Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 3, March 2026, pp. 1000∼1016
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i3.pp1000-1016

Classification of DoS/distributed DoS threats in software defined networks using advanced deep belief network-long short term memory architecture

Manjula Maraiah, Venkatesh
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India

Article Info

Article history:

Received Feb 7, 2025
Revised Jan 15, 2026
Accepted Feb 26, 2026

Keywords:

Adversarial deep belief network-long short term memory
Cybersecurity
Denial of service
Projected gradient descent
Software-defined networking
Wasserstein GAN

ABSTRACT

With the evolution of telecommunication core and access networks, the next generation networks leverages software defined networks (SDN) to provide flexibility, scalability and centralized control. Denial of service (DoS)/distributed DoS (DDoS) attacks have been a major threat to next generation networks especially to the centralized architecture of SDNs. The ever-changing and dynamic nature of the DoS/DDoS attacks makes it challenging to detect and resolve them. The existing models to handle DoS/DDoS attacks often suffer from false positive rates and adaptability. In order to solve these problems, this study aims to create and apply sophisticated deep learning framework namely adversarial DBN-LSTM to accurately detect and classify various DoS/DDoS attack types. The proposed adversarial DBN-LSTM model is based on the generative adversarial networks. The proposed model uses generator to generate the adversarial attack and discriminator to detect the attacks. The adversarial DBN-LSTM model is evaluated using a dataset specifically generated in a Mininet-based SDN controller environment to ensure relevance and practical applicability. The performance of the adversarial DBN-LSTM is compared with other prevalent models. The adversarial DBN-LSTM model achieves accuracy about 99.4%. The proposed work achieves a breakthrough in identifying and preventing DoS/DDoS threats in relation to SDN environment.

Corresponding Author:

Manjula Maraiah
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering
Bangalore University
Bengaluru, India
Email: manjula.m82@gmail.com

1. INTRODUCTION

The evolution of the software regulated (defined) networks (SDN) [1] is considered a radical shift in computer networking. By isolating the control plane and data planes, these networks reinterprets the conventional architecture [1]. In networking devices (for example, switches and routers), the decoupling of planes separates the functionalities. Whereas, conventional networking devices integrate these two planes. When network policies change or new protocols are added, physical equipment must be replaced or modified. Because of this, traditional architecture finds it challenging, to adapt to changing conditions and requirements. On the other hand, SDN centralizes the control plane under a software entity known as the controller. The controller is used to interact with the underlying network appliances. In standard networks, the networking appliances are closely

linked with the control logic. However, by centralizing the control functions, SDN facilitates administrators in governing and optimizing network resources dynamically. SDN uses a standardized, open interface, such as OpenFlow, to connect to switches and routers via a centralized controller. In addition to improving scalability and overall network efficiency, the centralized controller allows network administrators to react to shifting traffic patterns. The SDN increases the network's programmability and flexibility. Because SDN is so flexible and agile, it is an essential technology for addressing the evolving needs of contemporary networking environments.

The network architecture in the SDN is distinguished with functionality at three different layers depicted in the Figure 1. The SDN controller layer serves as the centralized system responsible for governing and handling the entire network. The application or the management plane hosts all the SDN applications. The application layer interacts through the controller via the northbound APIs to convey their requirements and policies and also collects network's status from the controller. The control plane receives the instructions from the application layer through the northbound APIs and utilizes southbound APIs to send these commands to the infrastructure layer. This layer includes the switches and routers as the networking devices. In compliance with the directives from the SDN Controller, this layer transmits the packets. The data planes ensure the implementation of centrally defined policies and configurations [1].

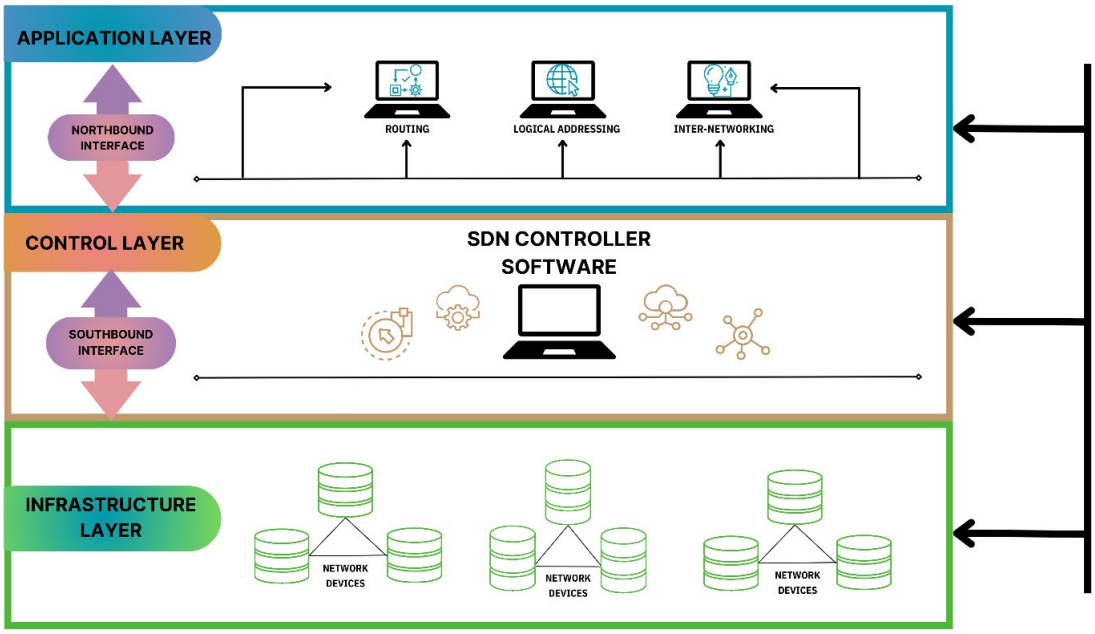

Figure 1. Architecture of the SDN with three layers

Separating the planes into data and control in an SDN network has its own advantages and disadvantages. Though it enhances network management and programmability, it also introduces a potential vulnerability. The entire network is concentrated in a central control plane, it is potential failure point, posing a security risk. The SDN controller is prone to service disruption threats, including both denial of service (DoS) and its counterpart variant (distributed DoS) attacks [2], [3]. The DoS/DDoS attack may be reflection based or exploitation based [4]. Also the DDoS attacks are categorized into volumetric, protocol based, application layer, reflection and amlification attacks [5], [6]. With in the framework of SDN, DDoS and DoS attacks targets the data plane, and the central control plane. The SDN control plane might be overloaded by such attempts, which would deplete resources of the network and ultimately bring down the network as a whole [7], [8].

When a large number of sources are initiating the attack simultaneously on a target machine it is called as the distributed form of denial-of-service (DDoS) attacks [9]. Online services are rendered inaccessible during a DDoS attack because of an excessive volume of malicious data traffic coming from many sources. The SDN controller, acting as the central point in an SDN network, becomes highly susceptible to DoS/DDoS cyber attacks, potentially impacting the entire network. The distributed DoS attack not only targets the SDN controller but can also impact the switches and routers in the infrastructure layer.

Most DoS/DDoS attacks occur in the TCP/IP stack's transport layer. Controlling data flow is the major task of the transport layer. The primary transport layer vulnerabilities include deliberate packet corruption or degradation, and the use of protocol flaws to initiate DDoS attacks targeting the network. The transport layer is frequently targeted by volumetric and protocol-driven attack vectors [4], [10], [11], which include:

- User datagram protocol (UDP) flood attacks occur when an intruder sends enormous amount of IP packets containg UDP datagrams to the random ports of targeted host [12].
- An attacker generates a huge count of ICMP echo request (ping) packets to overwhelm a intendent endpoint in an ICMP flood attack [13], sometimes referred to as a ping flood.
- Multiple synchronization (SYN) packets are sent to a server by an attacker using a fake IP address in a TCP synchronization (TCP-SYN) flood incident [14].

