Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 3, December 2025, pp. 1345~1356
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i3.pp1345-1356

Hybrid TCP SYN attack detection model in SDN

Saira Muzafar, Noor Zaman Jhanjhi
School of Computer Science, SCS, Taylor's University, Subang Jaya, Malaysia

Article Info

Article history:

Received Oct 28, 2024
Revised Aug 1, 2025
Accepted Oct 15, 2025

Keywords:

DDoS
Deep learning
SDN
TCP SYN attacks
Temporal convolutional network
Weighted stacking

ABSTRACT

Software defined network (SDN) is a developing concept that emerged recently to overcome the constraints of traditional networks. The distinguishing characteristic of SDN is the uncoupling of the control plane from the data plane. This facilitates effective network administration and enables efficient programmability of the network. Nevertheless, the updated architecture is susceptible to cyberattacks including distributed denial of service (DDoS) attacks, that can impair network regular functions and hinder the SDN controller from assisting authorized users. This paper introduces hybrid deep learning model, to detect DDoS assaults triggered by TCP SYN attacks in SDN environments. Our proposed model integrates a temporal convolutional network (TCN) with a stacking classifier that leverages logistic regression, which is an innovative hybrid approach. We assessed the performance of our model by utilizing the benchmark CICDDoS2019 dataset. When compared to other benchmarking techniques, our model significantly improves attack detection. The experimental results indicate that the proposed hybrid model attains 99.9% accuracy for attack detection compared to the available approaches.

Corresponding Author:

Noor Zaman Jhanjhi
School of Computer Science, SCS, Taylor's University
Subang Jaya, Malaysia
Email: noorzaman.jhanjhi@taylors.edu.my

1. INTRODUCTION

Over the last twenty years, the quantity of network devices has been significantly expanding, leading to more administrative complexity and obstructing future Internet progress. Moreover, the rigidity of the conventional network diminishes its capacity to expand and adjust, resulting in increased operational expenses. The constraints of traditional networks hamper the growth of modern applications such as cloud computing, IoT, big data and network function virtuliazation (NFV), which require higher bandwidth, flexibility and real time network management. Software defined networking (SDN) has emerged to address the modern network requirements based on novel architecture to isolate the control plane and the data palne from each other, facilitate centralized network control, programmability, and agility. Despite exciting and promising charectiistics, the centralized architecture of SDN is sensitive to cyber attacks, and can be a single point of failure, specifically for distributed denial of service (DDoS) assaults. DDoS assaults can overwhelm the controller through fake traffic, halt the normal network functionalities by depleting the bandwidth, computing power, and the controller memory. This can lead to a decrease in performance or complete network disruption [1].

DDoS attacks are generally classified into three categories: volumetric attacks, which saturate the bandwidth; application-layer attacks, which exhaust resources at the application level; and protocol attacks, which exploit weaknesses in network protocols to disrupt services. TCP SYN flood is a protocol-level DDoS assault that exploits the TCP three-way handshake by transmitting several SYN packets without terminating

the link, specifically by skipping the concluding ACK [2]. Figure 1 presents an overview of the TCP three-way handshake and its exploitation during a TCP SYN flood attack. Figure 1(a) illustrates the normal handshake process, in which the client initiates the connection by sending a SYN packet, the server responds with a SYN-ACK, and the client completes the connection with an ACK. In contrast, Figure 1(b) depicts the TCP SYN flood attack wherein the attacker repeatedly sends SYN packets without completing the final ACK step. This results in a large number of half-open connections that accumulate on the target machine, rapidly depleting its memory resources.

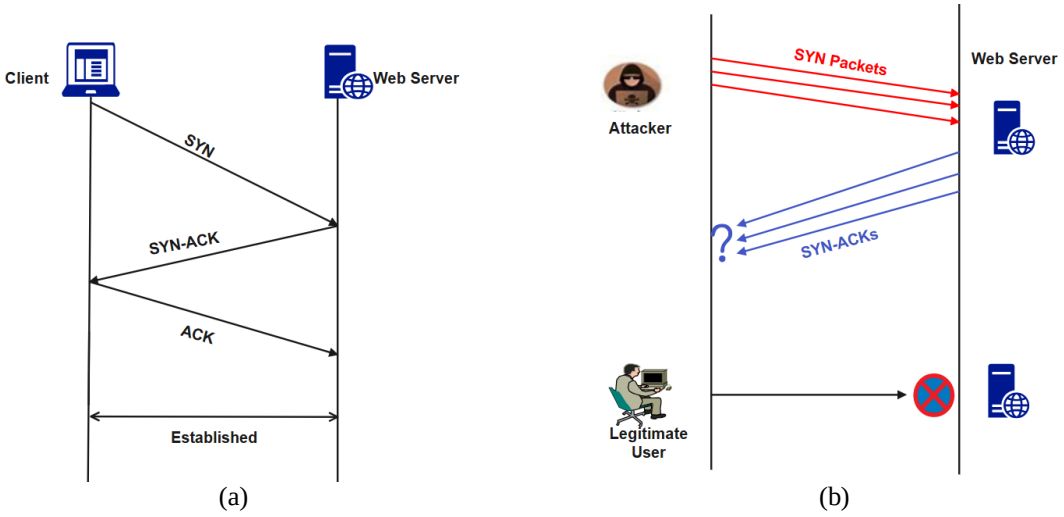

Figure 1. An overview of the TCP three-way handshake and its exploitation during a TCP SYN flood attack (a) standard successful three-way handshake and (b) TCP-SYN flood attack

In an SDN environment, where the controller centrally manages flow operations, such an overwhelming volume of incomplete handshakes can severely impair the control of plane control, making TCP SYN floods one of the most damaging DDoS threats to SDN controllers. Furthermore, the limited processing capability of most controllers and the openness of OpenFlow, which attracts attackers, and worsen the problem even more. Therefore, TCP SYN flood attacks are thus now acknowledged as among the most disastrous DDoS threats in SDN networks [3].

