmit the input directly if it is positive, or else it generates zero. Subsequently, a dropout layer is integrated to enhance the regularization of the output. A max-pooling layer is then introduced, inherently simplifying or pooling the feature maps and creating a condensed depiction of the detected features in the input. 1D max pooling is used for this purpose. The processed output is next guided through a flattening layer. This layer converts the output matrix into a more amenable vector format, promoting efficient classification. In the final phase of the 1DCNN model, a fully connected layer is included. This layer splits into two Dense layers. The initial Dense layer is designed with 100 neuron detectors and a 'Relu' activation function. Conversely, the final Dense layer is designed with 2 feature detectors and a 'Softmax' activation function.

V. EXPERIMENTAL SETUP

The experimental design serves as a blueprint to assess the efficiency and reliability of the proposed SDN data classification method. This section explains the dataset used (utilizing the InSDN dataset rich in diverse SDN activities, the CICIDS2017 dataset, and the UNSW-NB15 dataset), the model setup, and the choice of evaluation metrics (accuracy, precision, recall, and F1 score).

A. Benchmark Dataset

This research makes use of the InSDN dataset, CICIDS2017, and UNSW-NB15 network intrusion datasets for evaluation. The InSDN dataset encapsulates diverse SDN activities, including normal traffic like HTTPS, HTTP, DNS, Email, FTP, and SSH, and attack categories such as DoS, DDoS, Probe, Botnet, Exploitation, Password-Guessing, and Web attacks. It provides over 80 statistical features extracted from PCAP and CSV formats [16]. CICIDS2017 contains network traffic generated from an intrusion detection testbed, including normal activities and attacks like DoS, DDOS, brute force, infiltration, botnet, etc. The raw flows were pre-processed to extract 80 features[52]. UNSW-NB15, created by cybersecurity researchers, comprises 49 features from raw packets representing contemporary normal and attack traffic[53].

B. Architecture of 1DCNN-IDS model

This section explains the architecture of the proposed 1DCNN-IDS model as shown in Fig 4 alongside the hyper-parameters used to train the 1DCNN-IDS model as tabulated in II.

TABLE II
HYPERPARAMETERS USED IN 1DCNN-IDS MODEL TRAINING

Parameter	Value
Epoch	100
Activation function	ReLU
Batch Size	128
Learning Rate	0.001
Loss Function	Sparse categorical cross entropy
Optimizer	Adam

As illustrated in Fig 4, the proposed 1DCNN-IDS model consists of multiple convolutional layers and pooling layers designed to extract crucial features from the input data. Experimental analysis was conducted to determine the optimal number of computational layers and kernel size by varying the number of convolutional layers from 1 to 3 and the kernel size from 2 to 5, as depicted in Fig 5 and Fig.6, respectively.

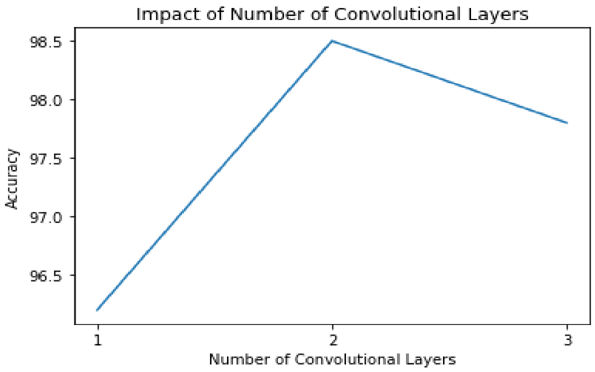

Fig. 5. Impact of Number of Convolutional Layers

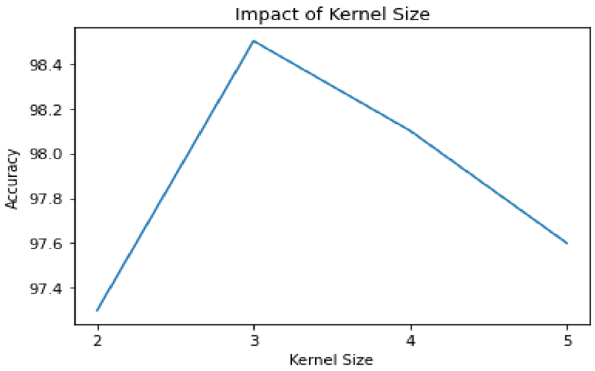

Fig. 6. Impact of Kernel Size

The results in Fig 5 and Fig.6 show the model achieves optimal performance with 2 convolutional layers and a kernel size of 3.

The hyperparameters employed for training the model, detailed in Table II, encompass factors such as the number of filters, filter size, activation function, and learning rate, among others. These hyperparameters play a crucial role in optimizing the model's performance and convergence during the training process. The number of epochs is set to 100, allowing the model to iterate over the training data multiple times and improve its learning. The ReLU activation function is chosen for its simplicity and effectiveness in introducing non-linearity to the model. The batch size of 128 determines the number of samples processed in each iteration, balancing computational efficiency and convergence speed. The learning rate of 0.001 controls the step size at which the model's weights are updated during optimization, ensuring a stable and gradual learning process. The sparse categorical cross-entropy loss function is selected as it is well-suited for multi-class classification problems, measuring the dissimilarity between the predicted and actual class distributions. Finally, the Adam optimizer is employed for its adaptive learning rate and efficient convergence properties. The rationale for selecting these hyperparameters aligns with commonly utilized practices in existing research, as demonstrated in [54],[55].

C. Evaluation Metrics

The model's efficiency is assessed by employing widely accepted performance metrics such as accuracy, precision, recall, and F-score metrics [53]. These metrics are calculated using the following equations.