To defend the DoS/DDoS attack scenarios in SDN environments, either source-based [15], network-based [16], or destination-based [17] mechanisms can be employed. Also, various AI/ML based anomaly identification techniques [13], [18], to illustrate, random forest (RF) algorithm [19], [20], support vector machine [21], logistic regression [22], K-Nearest neighbor (KNN) [23], Naive-Bayes classifiers [24], and are used. Apart from these, statistical analyses like entropy and correlation techniques are used to counteract DoS/DDoS attack scenarios in SDN environments [25]. Among these, deep learning based models like CNN, GRU, AE-BGRU, LSTM, BiLSTM, etc have evidenced as the most efficient in reducing DoS/DDoS attack incidents in SDN environment [26]-[28]. Most existing ML/DL models are trained on datasets collected from traditional networks, so they do not capture the fast, dynamic behavior of SDN traffic. They usually ignore controller–switch interactions, flow-setup delays, and control-plane signaling, which are critical in SDN. Many models also fail when attackers make small changes in traffic patterns because they lack adversarial robustness.

Most of the research work that is suggested to trace DoS/DDoS incidents in the SDN, with machine and deep learning techniques, uses publicly available datasets [6]. The most commonly used datasets in DoS/DDoS detection are CICDDoS2019, UNSW-NB15, and NSL-KDD datasets. These publicly available datasets are generally captured in typical networks and do not fully capture the real-time, dynamic nature of SDN traffic. The key limitation, and the main motivation for our study, is that publicly available datasets fail to capture these SDN-specific behaviors. Also, these datasets suffer from class imbalance issues, where benign traffic significantly outweighs attack traffic. This can cause the deep learning models give biased results. In this research article, DoS/DDoS cyberattacks on the SDN are traced and classified using a deep learning technique. In order to strengthen the detection process, an adversarially robust deep belief network-long short term memory (DBN-LSTM) architecture is proposed in the work. By integrating Wasserstein generative adversarial network (WGAN) for realistic attack-traffic synthesis and projected gradient descent (PGD) for modeling adaptive adversarial behavior, the framework reflects real-world attack evolution more accurately than conventional training setups. To overcome the drawback of using publicly available dataset and to demonstrate the model's performance, the current research work is gauged against the dataset generated on the Mininet emulator for the SDN environment. Listed below are this paper's main contributions:

- Put forward an adversarial model with DBN-LSTM to trace and classify DoS/DDoS incidents in the SDN environment;
- Generate normal and DoS/DDoS attack dataset with Mininet emulator testbed for SDN environment;
- Suggest an adversarially robust deep learning framework that combines WGAN to produce realistic attack traffic and PGD to mimic adaptive DoS/DDoS attacks in SDN;
- Carries out comprehensive tests using both a testbed and public datasets to estimate the effectiveness of suggested model in recognizing DoS/DDoS attack incidents in SDN systems against the most advanced models.

The upcoming sections are structured as follows: review of previous research works are elaborated in section 2. Problem definition of the research paper is addressed in section 3. A thorough narration of the suggested work is discussed next. The investigative findings and dataset description are demonstrated in section 5. In section 6, the research project is successfully concluded.

2. RELATED WORK

Several deep learning algorithmic strategies have been proposed by various researchers to defend the SDN environment from DoS and DDoS threats. The research work in [29] focuses on the early identification

and separation of network data as a crucial step in reducing DDoS threats. In their work, the authors proposed a LSTM based model using CICDDoS2019 as a training and testing dataset. Although the work attains an impressive accuracy of 98%, the study ignores issues including the demand for a huge number of labeled data, possible overfitting, and high computing cost. To reduce DDoS attacks, the authors in [30] suggested a unique source-based DDoS protection mechanism for fog and cloud settings. Since an LSTM model performs admirably with the sequential data, the authors of the suggested work employed it to detect anomalies at the network and transport levels. The Hogzilla dataset is used to train this deep learning model, and both simulated and actual DDoS attack packets are used for testing. However, the model suffers from high computational overhead as it incorporates the LSTM model.

Gebremeskel *et al.* [31] suggested a DDoS attack identification and classification method for the software enabled network environment with a multi-controller architecture. The proposed technique used an entropy-based approach for preliminary detection. It used LSTM as the deep learning model for fine-grained classification. The proposed work also suggested distributed controllers for SDN scalability and availability, overcoming the single-point failure. Though the work achieves an accuracy of 99.4%, the drawbacks include computational complexity, reliance on specific datasets for evaluation, and potential performance degradation under high network loads. Additionally, while categorical classification improves upon binary detection, the approach may face challenges in real-world deployment due to evolving attack patterns and network dynamics.

Gebremeskel *et al.* [31] justified necessity of distributed controller design in software-defined networking (SDN) due to several limitations in centralized systems. In particular, it discusses DDoS attack identification in multicontroller SDN for new data centers and suggests a deep learning algorithm for DDoS threat incident with an entropy. This two-level detection approach is employed to balance accuracy and computational complexity. The research work implements a solution for multi controller-based SDN environment. The research work in [32] proposed an LSTM-Autoencoder-based deep learning model. The work demonstrated high detection rates with a reduced feature set. The random forest and information gain methods are involved to reduce the count from 48 features to 10 key features. Even if the model accomplishes an remarkable level of 99% accuracy rate, it incorporates high computational overhead and a lack of real-time evaluation as it relies on a public dataset.

A hybrid CNN-BiLSTM strategy for identifying network intrusion in the SDN was proposed in the research study in [33]. Despite evaluating the model including the UNSW-NB15, NSL-KDD, and InSDN datasets, the work tackles the dataset's issues of class imbalance and data redundancy. A key technical drawback of this research work is the computational complexity introduced by the hybrid CNN-BiLSTM architecture, which may lead to increased inference time in real-time SDN environments. While the model reduces training time compared to other CNN-based approaches, the sequential nature of BiLSTM layers can slow down detection in high-throughput networks. CNNs rely on spatial feature extraction, which is not as effective for capturing deep, latent patterns in network traffic data. CNN-BiLSTM models require more fine-tuning and hyperparameter optimization to achieve robustness of the model.

Chen *et al.* [34] presented the adversarial approach with DBN-LSTM model to predict and thwart the DDoS attacks in SDNs using the CICDDoS 2019 dataset. However, the dependence on FGSM for adversarial sample generation reduces the model's robustness against more sophisticated adversarial strategies. The author's work highlights the reactive defense strategy focusing on mitigation after attack detection. However, it lacks a proactive component to adapt to evolving threats. Lim *et al.* [35] and Zacaron *et al.* [36] highlighted the importance of WGAN for analyzing network traffic patterns. WGAN uses Wasserstein distance and gradient penalty to gauge the difference in distributions of real and generated data. This results in more stable training and higher network anomaly detection accuracy. Also, Park *et al* [37] proposed an AI-driven network intrusion detection system (NIDS) using WGAN to synthesize attack samples. The authors suggested that class imbalance in intrusion detection can be addressed with WGAN. These research works demonstrate how WGAN may improve resilience, scalability, and representation learning, making it a useful instrument for spotting anomalies in network security applications.

For anomaly identification in IoT networks, Yao *et al.* [38] presented an unsupervised deep learning technique utilizing bidirectional generative adversarial networks (BiGAN). Using the CIC-IDS2017 and UNSW-NB15 datasets, the model achieves improved accuracy and a decreased false alarm rate by including Wasserstein distance to enhance attack identification performance and stability. Although the method successfully identifies abnormalities in the absence of labeled data, it has drawbacks like a high computing cost and the potential for adversarial attacks. In spite of these drawbacks, it presents a viable option for precise and scalable intrusion detection in internet of things settings.

3. PROBLEM STATEMENT

Next-generation networks, especially SDN environments, are always prone to DoS/DDoS attacks. These attacks increase so rapidly that conventional DoS/DDoS detection mechanisms generally fail to detect them. Also, the traditional DoS/DDoS detection mechanisms do not classify diverse DoS/DDoS attack subtypes. Generally, the SDN environment generates high-dimensional network data, and the complexity of feature extraction and classification is not properly handled by conventional solutions.