The precise identification of TCP SYN flood attacks is essential; however, it must conform to the resource limitations of SDN controllers. Current solutions include entropy-based analysis [4], [5], statistical models and flow based mechnisms [6]. Traditional machine learning and deep learning methods are commonly used, both single and hybrid such as CNN-BiLSTM [7], stacked and contractive autoencoders [8]-[10]. Machine learning models achieve high accuracy however exhibit constraints to adapt temporal traffic patterns while trade off between detection accuracy and computing efficiency [11]-[13]. Deep learning models enhance detection capabilities by automatically recognizing attack-related features with lower false positive rates compared to traditional ML models, making them more efficient but are frequently resource-intensive, rendering them inappropriate for lightweight SDN controller settings [1], [2], [14], [15]. To address these constraints, this study proposes a hybrid deep learning model integrating temporal convolutional network (TCN) with a dynamic weighted stacking classifier based on logistic regression (TCN+DWSR) to detect TCP SYN attacks in SDN. TCNs are particularly effective for modeling sequential data with long-range dependencies through causal and dilated convolutions, making them ideal for identifying SYN flood patterns in SDN environment [16]. Compared to earlier work based on statistical and traditional ML/DL models that consume excessive resources and often neglect temporal traffic patterns, our model captures both sequential and computational dependencies. The proposed model also addresses the need of lighweight solution to detect attacks with a less computational burden on controllers. By applying Boruta and LightGBM for relevant feature selection, 84 attributes reduced to only 14 features, lowering complexity with 99.9% accuracy

The research contributions are summed as follows:

a) A novel hybrid deep learning model, the proposed (TCN+DWSR) model is an innovative method that combines the benefits of TCNs with stacked classifiers to provide a robust solution for identifying

DDoS attacks in SDN environment. The adoption of early stopping and the use of validation data substantially enhance the model's performance and generalizability.

b) Comprehensive temporal analysis: The proposed TCN based approach captures bothe static and temporal patterns, making it suitable for precise identification of DDoS attacks.
c) Advanced feature extraction: Employing TCN for feature extraction enables the model to find intricate patterns in network traffic data that simpler models may overlook.
d) Enhanced classification accuracy: Incorporating stacking classifier in model with Logistic Regression enhanced model classification accuracy.

The rest of the paper is organized as follows: section 2 covers literature review. Section 3 presents the proposed model and methodology. The experimental findings are provided in section 4. Section 5 provides a discussion, while section 6 includes the conclusion and future work.

2. LITERATURE REVIEW

Academic research proposes diverse methods to detect and mitigate TCP SYN flood attacks in SDN, such as entropy based SAFETY system to analyze the randomness in flow data [4], a resource efficient approach SLICOTS is proposed in [17], that scans TCP connection requests and blocks malevolent connection. Another method is selective packet inspection, utilizing distributed monitors and centralized controllers to detect attack signatures [6]. Sinha [5] proposed a security system named SynFloWatch designed to protect hybrid SDN environments from TCP-SYN-based DDoS attacks. It utilizes Tsallis entropy analysis to effectively identify both low-rate and high-rate attacks. The proposed technique divides incoming network traffic into windows according to the count of packet_in messages. This allows for the early identification of attacks. Then by analyzing Tsallis entropy on the destination IP parameter to identify both low-rate and high-rate TCP-SYN DDoS assaults in hybrid SDN settings. This technique achieves higher accuracy than Shannon entropy-based methods. Swami *et al.* [18] examines the effects of both spoofed and non-spoofed TCP-SYN DDoS assaults on controller. It also demonstrates the implementation of machine learning based intrusion detection system to protect from attacks. Extracting essential attributes from packet headers, for instance the source IP address and port number, TCP flags, etc. This study involves five machine learning models (random forest, decision tree, AdaBoost, logistic regression, multi-layer perceptron) to break down traffic into two categories: normal and attack. Researchers in design [14] an experimental evaluation utilizing the CAIDA UCSD DDoS 2007 Attack Dataset to compare the performance of the SYNTROPY framework to the SAFETY algorithm in identifying TCP SYN DDoS attacks. However, this study lacks a thorough examination of the computational expense of the SYNTROPY algorithm on the controller.