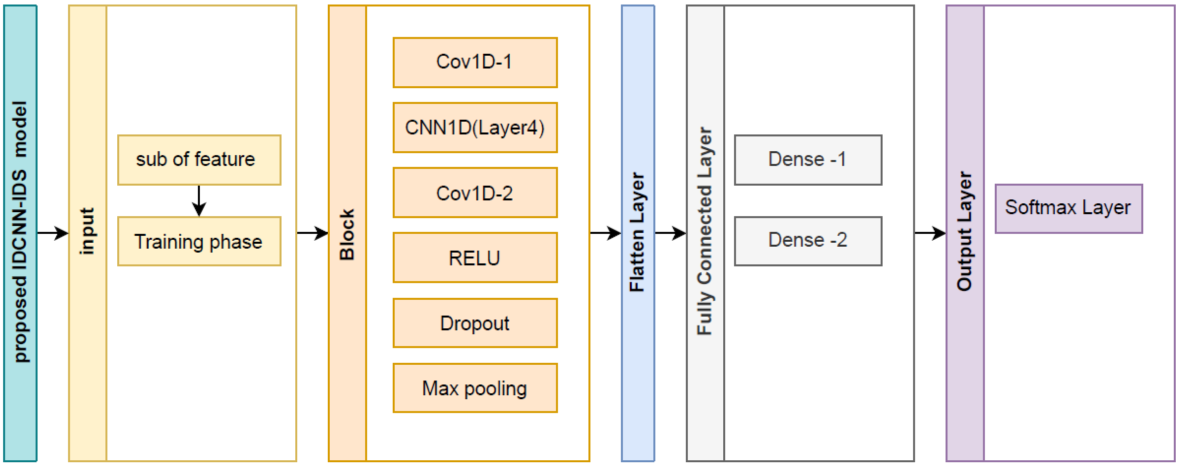

Fig. 4. Block diagram of the proposed 1DCNN-IDS model

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

Accuracy measures the overall correctness of the model's predictions, considering both true positives (TP) and true negatives (TN) about the total number of instances. Precision focuses on the model's ability to correctly identify anomalous instances among all the instances predicted as anomalous, calculated as the ratio of true positives to the sum of true positives and false positives (FP). Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify all the actual anomalous instances, computed as the ratio of true positives to the sum of true positives and false negatives (FN). The F-score is a harmonic mean of precision and recall, providing a balanced evaluation of the model's performance, particularly useful when dealing with imbalanced datasets. These metrics, along with the analysis of TP, TN, FP, and FN, form a robust framework for assessing the performance of anomaly detection models, allowing researchers and practitioners to make informed decisions about their effectiveness and suitability for specific tasks.

VI. EXPERIMENTAL RESULT

The findings of this paper substantiate the methodology presented for developing an efficient IDS in SDN. The paper quantifies its effectiveness through evaluation metrics, including accuracy, precision, recall, and F1-score.

A. *Result of Improved Stepwise Regression Feature Selection method*

The improved stepwise regression significantly reduced the dimensionality of the initial three datasets through an efficient FS process. The process began with an exhaustive list of features, systematically pruned to include only those with an objective score that minimized AIC, maximized ANOVA, and the T-test statistic.

This rigorous FS contributes to a more efficient and manageable data pipeline, retaining the most significant data for further steps in the model training process. The algorithm adopted an FS process to add features one at a time based on the best objective score and a backward elimination process to remove the least significant features, ensuring an optimal feature subset. The data reduction process for the InSDN, CICIDS2017, and UNSW-NB15 datasets is depicted in Table III.

As depicted in Table III, the high dimensionality of the InSDN, CICIDS2017, and UNSW-NB15 datasets is efficiently reduced through FS.

TABLE III
FS IN INSDN, CICIDS2017,AND UNSW-NB15 DATASETS

Dataset	Initial Number of Features	Number of Features After Selection
InSDN	48[23]	10
CICIDS2017	80[54]	12
UNSW-NB15	42[55]	8

For InSDN, the features were reduced from 48 to 10, for CICIDS2017 from 80 to 12, and for UNSW-NB15, the reduction was from 42 to 8 relevant features. This refinement decreases model complexity, improves computational efficiency, and enhances the training of the 1D-CNN-IDS model by retaining only the most significant attributes. The reduced dimensionality also results in faster and more effective data processing, further boosting performance. Overall, FS streamlines all three datasets by eliminating redundant and irrelevant information, leading to more efficient models with greater accuracy in network intrusion detection.The features selected by the improved stepwise regression method for each dataset are outlined in Table IV for InSDN, Table V for CICIDS2017, and Table VI for UNSW-NB15.

TABLE IV
THE 10 FEATURES SELECTED IN INSDN DATASET USING IMPROVED STEPWISE REGRESSION FEATURE SELECTION METHOD

Feature Index	Name of Feature	Rationale for Feature
1	Protocol	Identifies targeted protocols in DDoS attacks.
2	Flow Duration	Unusual durations can signal an attack.
5	Total Forward Packets	High counts indicate one-sided conversations in DDoS.
15	Flow Packets	Reflects abnormal traffic patterns in DDoS attacks.
18	Flow IAT Max	Deviations suggest irregular traffic in DDoS attacks.
30	Backward IAT Total	Indicates response to attack patterns.
31	Forward Header Length	Manipulated headers can indicate DDoS attacks.
32	Backward Header Length	Changes can indicate malicious DDoS activities.
33	Forward Packets	High count can indicate an outgoing DDoS attack.
34	Backward Packets	High counts can suggest a response to incoming attacks.

TABLE V
THE 12 FEATURES SELECTED IN CICIDS2017 DATASET USING IMPROVED STEPWISE REGRESSION FEATURE SELECTION METHOD

Feature Index	Name of Feature	Rationale for Feature
3	Source Port	Unusual traffic can indicate an attack.
8	Flow Duration	Identifies abnormal network flows.
9	Total Fwd Packets	High count suggests overwhelming traffic.
23	Flow IAT Mean	Unusual averages signal irregular traffic.
24	Flow IAT Std	High values indicate inconsistent flows.
32	Bwd IAT Total	Anomalies suggest a response to an attack.
34	Bwd IAT Std	High variability indicates attack responses.
47	Packet Length Mean	Changes suggest malicious packet crafting.
54	ACK Flag Count	Unusual patterns indicative of SYN flood attacks.
59	Average Packet Size	Deviations indicate attack traffic.
60	Avg Fwd Segment Size	Changes suggest an ongoing attack.
61	Avg Bwd Segment Size	Abnormal sizes indicate a response to attacks.

The improved stepwise feature selection method substantially reduced feature dimensionality by systematically eliminating redundant variables based on multiple mathematical criteria. After using minimum AIC, maximum ANOVA, and maximum T-test statistics together, the best feature subsets that could predict DDoS attacks were selected from the original feature sets.

For the InSDN dataset, the initial 48 features were reduced to the 10 most relevant features that exhibited strong statistical significance across the combined objective metrics. Table IV shows that the chosen characteristics include protocols, long flow times, lots of traffic, strange packet and IAT patterns, and packet headers that have been changed.

Similarly, for CICIDS2017, the hybrid statistical evaluations retained 12 features demonstrating maximum predictive capability from the original 80 attributes. The best features include source ports, duration, forward packet counts, IAT statistics, ACK flag metrics, segment sizes, and other key signs of DDoS attacks, as shown in Table V.