There is a need to devise techniques that can effectively address these challenges by adapting to evolving attack patterns. The design should efficiently detect and classify diverse DoS/DDoS flodding attacks for TCP-SYN, UDP, and ICMP protocols, ensuring robust network security. The techniques should handle the complexity of feature extraction and requirement of high dimensional network data.

4. PROPOSED METHOD

In this research work, an adversarial DBN-LSTM framework is proposed to effectively identify and perform classification of the diferent DoS/DDoS incidents in the SDNs. The framework can classify the DoS/DDoS flooding attack types like ICMP, UDP, and TCP-SYN flood. The suggested work performance is evaluated using the dataset created on the Mininet emulator for the SDN controller environment. The proposed approach involves dataset created using Mininet testbed emulator, data pre-processing, model design and implementation, and performance evaluation. The model demonstrates improved accuracy when evaluated on standard datasets such as CICDDoS2019.

4.1. Dataset generated with Mininet testbed

To conduct the experiment, a DoS/DDoS dataset is generated in an environment with Mininet emulator using Ryu framework. Mininet is often used as a tool to simulate SDN networks. Specifically, it can mimic an entire network with computers, connections, and switches running on one Linux system using process-based virtualization. The network simulation using Mininet was run on a virtual machine (VM).

The network topology was created using Mininet as depicted in Figure 2. SDN architecture was implemented with the Ryu controller operating as the control plane. Using the RYU controller, the feature's source and destination IP addresses, port number, and timestamp are taken out from the packets. The flow collector first contacts the controller to request traffic statistics. OpenFlow (OF) switches are used for the data plane. Flow tables are gathered using the OF protocol. The controller sends a flow-stats request to every switch that is linked to it, asking it to provide flow statistics. Consequently, all flow tables' flow entries, along with the flow description and any related counters, are included in a flow-stats reply message which is transmitted back to the controller. Once the controller has gathered all of the switch traffic data, it responds to this component.

The normal and attack data traffic was generated employing the Scapy tool. The data traffic was generated on TCP-SYN, UDP, and ICMP protocols with random hosts in the network by using HPing-3. The implemented system consisted of modules for attack identification and also to mitigate them. The dataset used in this research work includes a diverse set of DDoS attack floods of ICMP, UDP, and TCP-SYN, among the most common and impactful forms of DDoS attacks. To create a comprehensive and balanced dataset, both benign and malicious traffic was generated in fair proportions. This balanced dataset helps to prevent the proposed model from being biased and to ensure that the model can efficiently perform classification of the normal and attack traffic data.

4.2. Adversarial DBN-LSTM

The proposed method employs generator and discriminator components to incorporate adversarial concepts for DoS/DDoS threat identification in a SDN. The model uses PGD to craft adversarial attack generation techniques. The discriminator is built with WGAN to discriminate between real and attack samples. The advantage of PGD is that it allows for more sophisticated and diverse adversarial attack scenarios [39]. PGD ensures the resilience of the proposed work by generating stronger adversarial samples during the training process. The GAN training process is stabilized by WGAN [40]. WGAN helps to mitigate issues like mode collapse and non-convergence, which are common in standard GAN implementations. By improving the Wasserstein distance, WGAN creates a more stable training environment and increases the efficiency and dependability of adversarial samples. By combining PGD and WGAN, the adversarial DBN-LSTM approach is guaranteed to achieve greater resilience against adversarial attacks. This greatly enhances the robustness and detection accuracy of our suggested adversarial DBN+LSTM model in SDN settings. Adversarial samples

generated in this study using PGD attack introduces a bounded perturbations to network traffic features to simulate adaptive DoS/DDoS behavior. During training, these adversarial samples are used together with clean data to improve model robustness. A WGAN–based discriminator is employed to distinguish real and adversarial traffic samples. The model is evaluated on adversarial data before and after adversarial training. The equations used in the proposed method are presented below.

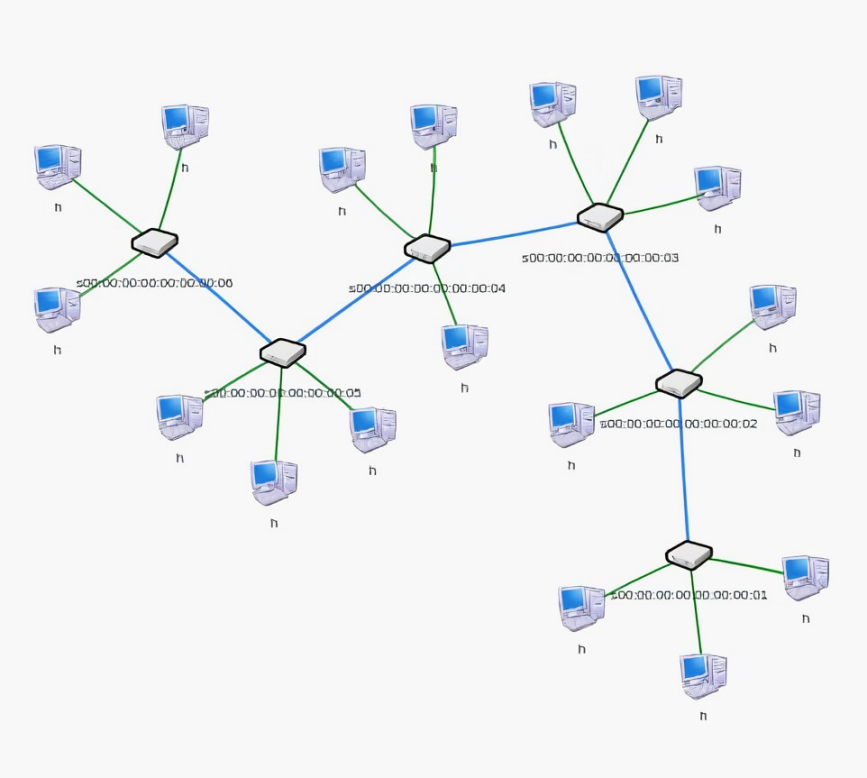

Figure 2. Mininet topology used to generate TCP SYN, ICMP, and UDP flooding traffic

4.2.1. PGD-based adversarial sample generation

The adversarial sample in the $(k+1)$-th round is computed using:

$$x^{(k+1)} = \text{Proj}_{\mathcal{B}_p(x,\epsilon)} \left(x^{(k)} + \alpha \cdot \text{sign} \left(\nabla_x \mathcal{L}(x^{(k)}, y, \theta) \right) \right) \tag{1}$$

where, $x^{(0)} = x$ is the original input sample.
α is the step size for perturbation.
$\mathcal{B}_p(x, \epsilon)$ is the ℓ_p-norm ball of radius ϵ around x.
$\mathcal{L}(x, y, \theta)$ is the models' function to calculate loss.
Proj denotes the projection back into the norm ball.
The final adversarial example after K iterations is $x^{\text{adv}} = x^{(K)}$.

4.2.2. Wasserstein GAN loss functions

The discriminator loss without gradient penalty is represented as:

$$\mathcal{L}_{\text{disc}} = \mathbb{E}_{z \sim Q_{\text{gen}}} [\mathcal{D}(G(z))] - \mathbb{E}_{y \sim Q_{\text{real}}} [\mathcal{D}(y)] \tag{2}$$

On the other side, the generator undergoes training to overcome the discriminator values on the samples that are generated. The loss is given as,

$$\mathcal{L}_{\text{gen}} = -\mathbb{E}_{z \sim Q_{\text{gen}}} [\mathcal{D}(G(z))] \tag{3}$$

The Lipschitz constraint is applied by a gradient penalty term.

$$\mathcal{L}_D^{\text{GP}} = \mathcal{L}_D + \lambda \cdot \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}} \left(\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2 \right) \tag{4}$$

$\hat{x}$ is defined as $\alpha x + (1-\alpha)\tilde{x}$, where $\alpha \sim \mathcal{U}(0,1)$. The generator's objective is defined as:

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \tag{5}$$

In addition to the adversarial model with PGD and WGAN, the proposed work uses LSTM and DBN as seen in Figure 3. This ADBN–LSTM combination is chosen to explicitly separate feature learning from temporal modeling. The DBN captures deep and non-linear relationships among heterogeneous flow and control-plane features in SDN environments. It does not rely on spatial assumptions, which are required by typical CNN-based models. The LSTM then models long-term temporal behaviors such as sustained flooding and burst patterns that characterize DoS/DDoS attacks. In contrast, CNN-LSTM and standalone deep models jointly learn spatial and temporal patterns together. This tight coupling in CNN-LSTM and standalone models reduces robustness; also increasing their sensitivity to noise and adversarial perturbations.