Shalini *et al.* [19] presents a strategy for promptly identifying and mitigating TCP SYN flood DDoS assaults in SDN. They develop and deploy a detection model for TCP SYN flood attacks at the source, utilizing the elongated chi-square goodness of fit test on network traffic characteristics that are gathered at the SDN controller. To configure the switch's behavior the ultimate ACK packet is directed to the controller. This will enable the controller to accurately calculate the amount of half-open connections. As compared to entropy-based methods, the efficacy of the suggested model in identifying network attacks from many perpetrators. The TFAD approach in [20] employs two proxies, one for mitigating TCP SYN flood assaults and another for mitigating TCP ACK flood assaults. The ML-TFAD module implements the C4.5 decision tree approach for SYN flood attack identification prior to their arrival at the server. The proposed approaches accelerate the disposing of partially constructed links from the server queue to accomodate authorized requests. Chuang *et al.* [21] presents a SDN architecture that incorporates an artificial intelligence (AI) module to detect aberrant attacks at an early stage. It assesses several ML and DL models and devises a hierarchical multi-class (HMC) architecture to enhance performance on datasets with imbalanced data. Niyas *et al.* [8] introduced an Intrusion Detection System implemented in the controller. For classification, a deep learning-based stacked autoencoder model is proposed. The system comprises of three major parts: traffic flow collection, feature extraction, and categorization of the destructive flows. The proposed IDS employs packet analysis to minimize false positive rates. The researchers employed Hping3 to initiate various flooding assaults, including TCP-SYN, UDP, and ICMP floods. The suggested model surpasses earlier approaches, with an accuracy of 99.65%. However, it encounters constraints regarding its processing capabilities, involving the time and resources necessary to manage each packet and accomplish feature extraction. Aktar and Nur [9] introduces an innovative intrusion detection model utilizing deep learning and a contractive autoencoder. The model attains an accuracy level of 92.45%. Alghazzawi *et al.* [7] introduced a deep learning model employing a hybrid architecture of CNN and BiLSTM to detect and classify DDoS assaults in SDN. Hamaeshe *et al.* [22] utilized the Random Forest method achieved to achieve the greatest accuracy rate of 68.9%. Kumar *et al.* [23] in suggests a LSTM based deep learning model that can reach a remarkable accuracy rate of up to 98% to detect DDoS attacks in CICDDoS2019 dataset, surpassing conventional machine learning methods.

3. PROPOSED MODEL DESCRIPTION AND METHODOLOGY

This section covers a comprehensive explanation of the proposed hybrid model for TCP SYN flood attack detection in SDN, including dataset selection, preprocessing, and feature engineering.

3.1. Dataset

A major obstacle faced by ML/DL based intrusion detection techniques is the limited accessibility of datasets. The primary factor contributing to the scarcity of datasets in the intrusion detection field can be attributed to concerns around privacy and legal implications. Network traffic comprises highly confidential information, the disclosure of which can expose customer and business secrets, as well as personal communications. To address the above deficiency, numerous researchers generate their own data through simulation to mitigate any potential privacy problems. However, in these instances, the majority of the datasets produced are not meticulous, and the selected samples are insufficient to encompass the application operations. The following datasets are often used for intrusion detection: KDDCUP'99 [24], NSL-KDD [25], Kyoto 2006+, ISCX2012 [26], and CICIDS2017. Additional information regarding various datasets utilized in the field of intrusion may be found in reference [27]. The performance of our model assessed using the state-of-the-art CICDDoS2019 dataset. It contains over 80 flow attributes which were obtained through CICFlowMeter tools. The dataset comprises of a substantial volume of diverse DDoS assaults that may be executed using application layer protocols utilizing TCP/UDP [28].

3.2. Data preprocessing

The CICDDoS2019 dataset is provided in a flow-based format, with over 80 attributes extracted utilizing the CICFlowMeter tool. The following are the steps to prepare data:

a) Eliminating socket features: we eliminate socket attributes including source and destination IP, source and destination port, timestamp, and flow ID. The property of these features varies across different networks; thus, it is necessary to train the model using the packet attributes directly. Moreover, it is possible for both the attacker and regular user to share an identical IP address. Consequently, training the deep learning model using socket data may lead to overfitting, as the model may become biased towards the socket information. After eliminating the undesired features, we acquired a total of 14 attributes for the model input.
b) Data cleaning: the actual dataset includes a significant number of duplicate values. Hence, unnecessary values are removed from the dataset.
c) Encoding categorical variables: we applied label encoding technique to transform categorical variables to numerical form. Label encoding was chosen for the project due to its simplicity, efficiency, and relevance to the nature of the categorical features in the dataset.
d) Feature scaling: the proposed model is trained using a stanadradization technique of feature scaling.

After preprocessing the dataset is ready for model training.

3.3. Feature selection

Feature selection is an influential step for optimized model performance specifically in case of high-dimensional datasets, it supports interpretability and reduce overfitting which is particulary important in network traffic analysis, where actionable insights are crucial for decision-making.

3.3.1. Techniques used for feature selection

We combined Boruta algorithm with LightGBM algorithm for relevent feature selection form dataset. Boruta is an iterative feature selection algorithm that runs by evaluating the significance of randomly generated shadow features with actual features. It iteratively identifies features that are significantly more important than random noise. In the proposed project, BorutaPy, a Python implementation of Boruta, is used along with the LightGBM classifier as shown in Figure 2. LightGBM is a gradient-boosting framework known for its efficiency and scalability, makes it appropriate for handling large datasets. BorutaPy with LightGBM efficiently selects the most essential features for the classification task and address overfitting issue by extracting only the pertinent features, this reduces model complexity and enhance generalization. It starts with the origianl feature set as shown in Figure 2, then create the shadow features by creating copies of original features and shuffle them. Later combine the shuffled and original features feed the extended dataset to LightGBM for z-score calculation, and repeat the process multiple times to eliminate the weak features. TCNs has been known to handle sequential data of varying length better than CNN [29]. We choose TCN for this research due to below mentioned characteristics:

– Captures Sequential Patterns: TCNs excel in processing sequential data by capturing temporal dependencies, rendering them highly effective at recognizing patterns in network traffic flow.