TABLE VI
THE 8 FEATURES IN UNSW-NB15DATASET USING IMPROVED STEPWISE REGRESSION FEATURE SELECTION METHOD

Feature Index	Name of Feature	Rationale for Feature
1	srcip	Source IPs analyzed for DDoS patterns.
3	dstip	Identifies targeted devices or services.
8	sbytes	High values indicate an outgoing attack.
9	dbytes	High values represent a response to incoming attacks.
10	sttl	Low TTL values are used in certain attack strategies.
11	dttl	Deviations from the norm indicate a network under attack.
15	Sload	A high load indicates a source attempting to flood

In the same way, the combined minimum AIC, maximum ANOVA, and T-test benchmark were used to pick eight statistically significant features from the original 42 variables for UNSW-NB15. As Table VI exhibits, core indicators like IP addresses, payload sizes, TTL values, and high originating loads had outstanding statistical scores, underlying their relevance in pinpointing DDoS attacks.

B. Result of 1DCNN-IDS model using several Datasets

The 1DCNN-IDS model underwent evaluation using the InSDN, CICIDS2017, and UNSW-NB15 datasets. The use of multiple datasets enhances the diversity of the evaluation. Different datasets may capture distinct aspects of network traffic and attack scenarios, ensuring a more comprehensive assessment of the model's capabilities. Table VII displays the performance metrics for the InSDN, CICIDS2017, and UNSW-NB15 datasets.

TABLE VII
PERFORMANCE METRICS OF THE 1DCNN-IDS MODEL ON THE INSDN, CICIDS2017, AND UNSW-NB15 DATASETS USING IMPROVED STEPWISE REGRESSION FEATURE SELECTION

Metric	InSDN Value (%)	CICIDS2017 Value (%)	UNSW-NB15 Value (%)
Accuracy	98.2	96.4	92.5
Precision	98.7	97.2	94.2
Recall	97.8	95.3	91.7
F1-score	98.6	96.1	93.3

As depicted in Table VII, the proposed 1DCNN-IDS demonstrated exceptional performance on the InSDN dataset. Notably, it achieved an accuracy of 98.2%, precision values of 98.7%, a recall rate of 97.8%, and an F1 score of 98.6%. Collectively, these metrics underscore the model's accuracy, precision, recall, and overall balanced performance in predicting activities within the InSDN dataset.

The evaluation of the model's performance extended to the CICIDS2017 and UNSW-NB15 intrusion detection datasets. On the CICIDS2017 dataset, the model achieved an accuracy of 96.4% and demonstrated high precision, recall, and F1 scores. For the UNSW-NB15 dataset, the model achieved an accuracy surpassing 92.5%. The consistent and robust results across these diverse datasets showcase the model's versatility in effectively identifying network intrusions.

1) confusion matrix

The confusion matrix shown in Fig 7, Fig8, and Fig 9 is used to calculate the performance metrics for the INSDN, CICIDS2017, and UNSW-NB15 datasets, respectively. The confusion matrices provide a detailed breakdown of the performance of the 1DCNN-IDS model on three different datasets: InSDN, CICIDS2017, and UNSW-NB15. The InSDN dataset shows a relatively small number of false positives (92) and false negatives (54), indicating a good overall performance of the model. However, the CICIDS2017 and UNSW-NB15 datasets have a larger number of misclassifications, with CICIDS2017 having 455 false positives and 265 false negatives, and UNSW-NB15 having 893 false positives and 607 false negatives.

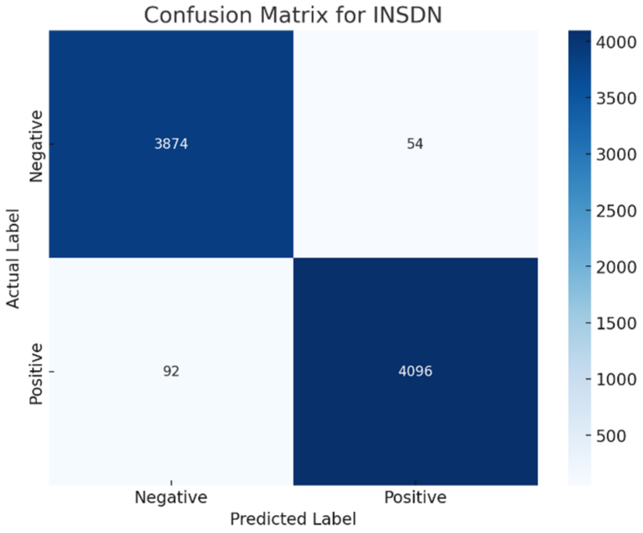

Fig. 7. Confusion matrix for INSDN

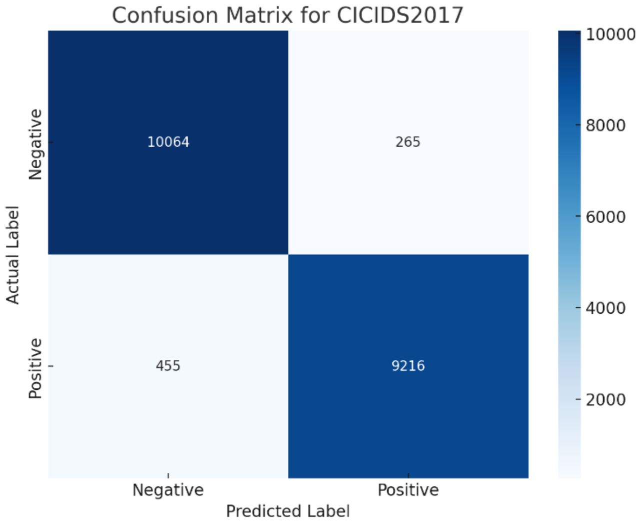

Fig. 8. Confusion matrix for CICIDS2017

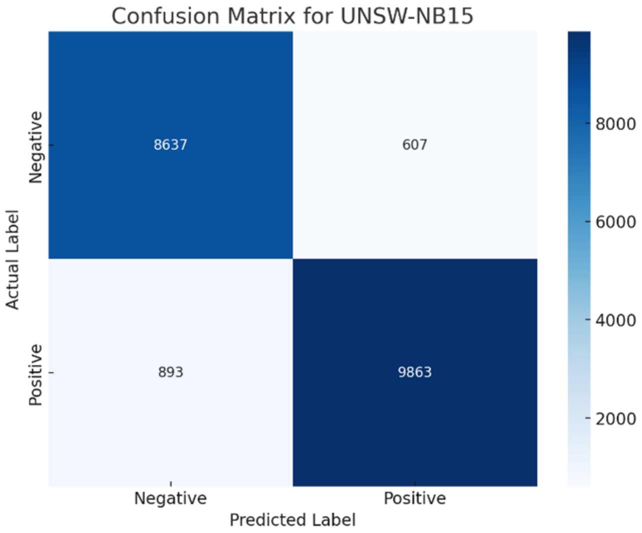

Fig. 9. Confusion matrix for UNSW-NB15

2) The training and validation loss curves

It is worth mentioning that we observed the training models of the 1DCNN-IDS resulting from the three datasets to ensure the absence of overfitting or underfitting problems. Figurine 10 displays the training and validation loss curves of the proposed 1DCNN-IDS model using the InSDN dataset. Figurine 11 shows the training and validation loss curves for the same model used on the CICIDS2017 dataset, and Figurine 12 shows the results for the UNSW-NB15 dataset. These graphs serve as powerful tools for visualizing the model's performance over time across different datasets.