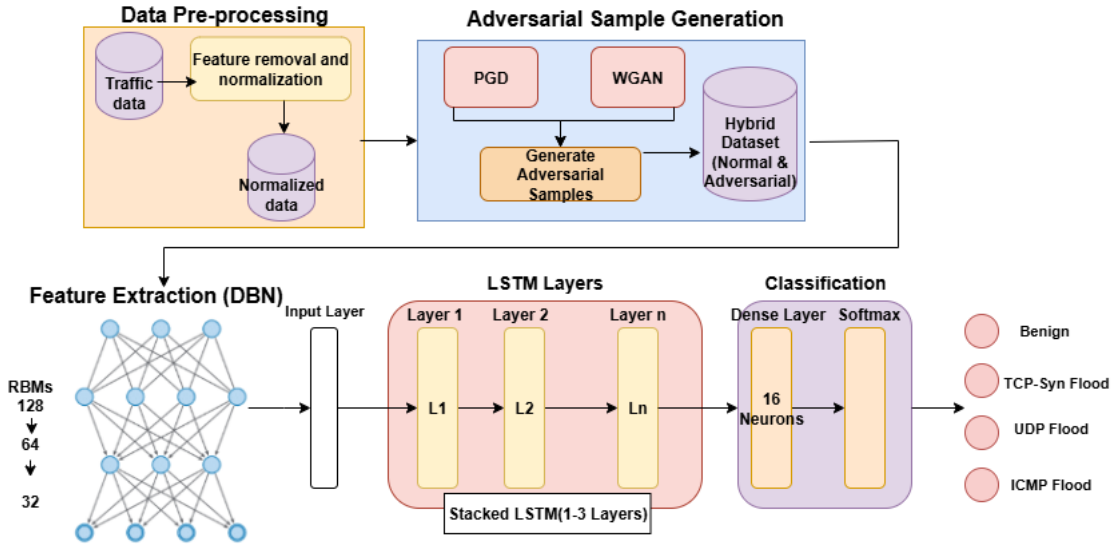

Figure 3. Architectural overview of the adversarial DBN+LSTM Model

The prepared input dataset is fed to DBNs that act as a multi-layer feature extractor to capture high-level, compact representations of the input data. The stacked Bernoulli restricted Boltzmann machines (RBMs) within the DBN are with progressively decreasing dimensions of 128, 64, and 32 units. Each RBM is given training in a greedy, layer-by-layer approach, learning hidden representations of the network traffic features while reducing noise and dimensionality. The input for the next LSTM network is the output of the last RBM layer. The LSTM is responsible for learning both high-level and granular temporal features in network traffic, such as sustained flooding and burst behaviors. The LSTM module is implemented as a stacked architecture. Different LSTM depths are evaluated during experimentation including one-layer (64 units), two-layer (64 $\rightarrow$ 32 units) and three-layer (128$\rightarrow$ 64$\rightarrow$ 32 units).

The LSTM processes fixed-length traffic sequences with a time-step size of T = 10, constructed from consecutive flow-level records. The proposed methodology incorporates advanced adversarial training techniques such as PGD and WGAN, to increase generalization to unknown data and strengthen resilience against adversarial attacks. This combination strengthens the model's capacity to handle adversarial perturbations, it ensures the extracted features remain robust and relevant. The final LSTM output is passed through a dense layer with 16 units to capture the complex feature interactions. Finally a Softmax output layer with 4 classes is used for multiclass classification. The complete process of the methodology is summarized as Algorithm 1.

Algorithm 1 Enhanced Adversarial DBN-LSTM Training with Norm Constraints

Input:
DBN-LSTM model parameters θ
Training dataset $\mathcal{D}$
Perturbation budget ϵ
Norm type p
Learning rate η
Maximum iterations T
Initialize model parameters θ for DBN-LSTM
while $t < T$ and stop condition not met **do**
for each mini-batch $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m+1}$ from $\mathcal{D}$ **do**
Generate adversarial samples:
Compute gradients: $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_i, y_i, \theta)$
Normalize perturbation:

$$\delta_i = \eta \cdot \frac{\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_i, y_i, \theta)}{\|\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_i, y_i, \theta)\|_p}$$

Generate adversarial sample:

$$\mathbf{x}_{i,\text{adv}} = \mathbf{x}_i + \delta_i \quad \text{such that } \|\mathbf{x}_{i,\text{adv}} - \mathbf{x}_i\|_p \leq \epsilon$$

Construct hybrid dataset:

$$\mathcal{D}_{\text{hybrid}} = \{(\mathbf{x}_i, y_i), (\mathbf{x}_{i,\text{adv}}, y_i)\}_{i=1}^{m+1}$$

Update model parameters:
Compute loss on $\mathcal{D}_{\text{hybrid}}$:

$$\mathcal{L}_{\text{m}} = \frac{1}{|\mathcal{D}_{\text{hybrid}}|} \sum_{(\mathbf{x},y) \in \mathcal{D}_{\text{hybrid}}} \mathcal{L}(\mathbf{x}, y, \theta)$$

Update θ using gradient descent:

$$\theta = \theta - \eta \cdot \nabla_\theta \mathcal{L}_{\text{m}}$$

end for
end while
Return: Trained DBN-LSTM parameters θ

5. RESULTS AND DISCUSSIONS

5.1. Dataset description

The dataset is preprocessed before model training to remove all redundant and irrelevant features. All numerical features are normalized using min–max scaling. This step brings all feature values into a common range and improves training stability. The preprocessed dataset is then divided into training, validation, and testing subsets using a fixed split of 70%, 15%, and 15%, respectively. The same data split is applied to all experiments to ensure fair and reproducible evaluation.

5.1.1. Mininet testbed dataset

The dataset utilized in this research work was created from an SDN testbed implemented using Mininet. Network features are identified from the generated dataset for both regular and attack traffic. The attributes for the attack and normal dataset that are generated during the experiment are listed in the Table 1. A variety of network traffic scenarios for the TCP, UDP, and ICMP protocols are included in the dataset that serves as the foundation for training and assessing detection and mitigation algorithms of malicious and legitimate traffics. The data collected are used for the development and evaluation of systems in SDN environments.

Table 1. Attributes in attack and normal traffic datasets generated in mininet testbed

Traffic type	Attributes
Attack dataset	Frame Features (frame.encap_type, frame.len, frame.protocols), IP Header Fields (ip.hdr_len, ip.flags.rb, ip.len, ip.flags.mf, ip.flags.df, ip.frag_offset, ip.src, ip.ttl, ip.proto, ip.dst), TCP Attributes (tcp.srcport, tcp.dstport, tcp.len, tcp.ack, tcp.flags.syn, tcp.flags.ack, tcp.window_size,), ICMP/UDP Fields (icmp.seq, icmp. checksum, icmp.id, udp.length), Flow Timing Info(flow_start_time, flow_end_time)
Normal dataset	ICMP Details (icmp_type, icmp _code), OpenFlow/SDN Identifiers (timestamp, flow_id, datapath_id, ip_src and ip_dst, ip_proto, tp_dst tp_src,), Timeout & Flag, Settings(idle_ timeout, flags, hard_timeout), Flow Durations (flow_duration_nsec, flow_duration_sec), Traffic Volume parameters (packet _count, byte_count_per_second, packet_count_per_second, byte_count, packet_count_per_nsecond,), Statistical Features (request_reply_ratio, byte_count_per_nsecond, syn_ack_ratio, packet_size_variance and label)

5.1.2. CICDDoS2019

To tackle the challenges laid out by DoS/DDoS attacks, the CICDDoS2019 dataset [41] was created. It includes a wide variety of DDoS attacks, such as those that are focused on reflection or exploitation attacks. Notably, the dataset encompasses various attack types including SSDP, MSSQL, SNMP, CharGen, NTP, LDAP, TFTP, NetBIOS, DNS, SYN and UDP flood, and UDP-Lag. The dataset use the data from packet capture (PCAP) files. Network traffic analysis information is provided through the use of labeled flows. These flows include informations, for instance ports, protocols, source and destination's IP addresses, timestamps, and categories of attacks. In order to accurately depict the complicated nature of actual network settings, the dataset models the abstract behavior of users across a variety of protocols, like HTTP, SSH, HTTPS, FTP, and email. Table 2 lists the DoS/DDoS attacks taken into consideration when producing the dataset.