- Effective for long-term dependencies: TCNs, in contrast to traditional convolutional layers, can capture long-range dependencies by the application of dilated convolutions and causal padding. This is crucial for precisely identifying complex patterns of DDoS attacks.
- Robust to noise: convolutional layers are useful for extracting dependable features from noisy data, which is common in network traffic analysis.
- Stacking with logistic regression: integrating TCN with logistic regression leverages the strengths of each of the models. TCN employs powerful algorithms to derive complex features from unprocessed data, while Logistic Regression excels in effectively classifying data based on these derived features.
- Improved accuracy: utilizing the TCN's predictions as input for the logistic regression model, the stacking methodology frequently attains superior accuracy relative to employing a solitary model independibility.
- Versatility: logistic regression is a simple yet powerful model capable of managing the altered feature space provided by the TCN, leading to an effective method that is both flexible and effective.

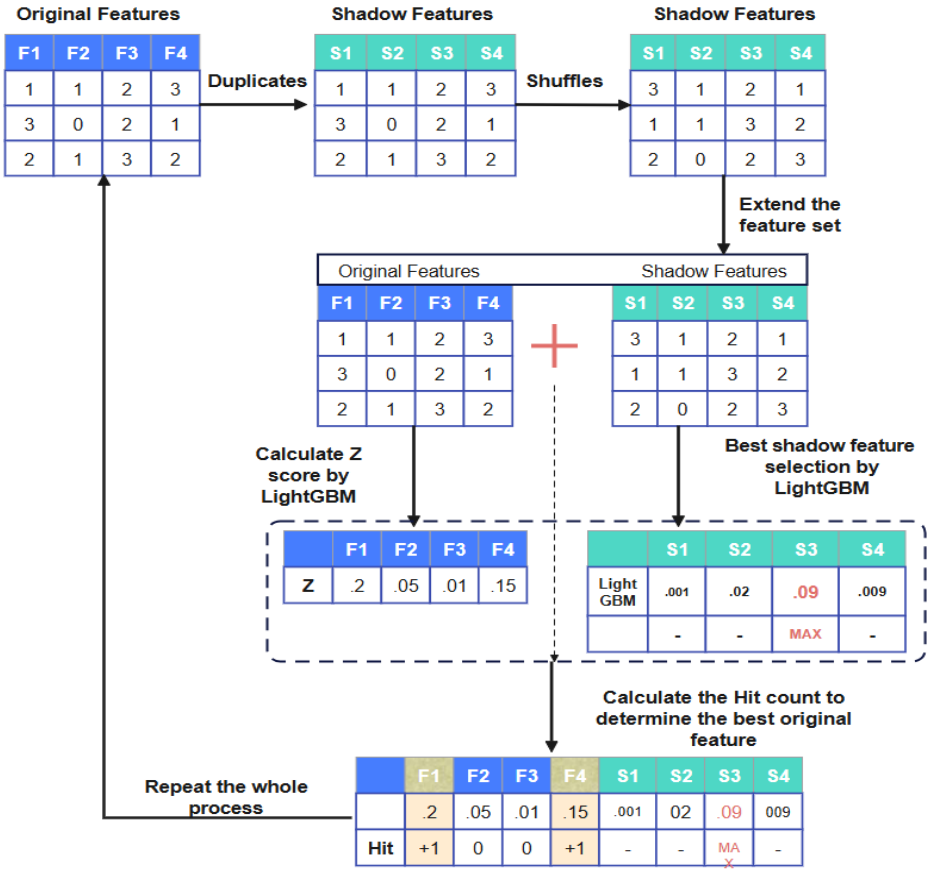

Figure 2. Boruta feature-LightGBM iterative feature selection process

3.4. Model architecture description

The model's architecture shown in Figure 3 comprises three residual blocks, each containing dilated 1D convolutional layer with ReLU activation function and residual skip connections. The dilation factors (1,2, and 4) enable the netwrok to capture both short- and long-range temporal dependencies in the traffic sequences without relying on pooling operations. This allows for the extraction of features from the input sequences while also maintaining their temporal relationships. The flattened outcome is fed into a dropout layer and a dense layer consists of 128 units. The utmost output layer uses SoftMax activation for binary classification, with two classes: Attack and Benign. The model is set out with the Adam optimizer and employs the sparse categorical cross-entropy loss function. Early stopping method is implemented throughout the process of training to avoid the issue of overfitting. The stacking classifier is learned by utilizing the stacked features and target labels. The stacking model makes predictions for the test data labels. This proposed model leverages the advantages of temporal convolutional networks (TCN) and stacking

techniques to enhance the performance of classification tasks. Below is an extended analysis of the TCN model:

a) Input layer: the input layer accepts sequences of features with a fixed number of time steps (14 in this case) and a single feature dimension.
b) Residual blocks with dilated convolutions: instead of traditional convolution + pooling, the model employs dilated 1D convolutions inside residual blocks. Each block consists of Conv1D layers with increasing dilation factors (1, 2, and 4) to capture both short- and long-range temporal dependencies in the traffic sequences. Residual skip connections are used to stabilize training and preserve information across layers.
c) Activation function: rectified linear unit (ReLU) an activation function is employed for every convolutional layer to establish non-linearity and facilitate the model in acquiring intricate patterns.
d) Dropout layer: dropout regularization is applied to combat overfitting by randomly discarding a fraction of the neurons' outputs during the training phase, thereby supporting the model to acquire increased resilient and generalized representations.
e) Dense layers: after flattening the output of the convolutional layers, dense (fully connected) layers are incorporated to further transform the features and prepare them for classification.
f) Output layer: the final output layer comprises neurons that correspond to the number of classes (2 in this case, representing SYN packets and non-SYN packets). The softmax activation function is used to transform the raw output scores into probabilities, denoting the likelihood of each class.
g) Stacking classifier: finally, the softmax outputs are fed into a logistic regression stacking classifier, which serves as a meta-learner to enhance the decision boundary and improve overall classification accuracy.