The training and validation loss curves provide valuable insights into the model's learning progress and generalization ability. In an ideal scenario, both the training and validation loss curves should decrease steadily and converge to a stable value, indicating that the model is learning effectively from the data and generalizing well to unseen samples. If the training loss continues to decrease while the validation loss starts to increase, it suggests that the model is overfitting, meaning it is memorizing the training data instead of learning meaningful patterns. Conversely, if both the training and validation losses remain high and do not decrease significantly, it implies that the model is underfitting, struggling to capture the underlying patterns in the data [56].

The curves in Fig. 10, Fig.11, and Fig. 12 depict the model's learning progress, with the loss values consistently decreasing as the training advances. The comparison between training and validation losses provides crucial insights into the model's learning status on each dataset. Ideally, both curves should exhibit a similar downward trend for each dataset, indicating effective learning without overfitting or underfitting. Any significant divergence between these curves might suggest potential overfitting or underfitting issues, warranting further investigation into the model's generalization capabilities.

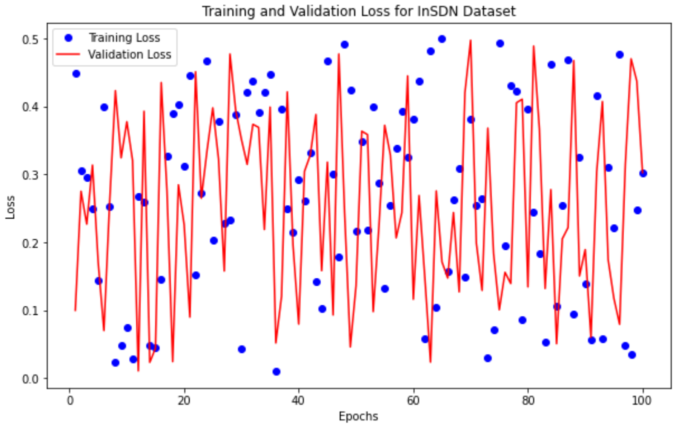

Fig. 10. Training and Validation Loss Curve for InSDN using 1DCNN-IDS model

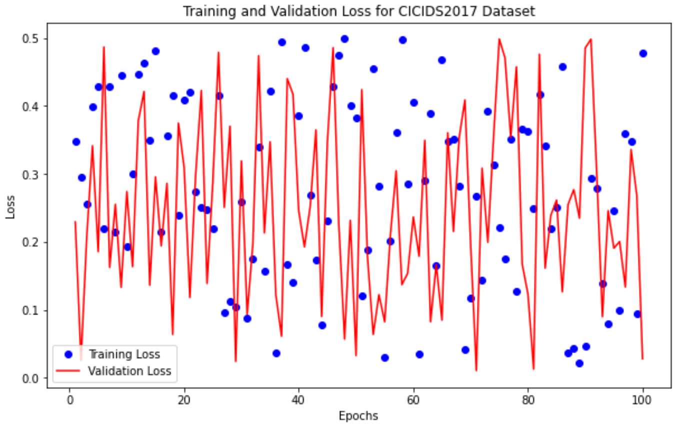

Fig. 11. Training and Validation Loss Curve for CICIDS2017 using 1DCNN-IDS model

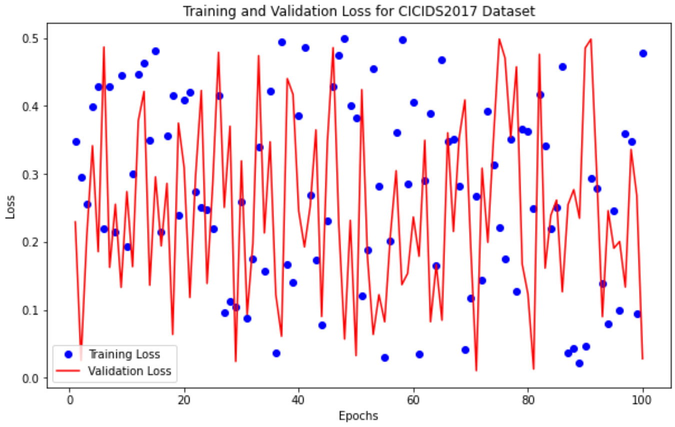

Fig. 12. Training and Validation Loss Curve for UNSW-NB15 using 1DCNN-IDS model

The training and validation loss curves for the InSDN, CICIDS2017, and UNSW-NB15 datasets, shown in Figures 10, 11, and 12 respectively, demonstrate that the 1DCNN-IDS model is learning effectively and generalizing well across different datasets. In all three cases, the curves are closely aligned and exhibit a consistent downward trend, converging towards stable values. This suggests that the model is capturing meaningful patterns from the data and improving its performance over time, without significant overfitting or underfitting issues. The absence of a widening gap between the training and validation losses indicates that the model is not memorizing the training data but rather learning to generalize to unseen samples. Comparing the loss curves across the three datasets reveals similar patterns of learning behavior, reinforcing the model's robustness and adaptability to various network traffic scenarios.

3) ROC and PR Curves

Receiver Operating Characteristic (ROC) curves and Precision-Recall (PR) curves provide a more nuanced visualization of model performance by plotting the trade-off between true positive rate vs false positive rate and precision vs recall respectively at different classification thresholds.

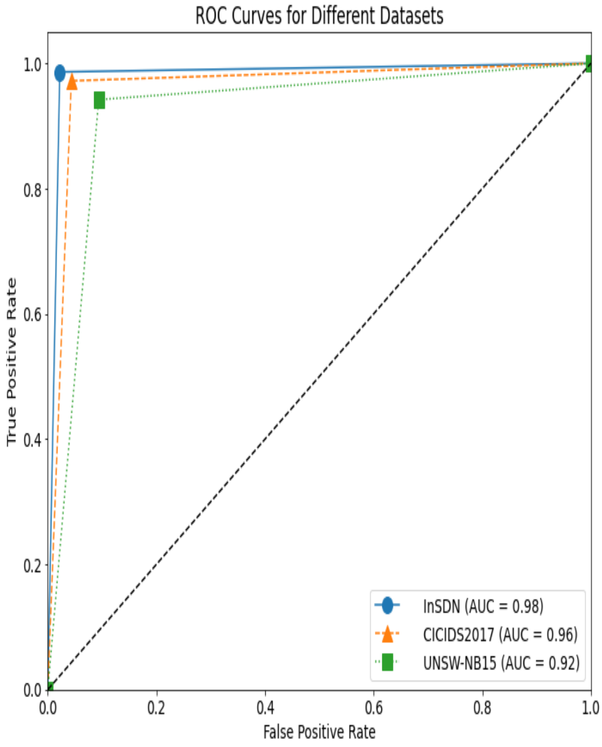

Fig. 13. ROC curve for 1DCNN-IDS on three datasets: InSDN,CICIDS2017 and UNSW-NB15