Table 2. DoS/DDoS attacks addressed in CICDDoS2019 dataset

Scenario	Attack names	Total count
Attacks carried out during the day of training	NTP, LDAP, DNS, NetBIOS, MSSQL, SSDP, SNMP, UDP-Lag, UDP, TFTP, WebDDoS, and SYN	12
Attacks carried out during the day of testing	PortScan, LDAP, NetBIOS, MSSQL, UDP-Lag, UDP and SYN	7

5.1.3. InSDN

InSDN dataset is proposed by author Elsayed *et al.* [42]. This dataset's objective is to give a complete set of data for examining and creating solutions for security issues in SDN networks. The data generation methodology involves creating attack scenarios specific to SDN environments. These attacks target critical SDN components such as controllers and OF switches. The attack also exploit vulnerabilities present in SDN applications, such as buffer overflow, command injection, and SQL injection. The attack classes addressed in the virtual environment are DoS, web attacks, DDoS, R2L, Probe, Malware, and U2R. In addition several SDN specific attacks are covered, including flow-rule flooding attack, data-to-control plane flooding attack, password-guessing attacks, link-flooding attack (LFA), and remote application exploitation.

5.2. Metrics for performance evaluation

The overall efficacy of the proposed research methodology is assessed using accuracy and F-score. Accuracy gauges the extent to which flows are accurately identified as normal or attack, using the intrusion identification system. In the entire dataset, the count of properly identified data flows with respect to the total amount of data flow is represented as accuracy. Accuracy is given as in (6).

$$Acc = \frac{TrueP_{os} + TrueN_{eg}}{TrueP_{os} + TrueN_{eg} + FalseP_{os} + FalseN_{eg}} \tag{6}$$

The F-score, is a more comprehensive indicator combining precision and recall. Precision measures the exactness of any IDS system, in correctly identifying attacks. It is estimated as fraction of correctly labeled attack flows to all the flows labeled as attacks through IDS. Recall, measures the completeness in identifying all the actual attacks by the system present in the given dataset. Recall is given as a fraction of correctly labeled

attack flows to all the actual attack flows in the dataset, indicating how well the system avoids false negatives. Precision (pre) and also recall are given in (7) and (8) respectively.

$$Pre = \frac{TrueP_{os}}{TrueP_{os} + FalseP_{os}} \tag{7}$$

$$Recall = \frac{TrueP_{os}}{TrueP_{os} + FalseN_{eg}} \tag{8}$$

The F-score, as given in (9), is representing the combined measure of recall and precision, and it offers a equal measure of the proficiency pertaining to the intrusion identification system.

$$F\text{-}Score = 2X\frac{PrecisionXRecall}{Precision + Recall} \tag{9}$$

The higher the F-score, the better the equilibrium among precision as well as recall, which indicates a higher overall efficiency in the model's intrusion detection. It's worth emphasizing that improving one measure, such as precision, at the expense of the other, such as recall, may not necessarily improve the F-score. It is calculated with the values of recall, and precision and the overall system performance should be evaluated with reference to both metrics.

5.3. Experimental analysis

All experiments for implementing the proposed model were conducted on a workstation equipped with an Intel Core i7-9700 CPU (3.0 GHz, 8 cores) and an NVIDIA GeForce RTX 2080 Ti GPU with 11 GB VRAM, enabling efficient training of deep learning models. The system was configured with 32 GB of RAM to support large-scale dataset processing during training and evaluation. The experiments were performed on a Linux-based operating system using Python 3.8.10. The deep learning models were implemented using the TensorFlow framework with GPU acceleration enabled.

The proposed adversarial DBN+LSTM model employs PGD in the generator and WGAN in the discriminator. Integration of PGD and WGAN enhances robustness against adversarial attacks and also improves classification accuracy. PGD-based adversarial training is included to generate perturbed input samples. These perturbed input data ensure the model stays resilient to adversarial manipulations. Meanwhile, WGAN synthesizes attack samples, improving data diversity and enhancing model generalization. The perturbed inputs are then passed to the LSTM network during training to enhance robustness against adversarial attacks.

Table 3 illustrates how the model is tested with different hidden layer configurations, neuron/unit counts, and dropout rates to optimize the performance. For enhancing the non-linearity of the model, ReLU activation function is involved. At last, the Softmax function guarantees multi-class classification. For the whole process, dropout layers are used to add regularization, preserving model generalization while avoiding overfitting. As indicated in the Table 3, a dropout rate of 0.2 provides optimal performance, balancing model complexity and stability. In order to improve resilience against evasion attacks, perturbed input samples are also used in adversarial training.

Table 3. Results of varying Adversarial DBN+LSTM models: Validation Accuracy (%)under different parameters

Parameter	DBN+LSTM (without hidden layers)	DBN+LSTM_1	DBN+LSTM_2	DBN+LSTM_3
Hidden Units				
32	96.45	97.33	98.67	99.10
64	96.89	98.12	99.02	99.25
128	97.34	98.79	99.40	99.40
Dropout				
0.0	97.56	98.64	99.10	99.12
0.1	98.12	99.02	99.25	99.30
0.2	98.56	99.12	99.40	99.40

The adversarially trained models (DBN+LSTM-2 and DBN+LSTM-3) are able to identify complex cyberthreats in SDN systems because of their consistently high accuracy. These findings suggest that a moderate

number of hidden layers, proper dropout adjustment, and adversarial training significantly enhance model performance and dependability for network intrusion detection. The findings show that accuracy increases with the number of hidden units (up to 128). Due to their deeper feature extraction capabilities, DBN+LSTM-2 and DBN+LSTM-3 achieved the greatest validation accuracy of 99.40%. Adversarial DBN+LSTM performance with various parameters is displayed in Table 4.

Table 4. Performance comparison of different adversarial DBN-LSTM architectures

Model type	DBN+LSTM_1	DBN+LSTM_2	DBN+LSTM_3
DBN Layers	1 RBM (128 units)	2 RBMs (128 → 64 units)	3 RBMs (128 → 64 → 32 units)
Learning Rate	0.001	0.0005	0.0001
LSTM Layers	1 LSTM (64 units)	2 LSTMs (64 → 32 units)	3 LSTMs (128 → 64 → 32 units)
Size of each Batch	32	64	128
Dropout Rate	0.1	0.2	0.3
Number of Hidden units	64	128	256
Activation Functions	ReLU, Softmax	ReLU, Softmax	ReLU, Softmax
Validation Accuracy (%)	97.5	98.8	99.4

The suggested adversarial DBN+LSTM model is contrasted with baseline models that are relied on deep learning for DoS/DDoS threat detection. Baseline models for comparison include Autoencoder [43], BRNN [44], BiLSTM [45], DNN [27], and GRU [46], evaluating their effectiveness in detecting network intrusions with DoS/DDoS attacks. The suggested model is found to have the maximum accuracy among the baseline models compared, demonstrating its effectiveness in detecting DoS/DDoS attacks. The adversarial training with PGD and WGAN boost the model's resilience, handling the adversarial perturbations and dynamic attack patterns in SDN environments. The comparison of precision, F1-score, recall, and accuracy among the proposed model and the baseline models is highlighted in Table 5. Table 6 shows relative results of adversarial DBN-LSTM model across various datasets used in this research work. In addition, the accuracy of various models is visualized in Figure 4 to illustrate their performance differences.