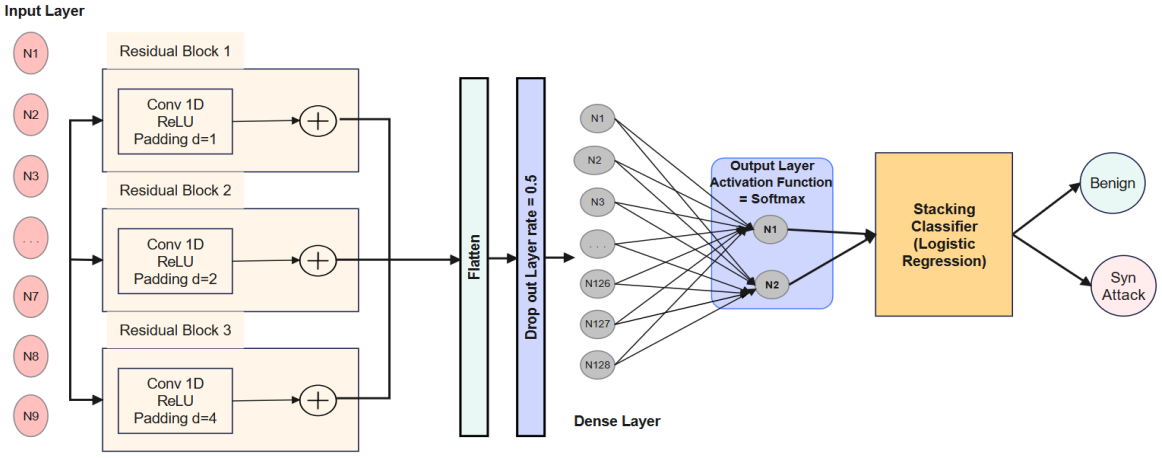

Figure 3. Proposed hybrid model combining TCN with stacking classifier for detecting TCP SYN attacks

3.5. Training the TCN model

The TCN model is trained utilizing the Adam optimizer, that is an adaptive learning rate optimization algorithm, with cross-entropy loss function for classification purposes. The training data is bifurcated into training and validation sets, with a portion reserved for early stopping. Incorporating early stopping helps to intercept overfitting by observing validation loss and halting training when refinement plateau, ensuring that the model generalizes well to new data. Once the TCN model is trained, its predictions on the training data are used as additional features to train a stacking classifier. The stacking classifier is an ensemble learning method used to aggregate the predictions of several base estimators (in this case, the TCN model's predictions) using a meta-estimator (logistic regression). The meta-estimator learns to weigh the predictions from the base estimators and makes the concluding decision. After training the stacking classifier, it is employed for making forecast on the test data. The accuracy of the hybrid model is then conducted by performing a comparison between the anticipated labels and the true labels obtained from the test data. Accuracy, which quantifies the proportion of precisely identified samples, serves as the performance metric for assessing the model's effectiveness in detecting SYN packets in SDN traffic data. Algorithm 1 presents the key steps from data preprocessing to attack prediction. Overall, the proposed hybrid model based on dynamic weighted stacking and regularization using TCN combines the advantages deep learning (TCN

model) and conventional machine learning (stacking classifier) techniques to achieve accurate detection of SYN packets, leveraging the TCN model's competence to cover up temporal dependencies and the stacking classifier's potential to blend diverse predictions for improved performance. The merger of these methodologies produces a robust, relible, and efficient model for DDoS attack detection in SDN, with high validation accuracy (0.9984).

Algorithm 1. Model training for attack detection

```
Input: cicddos_2019_dataset.csv
Output: TRUE = SYN Attack, FALSE = Benign
BEGIN
    Load and clean dataset.
    Select key features using Boruta + LightGBM.
    Train TCN model → get prediction score.
    IF score ≥ threshold THEN
         TRUE: traffic = SYN attack
    ELSE
         FALSE: traffic = benign
    END IF
END
```

4. RESULTS AND DISCUSSION

4.1. Configuration of the simulation

The network topology is established using Mininet. The topology consists of two hosts, one switch, and a Ryu controller with 65,535 transfer data size as shown in Figure 4. Specifications for the Mininet emulator version 2.3.0: A HP Pavilion system with the following characteristics was utilized for all tests and experiments: Requirements include 16 GB RAM, Windows 11 64-bit, a 1.8 GHz Intel Core (TM) i7-8550U processor, and VirtualBox Oracle VM version 6.0.18. Mininet functioning as a guest operating system on the Ubuntu 14.04 32-bit Linux OS, is installed with 4,096 MB of RAM and is controlled by VirtualBox, utilizing the Ryu controller. Our proposed hybrid model with stacking classifier shows outstanding results, we use Boruta with LightGBM for feature selection that significantly reduced the number of feature to 14 out of 84. to detect TCP SYN flood attacks in SDN environment. Compared to single deep learning models like LSTM, the hybrid architecture leverages the temporal awareness of TCN and the generalization power of ensemble learning, achieving higher detection accuracy and lower false positive rates. This not only reduces computational overhead but also enhances the model's interpretability. While earlier models (e.g., CNN or DNN-based) focused on raw accuracy, they often lacked robustness against redundant or irrelevant features, which our model addresses effectively.