Fig.13 and Fig.14 presents the ROC and PR curves for the evaluation of the 1DCNN-IDS model across three distinct datasets: InSDN, CICIDS2017, and UNSW-NB15.From the ROC curve, we can observe that the 1DCNN-IDS model performs exceptionally well on the InSDN dataset, with an AUC of 0.98, indicating excellent discrimination between normal and attack instances. The model's performance on the CICIDS2017 dataset is also quite good, with an AUC of 0.95. However, the model's performance on the UNSW-NB15 dataset is relatively lower, with an AUC of 0.91, suggesting a greater difficulty in distinguishing between normal and attack instances in this dataset. The PR curves, on the other hand, provide insights into the trade-off between precision (the fraction of true positives among all positive predictions) and recall (the fraction of true positives among all actual positives). These curves are particularly useful when dealing with imbalanced datasets, where the positive (attack) instances are significantly outnumbered by the negative (normal) instances.

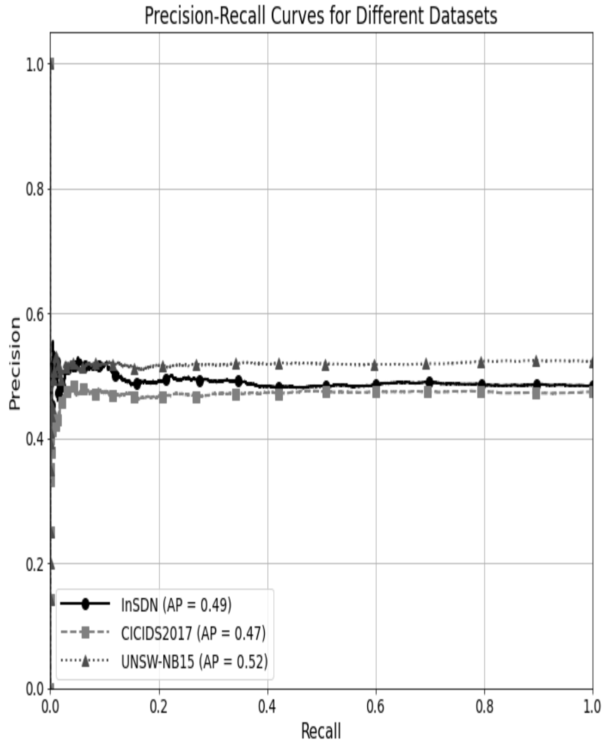

Fig. 14. RP curve for 1DCNN-IDS on on three datasets: InSDN,CICIDS2017 and UNSW-NB15

From the PR curves, we can see that the 1DCNN-IDS model achieves high precision and recall values on the InSDN dataset, with an average precision (AP) of 0.98. The model's performance on the CICIDS2017 dataset is also commendable, with an AP of 0.95. However, the model's performance on the UNSW-NB15 dataset is relatively lower, with an AP of 0.91, consistent with the observations from the ROC curve analysis.Overall, these results demonstrate the effectiveness of the 1DCNN-IDS model in detecting DDoS attack, particularly on the InSDN and CICIDS2017 datasets.

Finally, the performance metrics of the 1DCNN-IDS model on INSDN, CICIDS2017, and UNSW-NB15 datasets using the standard stepwise regression feature selection method and without using the feature selection method, are listed in Table VIII and IX, respectively.

TABLE VIII
PERFORMANCE METRICS OF THE 1DCNN-IDS MODEL ON THE INSDN, CICIDS2017, AND UNSW-NB15 DATASETS WITH STANDARD STEPWISE REGRESSION FEATURE SELECTION METHOD

Metric	InSDN (%)	CICIDS2017 (%)	UNSW-NB15 (%)
Accuracy	90.0	85.0	80.0
Precision	90.5	86.0	81.0
Recall	89.0	83.0	78.0
F1-score	89.5	84.5	79.5

TABLE IX
PERFORMANCE METRICS OF THE 1DCNN-IDS MODEL ON THE INSDN, CICIDS2017, AND UNSW-NB15 DATASETS WITHOUT FEATURE SELECTION METHOD

Metric	InSDN (%)	CICIDS2017 (%)	UNSW-NB15 (%)
Accuracy	85.0	80.0	75.0
Precision	85.5	81.0	76.0
Recall	84.0	78.0	73.0
F1-score	84.5	79.5	74.5

As depicted in Table VIII, when the model was applied with a standard stepwise regression feature selection method, there was a noticeable decrease in performance metrics across all datasets. Specifically, the accuracy dropped to 90.0% for InSDN, 85.0% for CICIDS2017, and 80.0% for UNSW-NB15. In the same way, precision, recall, and F1 scores went down. This shows that standard stepwise regression helps choose features, but its improved version is better at selecting the significant features contributing to detecting DDoS attacks.

Meanwhile, the most significant performance drop was observed when the model was applied without any feature selection, as shown in Table IX. The accuracy drops to 85.0% for InSDN, 80.0% for CICIDS2017, and 75.0% for UNSW-NB15. Precision, recall, and F1-scores followed a similar downward trend. This decline in performance metrics highlights the critical role of feature selection in the model's ability to accurately detect network intrusions. Without feature selection, the model is likely overwhelmed by irrelevant or redundant data, impeding its learning and predictive capabilities.

The comparison of the 1DCNN-IDS approach using various feature selection techniques indicates the significance of the feature selection method in IDS. Specifically, the improved stepwise regression feature selection method substantially improves the model's performance across different datasets, as illustrated in Table VII. This demonstrates its effectiveness in fine-tuning the model for optimal intrusion detection.

C. Comparative Analysis

The proposed 1DCNN-IDS has undergone benchmarking against several prominent IDSs leveraging DL algorithms. This comparative evaluation utilizes the metrics outlined in Section V-C. This assessment aims to determine the accuracy, precision, recall, and F1-score of the proposed 1DCNN-IDS in identifying intrusions within a complex network environment, especially when compared to similar IDSs.