Table 5. Model comparison

Model	Accuracy	Recall	Precision	F1-score
Autoencoder	94.52	94.3	94.4	94.35
BRNN	97.34	97.2	97.3	97.25
BiLSTM	98.91	98.8	98.9	98.85
DNN	94.22	94.1	94.2	94.15
GRU	96.05	96.03	96.04	96.03
Adversarial DBN+LSTM	99.4	99.2	99.3	99.3

Table 6. Estimated evaluation metrics for adversarial DBN+LSTM mdel using several datasets

Dataset	Accuracy	Recall	Precision	F1-Score
Mininet testbed Dataset	99.40	99.20	99.30	99.30
InSDN [42]	98.75	98.80	98.60	98.70
CICDDoS2019 [41]	99.45	99.26	99.33	99.35

The classification model's ability to differentiate among the various forms of network data traffic as well as attacks is demonstrated by the confusion matrix shown in Figure 5. The matrix's strong diagonal suggests that the model does well when it comes to accuracy, as most predictions align with the true labels. The model's robustness in distinguishing between attack types is demonstrated by the minimal off-diagonal values, which show that misclassifications are rare. As the darker diagonal draws attention to the concentration of accurate classifications, the color intensity in the matrix further shows the dominance of accurate predictions. Overall, the results indicate that the model is reliable and does well on multiclass classification tasks, especially when network traffic and intrusion detection are included.

The prediction matrix of the adversarial DBN+LSTM model using the CICDDoS2019 dataset is shown in the Figure 6. The expected class is represented by each column, and the actual class is represented by each row. The model's ability to accurately recognize many forms of DDoS attacks such as DrDoS_DNS, UDP_lag, DrDoS_MSSQL, DrDoS_LDAP, DrDoS_NTP, DrDoS_NetBIOS, DrDoS_SNMP, DrDoS_SSDP, DrDoS_UDP,

SYN Flood, TFTP Flood are demonstrated by the much higher diagonal values along with benign traffic. The off-diagonal values, which are relatively small, indicate minor misclassifications between similar attack types, likely due to overlapping traffic patterns or feature similarities. Flow-level statistical features provided by the CICDDoS2019 dataset are used as input to the model, including packet- and byte-based statistics, flow duration, inter-arrival time features, and protocol-related attributes. The similar prediction matrix for the InSDN dataset is given in Figure 7.

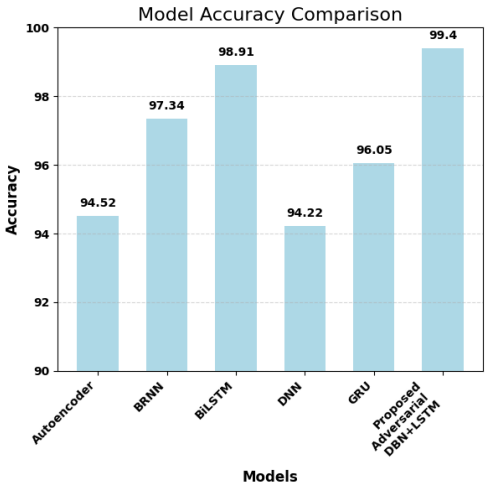

Figure 4. Accuracy comparisons of various models

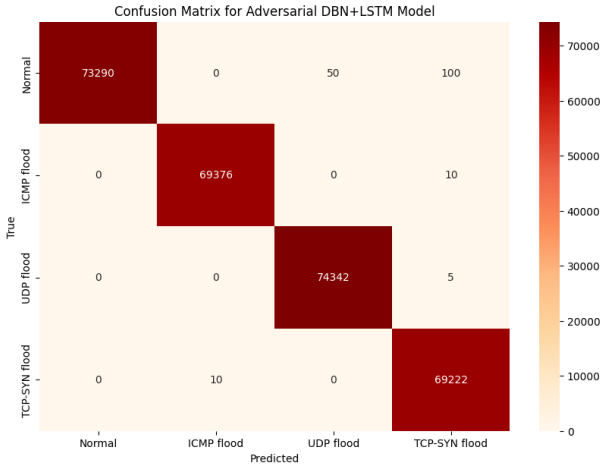

Figure 5. Confusion matrix of adversarial DBN+LSTM with the dataset generated by Mininet

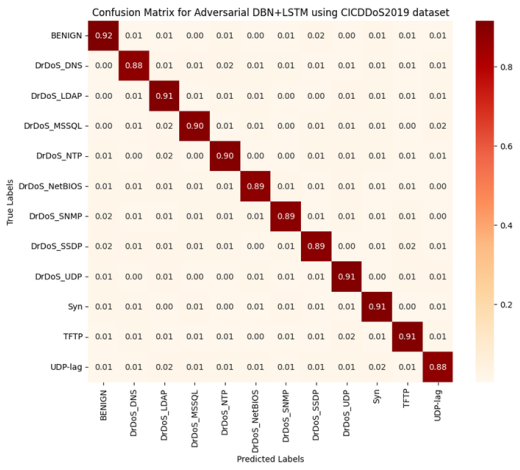

Figure 6. Confusion matrix of adversarial DBN+LSTM with CICDDoS2019 [41] dataset

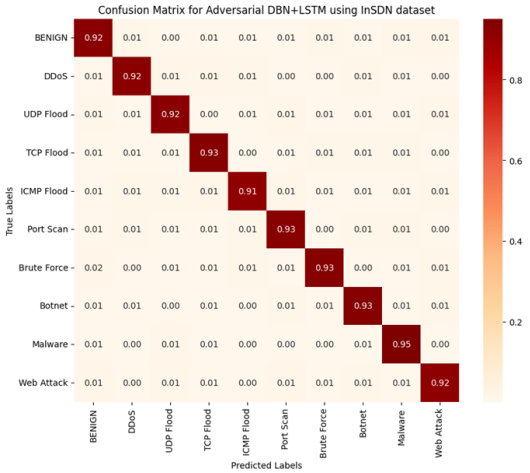

Figure 7. Confusion matrix of adversarial DBN+LSTM with InSDN [42] dataset

The training and validation error of the adversarial DBN + LSTM architecture, along with the baseline models, is showed in the Figure 8. The diagram highlights the convergence behavior and training efficacy of the formulated adversarial DBN+LSTM model in contrast to the other baseline models. The values demonstrate that the suggested model attains better and more stable convergence, indicating improved learning efficiency and robustness against adversarial perturbations.

Table 7 presents an examination of different deep learning models for multi-class classification of denial of services attack types in an SDN environment. The models considered as baseline model for comparing performance are Autoencoder, BRNN, BiLSTM, DNN, GRU, and the Proposed Adversarial DBN+LSTM.

The observed performance matrices (F1-score, accuracy, recall, and precision) vary due to multiple factors. The variations in output is because of feature extraction capability, sequence learning efficiency, and model robustness in handling class imbalances and adversarial characteristics of network traffic data.

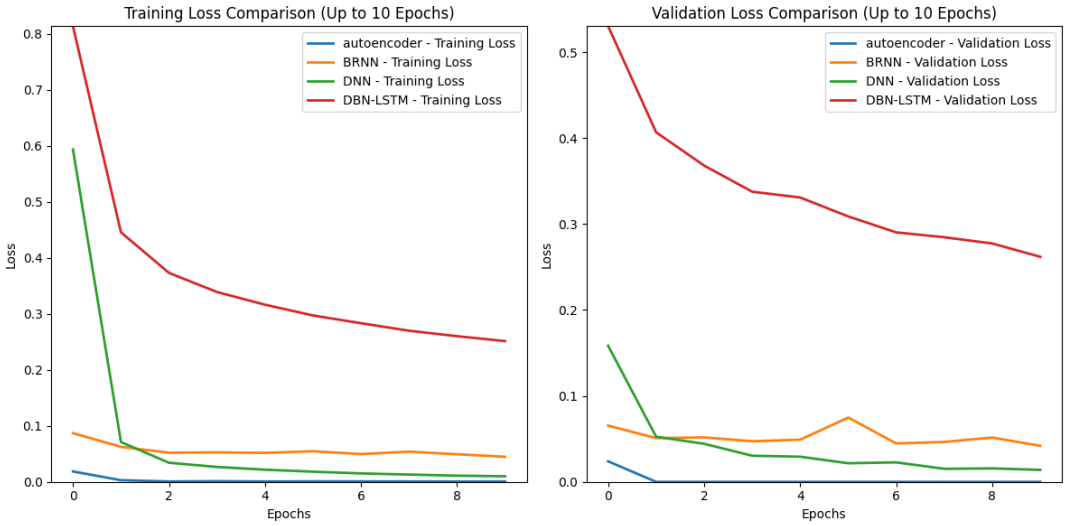

Figure 8. Comparing the loss of training and validation of adversarial DBN+LSTM with other baseline models