Figure 4. Network topology creation

Table 1 shows that the model obtained an overall accuracy of 99.9%, with precision, recall, and F1-score all equal to 1.00 (100%) for both the benign and attack classes. We specifically evaluated macro-average and weighted-average metrics to account for class imbalance in the dataset, and these too are approximately 1.00, indicating that the model performs equally well on minority and majority classes.

Table 1. Testing performance of the proposed model

	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	19562
1	1.00	1.00	1.00	9879
Accuracy			1.00	29441
Macro Avg	1.00	1.00	1.00	29441
Weighted Avg	1.00	1.00	1.00	29441

The confusion matrix is shown in Figure 5 further confirms this balanced performance: it reports zero false positives and only a negligible number of false negatives, meaning that virtually all normal traffic is correctly identified while almost no attack packets are missing. Such a nearly flawless classification pattern signifies a robust discriminative capability. This level of reliability is critical for real-time SYN flood detection in SDN systems, where false alarms could needlessly disrupt legitimate traffic and missed attacks could cause serious damage.

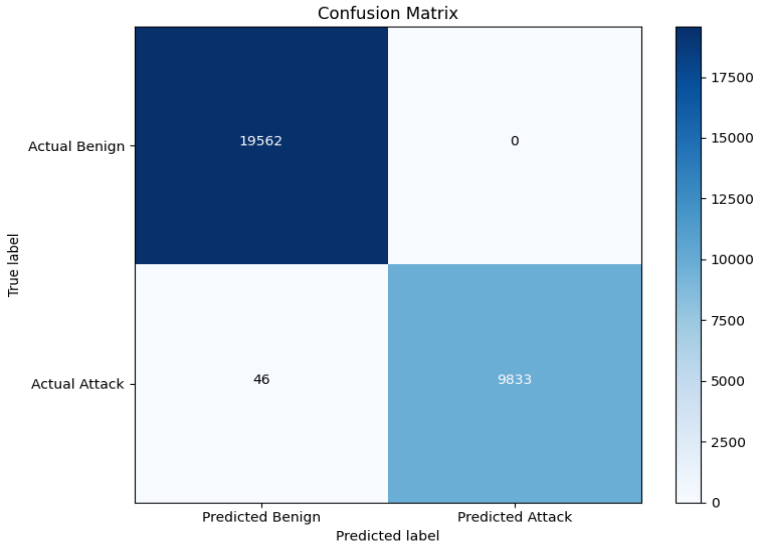

Figure 5. Confusion matrix for classification performane of proposed model

Additionally, the training and validation curves presented in Figures 6 and 7 exhibited steadily decreasing loss and consistently high validation accuracy during training, indicating stable convergence without overfitting. In summary, our results demonstrate that the hybrid TCN + logistic regression stacking model effectively learns the distinguishing temporal patterns of TCP SYN flood attacks, delivering highly accurate and balanced detection performance.

Compared to previous approaches, the proposed model offers superior performance and novel advantages. As shown in Table 2, Niyaz *et al.* [8] achieved about 99.65% accuracy using a stacked autoencoder-based deep model for DDoS detection, whereas a conventional random forest method reached only around 68.9% in a similar task [30]. Most other recent techniques report detection accuracies in the 92–98% range for TCP SYN flood attacks [28], [29]. Against this backdrop, our model's ~99.9% accuracy (with precision and recall essentially 100%) represents a notable improvement, pushing the performance closer to a practically perfect detection of SYN flood traffic. Beyond the metrics, our approach contributes a novel hybrid architecture: the integration of a temporal convolutional network (TCN) with a dynamic weighted stacking classifier using logistic regression. This design leverages TCN's robustness in catching long-range temporal patterns in network traffic and the simplicity of logistic regression for final classification. TCNs have been shown to outperform conventional recurrent models like LSTM in similar time-series and intrusion detection tasks, due to their ability to learn long-term dependencies without vanishing gradient issues and to train in parallel for faster convergence. By using TCN to extract high-level sequential features and then applying a lightweight logistic regression stacker, our model harnesses the best of both worlds, powerful feature learning and efficient decision-making.

Figure 6. Training and validation loss graph indicating stable model convergence

Figure 7. Training and validation accuracy graph indicating consistent classification performance

To our knowledge, this TCN+LR stacking combination is relatively novel for detecting DDoS attacks, and research outcomes validate its effectiveness. Another strength of our model is its computational efficiency. We employed the Boruta algorithm with LightGBM to select the 14 most salient features from the dataset, significantly reducing the input dimensionality without sacrificing accuracy. This feature selection led to roughly a 40% reduction in training time (compared to using the full feature set) and helps avoid overfitting by eliminating irrelevant features. Notably, even though our stacking classifier uses a straightforward Logistic Regression, the model did not compromise on performance – an unexpectedly positive outcome that underlines the quality of the temporal features extracted by the TCN. In other words, TCN learned such informative patterns that even a simple classifier was able to achieve optimal results. These comparisons and strengths highlight the contribution of our work: a high accuracy, balanced, and efficient DDoS detection model that improves prior studies in both performance and design. The hybrid TCN + logistic regression approach not only attained excellent evaluation metrics but also demonstrated how combining advanced sequential modeling with an ensemble strategy can bolster DDoS defense in SDN environments. Broadly, these results suggest that more effective and faster DDoS detection is attainable in practice, strengthening the stability of SDN-based networks against disruptive attacks. Nevertheless, there are certain limitations and open questions that need further investigation. Our evaluation is conducted on a single benchmark dataset (CICDDoS2019) and a binary classification of traffic (attack vs. benign). Future research

should validate the model's generalizability by testing it on different DDoS datasets and in real-world SDN deployments. Another important extension will be to implement a multi-class classification framework that can distinguish between various types of DDoS attacks (not just SYN floods), providing a more fine-grained defense mechanism. Exploring these directions will address the remaining questions about the model's broader applicability and robustness. In summary, the hybrid model exhibits a progressive capability for SYN flood capturing in SDN and represents a significant step forward in the quest to safeguard programmable networks from DDoS threats.