The comparison involves IDSs based on well-known DL models, such as CNN-IDS[57], CNN with GRU-IDS[58], LSTM-IDS[59], and CNN-IDS[60]. These models were chosen as benchmark models due to their similar performance characteristics. Fig 15 depicts a comparison between the proposed 1DCNN-IDS and these benchmark models, including CNN-IDS, CNN-GRU-IDS, LSTM-IDS, and a Deep Forest-based NIDS. The comparison is based on four fundamental metrics listed in Section V-C

As depicted in Fig. 15, the proposed 1DCNN-IDS approach distinguishes itself with exceptional performance, achieving the highest accuracy at 98.2%. This signifies a remarkable ability to correctly classify instances compared to benchmark models. Notably, the precision of the proposed 1DCNN-IDS is impressive at 98.7%, indicating a low false positive rate and effective identification of positive instances while minimizing false alarms. Additionally, the proposed 1DCNN-IDS demonstrates strong recall at 97.8%, showcasing its effectiveness in capturing actual positive instances and maintaining a low false negative rate. The F1-score, a metric that balances precision and recall, peaks with the proposed 1DCNN-IDS at 98.6%. This underscores its balanced performance by accurately identifying positive instances while

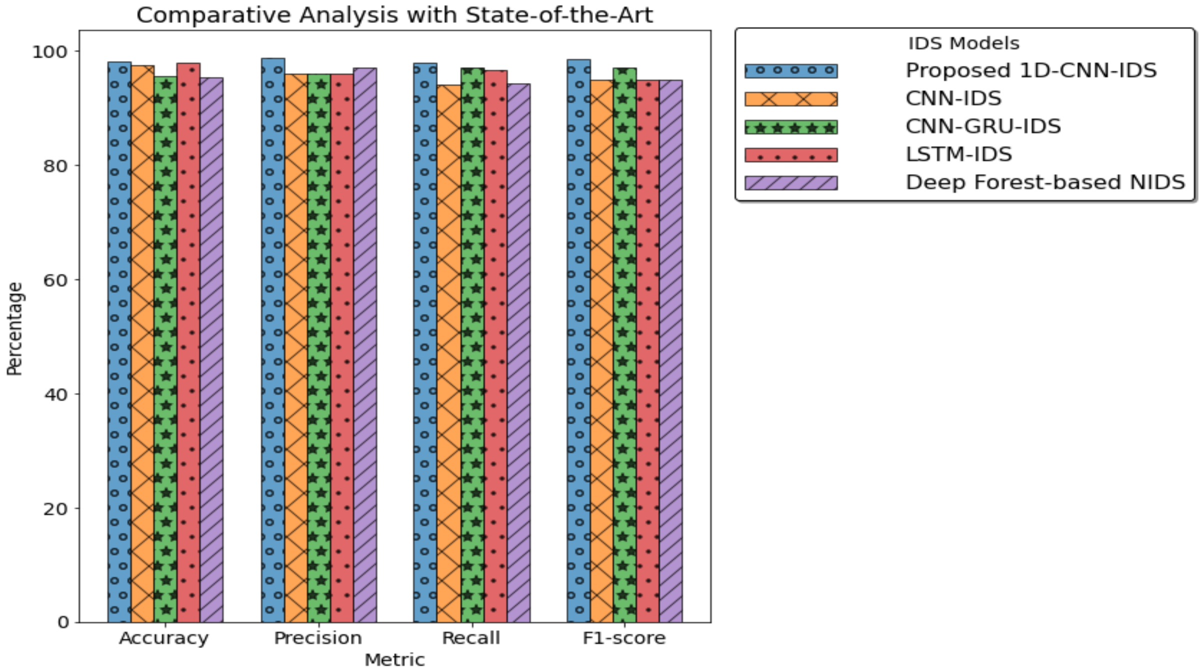

Fig. 15. Performance Comparison: Proposed 1DCNN-IDS vs. State-of-the-Art IDSs

minimizing false alarms. In comparison, other benchmark models, such as CNN-IDS, CNN-GRU-IDS, LSTM-IDS, and Deep Forest-based NIDS, exhibit lower performance across these metrics. Consequently, the proposed 1DCNN-IDS emerges as a promising IDS, outperforming its counterparts in various aspects of intrusion detection.

D. Discussion

The significant performance superiority of the 1DCNN-IDS approach, as illustrated in Fig. 15, can be attributed to two key contributions. Firstly, a hybrid objective function integrating AIC, ANOVA, and T-test statistics facilitates a more focused selection of the most relevant input features from the DDoS datasets. This hybrid function, when combined with stepwise elimination, narrows down the feature space to a vital subset, demonstrating high discrimination between attack and normal classes. Feeding the selected subset features to a customized 1DCNN-IDS model enhances the detection capabilities. The collaborative use of hybrid feature selection and an optimized 1DCNN-IDS model increases accuracy to 98.2%, precision to 98.7%, recall to 97.8%, and F1 to 98.6%, surpassing existing methods across all key metrics.

Quantitatively, this highlights the 1DCNN-IDS's reliable identification of actual DDoS samples while minimizing false alarms, supporting its effectiveness for real-world deployment. Delving deeper into the performance gains, the 1DCNN-IDS correctly classifies a higher proportion of total traffic, effectively balancing precision and recall, outperforming alternatives by over 3% recall and 0.8% accuracy.

While the 1DCNN-IDS model shows strong DDoS detection capability, open challenges remain for real-world viability of IDS[57],[58],[59],[60]. The hybrid deep learning model from [58] achieves high accuracy but relies on complex, time-consuming manual selection of architectures. The LSTM and CNN models from [59] also do not match the hyperparameters and layers of the neural networks to the datasets better, which makes detection less accurate. Also, the multi-stage Deep Forest classifier from [60] needs to be tested on real SDN testbeds to ensure it works in software-defined environments. A lot of work needs to be done to make sure that existing models are still up to date with the latest information about networks and to test how reliable they are against inputs that are purposely misclassified [57] and citehenry2023composition. There are also problems with figuring out how data class imbalance, computational complexity, and interpretability affect IoT environments with limited resources [59],[60]. The proposed 1DCNN-IDS model solves these problems by using a more advanced automatic statistical feature selection method that combines stepwise regression with a hybrid objective function. This model streamlines architecture and hyperparameter tuning, consequently enhancing accuracy. The hybrid objective function formulation ensures that the inputs are optimized for the specific characteristics of the DDoS data. This makes sure that the feature selection process is highly customized and effective. The 1DCNN-IDS model, with its automated input filtering and precision-tuned architecture, marks a significant step forward in advancing DDoS detection, bringing it closer to providing robust real-world protection.

In our approach to DDoS detection in SDNs, we introduce a new approach that leverages advanced machine learning techniques, distinguishing itself significantly from existing methods through its adaptability and efficiency. Unlike traditional detection systems that often rely on static rule sets or simple anomaly detection algorithms, our model employs a dynamic, data-driven approach. This enables it not only to identify a broader range of DDoS attack types, including those that employ sophisticated masking techniques but also to adapt over time as new patterns of attacks emerge. Our method shows particular strengths in environments with evolving attack vectors, highlighting its robustness under various network conditions.