Table 7. Multi-class classification results on Mininet generated dataset

Model	Metric (%)	Normal	ICMP Flood	UDP Flood
Autoencoder	Accuracy	94.52	94.52	94.52
	Recall	94.3	94.3	94.3
	Precision	94.4	94.4	94.4
	F1-Score	94.35	94.35	94.35
BRNN	Accuracy	97.34	97.34	97.34
	Recall	97.2	97.2	97.2
	Precision	97.3	97.3	97.3
	F1-Score	97.25	97.25	97.25
BiLSTM	Accuracy	98.91	98.91	98.91
	Recall	98.8	98.8	98.8
	Precision	98.9	98.9	98.9
	F1-Score	98.85	98.85	98.85
DNN	Accuracy	94.22	94.22	94.22
	Recall	94.1	94.1	94.1
	Precision	94.2	94.2	94.2
	F1-Score	94.15	94.15	94.15
GRU	Accuracy	96.05	96.05	96.05
	Recall	96.03	96.03	96.03
	Precision	96.04	96.04	96.04
	F1-Score	96.03	96.03	96.03
Proposed Adversarial DBN+LSTM	Accuracy	**99.4**	**99.4**	**99.4**
	Recall	**99.2**	**99.2**	**99.2**
	Precision	**99.3**	**99.3**	**99.3**
	F1-Score	**99.3**	**99.3**	**99.3**

The combination of DBN and LSTM provides superior feature extraction and temporal pattern recognition. DBNs capture abstract high-level feature informations from network flow, allowing better representation of normal and attack patterns. Since DoS/DDoS attacks exhibit temporal patterns, LSTM effectively models long-term temporal dependencies, Differentiating between harmful and legitimate traffic with high precision. The introduction of adversarial training enhances robustness against sophisticated attacks, reducing misclassification

and improving generalization. Higher recall value (99.2%) indicate that most attacks are correctly detected, the high precision value (99.3%) indicates a low false positive rate, leading to an optimal F1-score.

The Proposed Adversarial DBN+LSTM model outperforms existing deep learning models in DoS/DDoS attack identification on the CICDDoS2019 dataset, achieving a 99.45% accuracy. This is higher than traditional CNN-LSTM (98.92%) [28], LSTM (98%) [29], and CNN+GRU (98.36%) [47] models as shown in Table 8. The Adversarial DBN+LSTM model improves robustness with adversarial training, enhancing resilience against adversarial attacks. Additionally, models implemented in SDN environments like CNN-LSTM (95.03%) [33] and CNN-BiLSTM (97.77%) [33] demonstrate lower accuracy due to the lack of adversarial robustness and feature extraction limitations. The notable benefit of the DBN+LSTM hybrid model is its proficiency to learn deep hierarchical features using DBN's stacked Restricted Boltzmann Machines (RBMs) while effectively modeling temporal dependencies in network traffic using LSTM. In addition, the adversarial training strategy ensures that the model can mitigate evasion attacks, making it perfect for SDN-based settings' real-time intrusion detection. The improved precision, F1-score, and recall values highlight its superior detection performance, indicating reduced false positives.

Table 8. Comparison results of adversarial DBN-LSTM models with state-of-the-art models

Articles	DL method	Dataset	Accuracy	Recall	Precision	F1-Score
Rajan *et al.* [28]	CNN-LSTM	CICIDS-2017	98.92	99.67	97.82	98.74
Kumar *et al.* [29]	LSTM	CICDDoS2019	98.00	97.00	97.00	98.00
Gebremedhin *et al.* [31]	LSTM + Feature Selection	CICDDoS2019	99.40	99.46	99.40	99.42
Said *et al.* [33]	CNN-LSTM	InSDN	95.03	96.11	93.59	94.60
Said *et al.* [33]	CNN-BiLSTM	InSDN	97.77	95.28	99.85	97.51
Whitworth *et al.* [47]	CNN+GRU	CICDDoS2019	98.36	95.45	93.09	94.56
Proposed Work	Adversarial DBN+LSTM	CICDDoS2019	99.45	99.26	99.33	99.35

6. CONCLUSION

DoS/DDoS cyber attacks are a serious risks for SDN networks. Even the attacker tries to cheat the attack prediction models using adversarial training. The current research work focuses on identifying and mitigating the DoS/DDoS threats in adversarial conditions. Also, the traditional models to identify DDoS attacks, are unable to identify changing DDoS patterns and has significant false positive rates. The proposed adversarial DBN-LSTM methodology integrates PGD and WGAN to generate robust adversarial samples, enhancing the model's resilience against sophisticated attacks in SDN environments. By leveraging DBN for dimensionality reduction and LSTM for temporal feature extraction, the model effectively captures complex attack traffic distribution profiles, improving detection accuracy. The combination of adversarial training and deep learning ensures a highly adaptive and robust intrusion detection system capable of mitigating evolving cyber threats. The proposed work gives an accuracy value of 99.4% with the mininet generated dataset. The results of adversarial DBN-LSTM model is compared with other baseline models, such as, Autoencoder, BRNN, BiLSTM, DNN and GRU. While WGAN-generated synthetic data aids in addressing class imbalance, it might not accurately represent the intricacy and unpredictability of actual attacks, which could result in overfitting to particular attack patterns. While WGAN-generated synthetic traffic helped reduce class imbalance, the study also showed that these samples do not always capture the full unpredictability of real attack behavior, which may lead to overfitting on certain patterns. Moreover, although WGAN is more stable than traditional GANs, its training overhead remains high and is not ideal for real-time SDN environments with limited computing resources. To address these limitations, future work will focus on developing a lightweight and resource-optimized WGAN architecture that can operate efficiently in real-time SDN settings. Also, integrating adaptive learning mechanisms will allow the model to continually update itself using live SDN traffic, ensuring better generalization to evolving DoS/DDoS behaviors.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to Vishnu Raghavendra and Vishwajith U. K. from the Department of Computer Science and Engineering (IoT & Cybersecurity including Blockchain), BMS College of Engineering (BMSCE), Bengaluru, India, for their valuable assistance with minor coding implementations during the early development of this work. Although their contributions do not meet the criteria for authorship, their support significantly aided the smooth execution of the experimental setup.

FUNDING INFORMATION

The authors declare that no funding was received to support this research.

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
Manjula Maraiah	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Venkatesh		✓			✓	✓		✓	✓	✓	✓	✓	✓	

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

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, MM, upon reasonable request.