Table 2. Performance comparison of TCN + DWSR with existing models

Author	Model	Performance metrics (%)			
		Accuracy	Precision	Recall	F Score
Niyaz *et al.* [8]	Deep learning	99.65			99.75
Aktar *et al.* [9]	Model with stacked autoencoder deep learning Model with contractive autoencoder	92.45	92.46	92.45	92.45
Alghazzawi *et al.* [7]	CNN, BiLSTM	94.52	94.74	92.04	93.44
Hamarshe *et al.* [22]	RF	68.9	56.0	80.0	66.0
Ahuja *et al.* [30]	SVC, RF	98.8	98.27		97.65
Kumar *et al.* [23]	LSTM	98.0	98	97	97
Proposed model	TCN + DWSR	99.9	1.00	1.00	1.00

5. CONCLUSION AND FUTURE WORK

DDoS is considered one of the most destructive forms of cyber attacks currently, exerting a substantial impact on the entire network. The application of deep learning within SDN environments presents some practical challenges. However, deep learning models are compute intensive. To overcome this issue this paper presents a novel lightweight hybrid approach utilizing dynamic weighted stacking and regularization with TCN to identify TCP SYN DDoS attacks intended for SDN environments. The CICDDoS2019 dataset includes a thorough and updated collection of DDoS attack types. For optimum feature selection we integrate Boruta algorithm with LightGBM algorithm. This approach effectively addresses the common problem of overfitting in ML/DL models by identifying and selecting only the pertinent features, thereby minimizing the potential for model intricacy and enhancing the overall performance of generalization. Our model evaluation demonstrated that the proposed hybrid approach outperforms over the well-known existing techniques for recall, precision, F-score, and accuracy. The proposed model performance is assessed by considering both validation loss and accuracy, promising its effectiveness in practical scenarios. The results show a remarkable enhancement in SDN security, as the lightweight design of our method facilitates real-time application in resource-limited SDN controllers. The incorporation of Boruta-LightGBM feature selection decreases computational complexity, hence improving model generalizability. The practical implication of our research is the potential implementation of effective, low-latency DDoS detection systems in real-world SDN environments, hence enhancing network resilience against emerging cyber threats. Further, to broaden our research in future we aim to classify each attack class separately by using a multi-class categorization method and to assess the efficacy of proposed hybrid model on different datasets.

FUNDING INFORMATION

No external funding involved for the conduct of this study.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Saira Muzafar	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Noor Zaman Jhanjhi		✓		✓		✓	✓			✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known conflict of interest associated with this work.

DATA AVAILABILITY

The data supporting the findings of this study are openly available in the CICDDoS2019 dataset at https://www.unb.ca/cic/datasets/ddos-2019.html. Derived data generated during the analysis are available from the corresponding author (N.Z.J.) on request.

REFERENCES

[1] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Computer Science Review*, vol. 37, p. 100279, 2020, doi: 10.1016/j.cosrev.2020.100279.

[2] C. S. Shieh, W. W. Lin, T. T. Nguyen, C. H. Chen, M. F. Horng, and D. Miu, "Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model," *Applied Sciences 2021, Vol. 11, Page 5213*, vol. 11, no. 11, p. 5213, Jun. 2021, doi: 10.3390/APP11115213.

[3] T. Das, O. A. Hamdan, S. Sengupta, and E. Arslan, "Flood control: TCP-SYN flood detection for software-defined networks using openflow port statistics," *Proceedings of the 2022 IEEE International Conference on Cyber Security and Resilience, CSR 2022*, pp. 1–8, 2022, doi: 10.1109/CSR54599.2022.9850339.

[4] P. Kumar, M. Tripathi, A. Nehra, M. Conti, and C. Lal, "SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545–1559, Dec. 2018, doi: 10.1109/TNSM.2018.2861741.

[5] M. Sinha, "SynFloWatch: A detection system against TCP-SYN based DDoS attacks using entropy in hybrid SDN," *ACM International Conference Proceeding Series*, pp. 359–364, 2024, doi: 10.1145/3631461.3631463.

[6] T. Chin, X. Mountrouidou, X. Li, and K. Xiong, "Selective packet inspection to detect DoS flooding using software defined networking (SDN)," *Proceedings - 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops, ICDCSW 2015*, pp. 95–99, Jul. 2015, doi: 10.1109/ICDCSW.2015.27.

[7] D. Alghazzawi, O. Bamasaq, H. Ullah, and M. Z. Asghar, "Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," *Applied Sciences 2021, Vol. 11, Page 11634*, vol. 11, no. 24, p. 11634, Dec. 2021, doi: 10.3390/APP112411634.

[8] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," *ICST Transactions on Security and Safety*, vol. 4, no. 12, p. 153515, Dec. 2017, doi: 10.4108/eai.28-12-2017.153515.

[9] S. Aktar and A. Yasin Nur, "Towards DDoS attack detection using deep learning approach," *Computers and Security*, vol. 129, p. 103251, Jun. 2023, doi: 10.1016/J.COSE.2023.103251.