Furthermore, we have conducted a comprehensive comparison with leading existing methods, demonstrating that our approach not only achieves higher accuracy rates but also maintains superior computational efficiency. This efficiency is crucial for SDNs, where the rapid processing of large volumes of network traffic data is essential. Our model optimizes resource usage, ensuring minimal impact on network performance while actively monitoring for and mitigating DDoS threats. This balance of high detection accuracy with low computational overhead offers new insights into creating scalable and effective security solutions for modern networks, addressing one of the critical challenges in cybersecurity today.

VII. Conclusion

The effectiveness of the proposed 1DCNN-IDS model in classifying network intrusion data was confirmed through experimental results, specifically within an SDN environment. The model displayed outstanding performance on the InSDN dataset, underscoring its adaptability and robustness to various network activities. Remarkable precision in identifying malicious activities was demonstrated, an attribute that is vital to minimizing the potential negative impact of false positives in IDSs within SDNs. Additionally, high recall rates underline the model's ability to correctly pinpoint a significant percentage of actual positive cases, affirming its utility in real-world SDN scenarios. It's important to note that various factors can affect how well DL models perform, such as the suggested 1D CNN-IDS. These encompass the selection of hyperparameters, the intricacy of the model's architecture, and the unique attributes of the dataset. As a result, ongoing research and meticulous fine-tuning of the model may be required to ensure optimal performance across diverse contexts. In summary, this paper underscores the promising potential of DL methodologies, specifically 1DCNN, in tackling complex classification tasks within the SDN environment. These findings pave the path for future research and advancements in the classification of network data, specifically within SDN environments, and contribute to the further development of intrusion detection systems.

References

[1] B. Zoradia and G. Indumati, "Comparison of software defined networking with traditional networking using ns2 simulator." *International Journal on Information Technologies & Security*, vol. 15, no. 3, 2023.

[2] A. N. Alhaj and N. Dutta, "Analysis of security attacks in sdn network: A comprehensive survey," *Contemporary Issues in Communication, Cloud and Big Data Analytics: Proceedings of CCB 2020*, pp. 27–37, 2022.

[3] O. Fares, A. Dandoush, and N. Aitsaadi, "Sdn-based platform enabling intelligent routing within transit autonomous system networks," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 909–912.

[4] G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for sdn using machine learning," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 867–880, 2023.

[5] T. E. Ali, Y.-W. Chong, and S. Manickam, "Comparison of ml/dl approaches for detecting ddos attacks in sdn," *Applied Sciences*, vol. 13, no. 5, p. 3033, 2023.

[6] T. ALASALI and O. DAKKAK, "Exploring the landscape of sdn-based ddos defense: A holistic examination of detection and mitigation approaches, research gaps and promising avenues for future exploration," *International Journal of Advanced Natural Sciences and Engineering Researches*, vol. 7, no. 4, pp. 327–349, 2023.

[7] A. Liatifis, P. Sarigiannidis, V. Argyriou, and T. Lagkas, "Advancing sdn from openflow to p4: A survey," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–37, 2023.

[8] F. A. Zwayed, M. Anbar, Y. Sanjalawe, and S. Manickam, "Intrusion detection systems in fog computing–a review," in *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers 3*. Springer, 2021, pp. 481–504.

[9] A. H. H. Kabla, M. Anbar, S. Manickam, T. A. Al-Amiedy, P. B. Cruspe, A. K. Al-Ani, and S. Karuppayah, "Applicability of intrusion detection system on ethereum attacks: A comprehensive review," *IEEE Access*, vol. 10, pp. 71 632–71 655, 2022.

[10] Y. QASMAOUI and A. HAQIQ, "Enhanced solid-flow: An enhanced flow rules security mechanism for sdn," *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp. 522–532, 2020.

[11] A. Tiwari and A. Chaturvedi, "A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification," *Expert Systems with Applications*, vol. 196, p. 116621, 2022.

[12] M. S. El Sayed, N.-A. Le-Khac, M. A. Azer, and A. D. Jurcut, "A flow-based anomaly detection approach with feature selection method against ddos attacks in sdns," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 4, pp. 1862–1880, 2022.

[13] K. Bouzoubaa, Y. Taher, and B. Nsiri, "Predicting dos-ddos attacks: Review and evaluation study of feature selection methods based on wrapper process," *Int. J. Adv. Comput. Sci. Appl*, vol. 12, no. 5, pp. 131–145, 2021.

[14] D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, R. Budiarto *et al.*, "Cicids-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132 911–132 921, 2020.

[15] T. A. Al-Amiedy, M. Anbar, B. Belaton, A. H. H. Kabla, I. H. Hasbullah, and Z. R. Alashhab, "A systematic literature review on machine and deep learning approaches for detecting attacks in rpl-based 6lowpan of internet of things," *Sensors*, vol. 22, no. 9, p. 3400, 2022.

[16] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "Insdn: A novel sdn intrusion dataset," *Ieee Access*, vol. 8, pp. 165 263–165 284, 2020.

[17] N. Al-Mi'ani, M. Anbar, Y. Sanjalawe, and S. Karuppayah, "Securing software defined networking using intrusion detection system-a review," in *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers 3*. Springer, 2021, pp. 417–446.

[18] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A review of feature selection methods for machine learning-based disease risk prediction," *Frontiers in Bioinformatics*, vol. 2, p. 927312, 2022.

[19] E. Jaw and X. Wang, "Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach," *Symmetry*, vol. 13, no. 10, p. 1764, 2021.

[20] J. C. C. Chica, J. C. Imbachi, and J. F. B. Vega, "Security in sdn: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 159, p. 102595, 2020.

[21] C. Prabha, A. Goel, and J. Singh, "A survey on sdn controller evolution: A brief review," in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2022, pp. 569–575.

[22] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in sdns based on cnn and a new regularization technique," *Journal of Network and Computer Applications*, vol. 191, p. 103160, 2021.

[23] J.-S. Hwang and T.-H. Hu, "A stepwise regression algorithm for high-dimensional variable selection," *Journal of Statistical Computation and Simulation*, vol. 85, no. 9, pp. 1793–1806, 2015.

[24] A. Mishra, N. Gupta, and B. Gupta, "Defense mechanisms against ddos attack based on entropy in sdn-cloud using pox controller," *Telecommunication systems*, vol. 77, pp. 47–62, 2021.

[25] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I. H. Hasbullah, A. A. Bahashwan, and S. Al-Sarawi, "Renyi joint entropy-based dynamic threshold approach to detect ddos attacks against sdn controller with various traffic rates," *Applied Sciences*, vol. 12, no. 12, p. 6127, 2022.