[10] S. Aktar and A. Y. Nur, "Advancing network anomaly detection: An ensemble approach combining optimized contractive autoencoders and K - means clustering," *2024 IEEE 3rd International Conference on Computing and Machine Intelligence, ICMI 2024 - Proceedings*, 2024, doi: 10.1109/ICMI60790.2024.10585975.

[11] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, "Machine learning algorithms to detect DDoS attacks in SDN," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 16, p. e5402, Aug. 2020, doi: 10.1002/CPE.5402.

[12] M. Myint Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, "Advanced support vector machine- (ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN)," *Journal of Computer Networks and Communications*, vol. 2019, no. 1, p. 8012568, Jan. 2019, doi: 10.1155/2019/8012568.

[13] J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020, doi: 10.1109/ACCESS.2020.3019330.

[14] V. A. Shirsath, M. M. Chandane, C. Lal, and M. Conti, "SYNTROPY: TCP SYN DDoS attack detection for software defined network based on Rényi entropy," *Computer Networks*, vol. 244, p. 110327, May 2024, doi: 10.1016/J.COMNET.2024.110327.

[15] T. E. Ali, Y. W. Chong, and S. Manickam, "Machine learning techniques to Detect a DDoS attack in SDN: A systematic review," *Applied Sciences 2023, Vol. 13, Page 3183*, vol. 13, no. 5, p. 3183, Mar. 2023, doi: 10.3390/APP13053183.

[16] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," Oct. 2019, doi: 10.1109/ccst.2019.8888419.

[17] R. Mohammadi, R. Javidan, and M. Conti, "SLICOTS: An SDN-based lightweight countermeasure for TCP SYN flooding attacks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 487–497, Jun. 2017, doi: 10.1109/TNSM.2017.2701549.

[18] R. Swami, M. Dave, and V. Ranga, "Detection and Analysis of TCP-SYN DDoS attack in software-defined networking," *Wireless Personal Communications*, vol. 118, no. 4, pp. 2295–2317, 2021, doi: 10.1007/s11277-021-08127-6.

[19] P. V. Shalini, V. Radha, and S. G. Sanjeevi, *Early detection and mitigation of TCP SYN flood attacks in SDN using chi-square test*, vol. 79, no. 9. Springer US, 2023.

[20] K. M. Sudar, P. Deepalakshmi, A. Singh, and P. N. Srinivasu, "TFAD: TCP flooding attack detection in software-defined networking using proxy-based and machine learning-based mechanisms," *Cluster Computing*, vol. 26, no. 2, pp. 1461–1477, 2023, doi: 10.1007/s10586-022-03666-4.

[21] H. M. Chuang, F. Liu, and C. H. Tsai, "Early detection of abnormal attacks in software-defined networking using machine learning approaches," *Symmetry 2022*, vol. 14, no. 6, p. 1178, Jun. 2022, doi: 10.3390/SYM14061178.

[22] A. Hamarshe, H. I. Ashqar, and M. Hamarsheh, "Detection of DDoS attacks in software defined networking using machine learning models," *Lecture Notes in Networks and Systems*, vol. 700 LNNS, pp. 640–651, 2023, doi: 10.1007/978-3-031-33743-7_51/FIGURES/5.

[23] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "DDoS detection using deep learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, Jan. 2023, doi: 10.1016/J.PROCS.2023.01.217.

[24] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for Anomaly-based network intrusion detection: KDD CUP 99 alternatives," *Proceedings on 2018 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2018*, pp. 1–8, Dec. 2018, doi: 10.1109/CCCS.2018.8586840.

[25] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," Jul. 2009, doi: 10.1109/CISDA.2009.5356528.
[26] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, May 2012, doi: 10.1016/J.COSE.2011.12.012.
[27] 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, Sep. 2019, doi: 10.1016/J.COSE.2019.06.005.
[28] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "DDoSNet: A deep-learning model for detecting network attacks," Aug. 2020, doi: 10.1109/wowmom49955.2020.00072.
[29] P. Hewage *et al.*, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Computing*, vol. 24, no. 21, pp. 16453–16482, Nov. 2020, doi: 10.1007/S00500-020-04954-0/FIGURES/17.
[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, no. November 2020, p. 103108, 2021, doi: 10.1016/j.jnca.2021.103108.

BIOGRAPHIES OF AUTHORS

Saira Muzafar is a Ph.D. research scholar at the School of Computer Science, SCS, Taylor's University, Subang Jaya, Malaysia. Her current areas of interest include cyber security, security issues in Software-defined networking, and DDoS attacks. She can be contacted at sairamuzafar@hotmail.com.

Prof. Dr. Noor Zaman Jhanjhi is a highly esteemed Senior Professor of Computer Science, specializing in Artificial Intelligence and Cybersecurity. He currently holds the position of Professor at the School of Computer Science at Taylor's University, Malaysia, and serves as the Program Director for Postgraduate Research Degree Programmes as well as the Director of the Center for Smart Society (CSS5). With a career marked by academic leadership and groundbreaking research, Prof. Jhanjhi has been pivotal in advancing research and education in computer science. Recognized globally, Prof. Jhanjhi has been ranked among the world's top 2% research scientists for three consecutive years (2022, 2023, and 2024). In Malaysia, he is ranked among the top three computer science researchers and was honored with the Outstanding Faculty Member award by MDEC Malaysia in 2022, as well as the Vice Chancellor's Best Research Citations Award from Taylor's University in 2023. He can be contacted at email: noorzaman.jhanjhi@taylors.edu.my.