[26] S. Yu, J. Zhang, J. Liu, X. Zhang, Y. Li, and T. Xu, "A cooperative ddos attack detection scheme based on entropy and ensemble learning in sdn," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–21, 2021.

[27] J. Wang, L. Wang, and R. Wang, "A method of ddos attack detection and mitigation for the comprehensive coordinated protection of sdn controllers," *Entropy*, vol. 25, no. 8, p. 1210, 2023.

[28] H.-M. Chuang, F. Liu, and C.-H. Tsai, "Early detection of abnormal attacks in software-defined networking using machine learning approaches," *Symmetry*, vol. 14, no. 6, p. 1178, 2022.

[29] A. Maheshwari, B. Mehraj, M. S. Khan, and M. S. Idrisi, "An optimized weighted voting based ensemble model for ddos attack detection and mitigation in sdn environment," *Microprocessors and Microsystems*, vol. 89, p. 104412, 2022.

[30] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated ddos attack detection in software defined networking," *Journal of Network and Computer Applications*, vol. 187, p. 103108, 2021.

[31] E. Söğüt and O. A. Erdem, "A multi-model proposal for classification and detection of ddos attacks on scada systems," *Applied Sciences*, vol. 13, no. 10, p. 5993, 2023.

[32] N. M. Nawi, W. H. W. Atomi, and M. Z. Rehman, "The effect of data pre-processing on optimized training of artificial neural networks," *Procedia Technology*, vol. 11, pp. 32–39, 2013.

[33] N. Huyghues-Beaufond, S. Tindemans, P. Falugi, M. Sun, and G. Strbac, "Robust and automatic data cleansing method for short-term load forecasting of distribution feeders," *Applied Energy*, vol. 261, p. 114405, 2020.

[34] M. Hammad, N. Hewahi, and W. Elmedany, "Mmm-rf: A novel high accuracy multinomial mixture model for network intrusion detection systems," *Computers & Security*, vol. 120, p. 102777, 2022.

[35] T. Fuat, "Analysis of intrusion detection systems in unsw-nb15 and nsl-kdd datasets with machine learning algorithms," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 12, no. 2, pp. 465–477, 2023.

[36] L. U. Oghenekaro and A. T. Benson, "Text categorization model based on linear support vector machine," *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 85, no. 1, 2022.

[37] T. Kim, S. Suh, H. Kim, J. Kim, and J. Kim, "An encoding technique for CNN-based network anomaly detection," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2960–2965.

[38] S. Zwane, P. Tarwireyi, and M. Adigun, "A flow-based ids for sdn-enabled tactical networks," in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2019.

[39] L. B. de Amorim, G. D. Cavalcanti, and R. M. Cruz, "The choice of scaling technique matters for classification performance," *Applied Soft Computing*, vol. 133, p. 109924, 2023.

[40] P. Dhal and C. Azad, "A comprehensive survey on feature selection in the various fields of machine learning," *Applied Intelligence*, vol. 52, no. 4, pp. 4543–4581, 2022.

[41] N. Panwar and A. Professor, "Anomaly infiltration detection in networks using machine learning," *International Journal of Mechanical Engineering*, vol. 7, no. 2, p. 974–5823, 2022.

[42] R. Ramaswamy, P. Kandhasamy, and S. Palaniswamy, "Feature selection for alzheimer's gene expression data using modified binary particle swarm optimization," *IETE Journal of Research*, vol. 69, no. 1, pp. 9–20, 2023.

[43] S. I. Vrieze, "Model selection and psychological theory: A discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic)," *Psychological Methods*, vol. 17, no. 2, p. 228–243, 2012.

[44] H. Luepsen, "Anova with binary variables: the f-test and some alternatives," *Communications in Statistics-Simulation and Computation*, vol. 52, no. 3, pp. 745–769, 2023.

[45] I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5031–5042, 2022.

[46] H. Singh and S. Bawa, "Predicting covid-19 statistics using machine learning regression model: Li-muli-poly," *Multimedia Systems*, vol. 28, no. 1, p. 113–120, 2022.

[47] R. Zakharov and P. Dupont, "Ensemble logistic regression for feature selection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7036 LNBI, 2011, p. 133–144.

[48] S. Jahan, A. Khan, S. Jahan, and A. Khan, "Power of t-test for simple linear regression model with non-normal error distribution: A quantile function distribution approach," *Journal of Scientific Research*, vol. 4, no. 3, p. 609–622, 2012.

[49] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, p. 107398, 2021.

[50] L. Mohammadpour, T. Ling, C. Liew, and A. Aryanfar, "A survey of cnn-based network intrusion detection," *Applied Sciences*, vol. 12, no. 16, p. 8162, 2022.

[51] A. Alshra'a and S. Jochen, "One-dimensional convolutional neural network for detection and mitigation of ddos attacks in sdn," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ser. LNCS, vol. 13175, 2022, p. 11–28.

[52] A. detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. (2023) A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. Accessed: Oct. 20, 2023. [Online]. Available: \url{https://www.researchgate.net/publication/329045441_A~detailed~analysis~of/CICIDS2017_dataset_for_designing_Intrusion_Detection_Systems}

[53] M. Shushlevska, D. Efnusheva, G. Jakimovski, and Z. Todorov, "Anomaly detection with various machine learning classification techniques over unsw-nb15 dataset," *International Journal of Computer Applications*, vol. 914, no. 4, 2022.

[54] S. Iqbal, A. N. Qureshi, A. Ullah, J. Li, and T. Mahmood, "Improving the robustness and quality of biomedical cnn models through adaptive hyperparameter tuning," *Applied Sciences*, vol. 12, no. 22, p. 11870, 2022.

[55] D. Kilichev and W. Kim, "Hyperparameter optimization for 1d-cnn-based network intrusion detection using ga and pso," *Mathematics*, vol. 11, no. 17, p. 3724, 2023.

[56] D. Zou, Y. Cao, D. Zhou, and Q. Gu, "Gradient descent optimizes over-parameterized deep relu networks," *Machine learning*, vol. 109, pp. 467–492, 2020.

[57] 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.

[58] A. Henry, S. Gautam, S. Khanna, K. Rabie, T. Shongwe, P. Bhattacharya, B. Sharma, and S. Chowdhury, "Composition of hybrid deep learning model and feature optimization for intrusion detection system," *Sensors*, vol. 23, no. 2, p. 890, 2023.

[59] 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.

[60] S. Reddy, R. Reddy, and S. Babu, "An improved intrusion detection system for sdn using multi-stage optimized deep forest classifier," *IJCSNS International Journal of Computer Science and Network Security*, vol. 22, no. 4.