REFERENCES

[1] N. McKeown, "Software-Defined Networking," *INFOCOM Keynote Talk*, vol. 17, no. 2, pp. 30–32, 2009.
[2] H. Wang and Y. Li, "Overview of DDoS Attack Detection in Software-Defined Networks," *IEEE Access*, vol. 12, pp. 38351–38381, 2024.
[3] A. Kaur, C. R. Krishna, and N. V. Patil, "A comprehensive review on Software-Defined Networking (SDN) and DDoS attacks: Ecosystem, taxonomy, traffic engineering, challenges and research directions," *Computer Science Review*, vol. 55, p. 100692, 2025.
[4] A. Hirsi, M. A. Alhartomi, L. Audah, A. Salh, N. M. Sahar, S. Ahmed, G. O. Ansa, and A. Farah, "Comprehensive Analysis of DDoS Anomaly Detection in Software-Defined Networks," *IEEE Access*, vol. 13, pp. 23013–23071, 2025.
[5] A. A. Habib, A. Imtiaz, D. Tripura, M. O. Faruk, M. A. Hossain, I. Ara, S. Sarker, and A. Z. Abadin, "Distributed Denial-of-Service Attack Detection Short Review: Issues, Challenges, and Recommendations," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 1, pp. 438–446, 2025.
[6] W. Hill, Y. T. Acquaah, J. Mason, D. Limbrick, S. Teixeira-Poit, C. Coates, and K. Roy, "DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies," *Discover Applied Sciences*, vol. 6, no. 9, p. 472, 2024.
[7] T. Jafarian, A. Ghaffari, A. Seyfollahi, and B. Arasteh, "Detecting and mitigating security anomalies in Software-Defined Networking (SDN) using Gradient-Boosted Trees and Floodlight Controller characteristics," *Computer Standards & Interfaces*, vol. 91, p. 103871, 2025.
[8] M. A. Setitra, M. Fan, I. Benkhaddra, and Z. E. A. Bensalem, "DoS/DDoS attacks in Software Defined Networks: Current situation, challenges and future directions," *Computer Communications*, vol. 222, pp. 77–96, 2024.
[9] M. AbdulRaheem, I. D. Oladipo, A. L. Imoize, J. B. Awotunde, C.-C. Lee, G. B. Balogun, and J. O. Adeoti, "Machine Learning Assisted Snort and Zeek in Detecting DDoS Attacks in Software-Defined Networking," *International Journal of Information Technology*, vol. 16, no. 3, pp. 1627–1643, 2024.
[10] R. Uddin, S. A. Kumar, and V. Chamola, "Denial of service attacks in edge computing layers: Taxonomy, vulnerabilities, threats and solutions," *Ad Hoc Networks*, vol. 152, p. 103322, 2024.
[11] D. M. A. A. Afraji, J. Lloret, and L. Peñalver, "Deep Learning-Driven Defense Strategies for Mitigating DDoS Attacks in Cloud Computing Environments," *Cyber Security and Applications*, p. 100085, 2025.
[12] S. Javanmardi, M. Ghahramani, M. Shojafar, M. Alazab, and A. M. Caruso, "M-RL: A mobility and impersonation-aware IDS for DDoS UDP flooding attacks in IoT-Fog networks," *Computers & Security*, vol. 140, p. 103778, 2024.
[13] A. El Ksimi, C. Leghris, S. Lafraxo, and V. K. Verma, "ICMPv6-based DDoS Flooding-Attack Detection Using Machine and Deep Learning Techniques," *IETE Journal of Research*, vol. 70, no. 4, pp. 3753–3762, 2024.
[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, 2024.
[15] N. Pandey and P. K. Mishra, "Devising a Hybrid Approach for Near Real-time DDoS Detection in IoT," *Computers and Electrical Engineering*, vol. 118, p. 109448, 2024.

[16] P. Cheng, C. Zhang, W. Xie, W. Zhang, and S. He, "Network-Based Adaptive Multievent-Triggered Fuzzy Dynamic Positioning Controller Design for Unmanned Surface Vehicles Against Denial-of-Service Attacks," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 2, pp. 612–624, 2023.
[17] R. Bukhowah, A. Aljughaiman, and M. H. Rahman, "Detection of DoS Attacks for IoT in Information-Centric Networks Using Machine Learning: Opportunities, Challenges, and Future Research Directions," *Electronics*, vol. 13, no. 6, p. 1031, 2024.
[18] N. Aslam, S. Srivastava, and M. Gore, "A Comprehensive Analysis of Machine Learning-and Deep Learning-based Solutions for DDoS Attack Detection in SDN," *Arabian Journal for Science and Engineering*, vol. 49, no. 3, pp. 3533–3573, 2024.
[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
[20] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random Forest," in *International Conference on Information Computing and Applications*, Springer, 2012, pp. 246–252.
[21] A. Christmann and I. Steinwart, "Support Vector Machines," 2008.
[22] E. Bisong and E. Bisong, "Logistic Regression," *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 243–250, 2019.
[23] Z. Zhang, "Introduction to Machine Learning: K-Nearest Neighbors," *Annals of Translational Medicine*, vol. 4, no. 11, p. 218, 2016.
[24] F.-J. Yang, "An implementation of Naive Bayes Classifier," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2018, pp. 301–306.
[25] M. J. Santos-Neto, J. L. Bordim, E. A. Alchieri, and E. Ishikawa, "DDoS Attack Detection in SDN: Enhancing Entropy-based Detection with Machine Learning," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 11, p. e8021, 2024.
[26] T. E. Ali, Y.-W. Chong, and S. Manickam, "Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review," *Applied Sciences*, vol. 13, no. 5, 2023.
[27] V. Hnamte, A. A. Najar, H. Nhung-Nguyen, J. Hussain, and M. N. Sugali, "DDoS attack detection and mitigation using deep neural network in SDN environment," *Computers & Security*, vol. 138, p. 103661, 2024.
[28] D. M. Rajan and D. J. Aravindhar, "Detection and Mitigation of DDOS Attack in SDN Environment Using Hybrid CNN-LSTM," *Migration Letters*, vol. 20, no. S13, pp. 407–419, 2023.
[29] D. Kumar, R. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "DDoS Detection using Deep Learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023.
[30] R. Priyadarshini and R. K. Barik, "A Deep Learning Based Intelligent Framework to Mitigate DDoS Attack in Fog Environment," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 825–831, 2022.
[31] T. G. Gebremeskel, K. A. Gemeda, T. G. Krishna, and P. J. Ramulu, "DDoS Attack Detection and Classification Using Hybrid Model for Multicontroller SDN," *Wireless Communications and Mobile Computing*, vol. 2023, 2022.
[32] M. S. E. 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.
[33] R. Ben Said, Z. Sabir, and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," *IEEE Access*, vol. 11, pp. 138732–138747, 2023.
[34] L. Chen, Z. Wang, R. Huo, and T. Huang, "An Adversarial DBN-LSTM Method for Detecting and Defending against DDoS Attacks in SDN Environments," *Algorithms*, vol. 16, no. 4, 2023.
[35] W. Lim, K. S. C. Yong, B. T. Lau, and C. C. L. Tan, "Future of generative adversarial networks (GAN) for anomaly detection in network security: A review," *Computers & Security*, vol. 139, p. 103733, 2024.
[36] A. M. Zacaron, D. M. B. Lent, V. G. da Silva Ruffo, L. F. Carvalho, and M. L. Proença Jr., "Generative adversarial network models for anomaly detection in software-defined networks," *Journal of Network and Systems Management*, vol. 32, no. 4, p. 93, 2024.
[37] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2330–2345, 2023.
[38] W. Yao, H. Shi, and H. Zhao, "Scalable Anomaly-based Intrusion Detection for Secure Internet of Things using Generative Adversarial Networks in Fog Environment," *Journal of Network and Computer Applications*, vol. 214, p. 103622, 2023.
[39] W. Villegas-Ch, A. Jaramillo-Alcázar, and S. Luján-Mora, "Evaluating the Robustness of Deep Learning Models against Adversarial Attacks: An Analysis with FGSM, PGD and CW," *Big Data and Cognitive Computing*, vol. 8, no. 1, p. 8, 2024.
[40] M. Arafah, I. Phillips, A. Adnane, W. Hadi, M. Alauthman, and A.-K. Al-Banna, "Anomaly-based network intrusion detection using denoising autoencoder and Wasserstein GAN synthetic attacks," *Applied Soft Computing*, vol. 168, p. 112455, 2025.
[41] "University of New Brunswick Est.1785," [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html. Accessed: Feb. 26, 2024.
[42] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A Novel SDN Intrusion Dataset," *IEEE Access*, vol. 8, pp. 165263–165284, 2020.
[43] S. Chen and W. Guo, "Auto-Encoders in Deep Learning—a review with new perspectives," *Mathematics*, vol. 11, no. 8, p. 1777, 2023.
[44] T. F. De Barrena, J. L. Ferrando, A. García, X. Badiola, M. S. de Buruaga, and J. Vicente, "Tool remaining useful life prediction using bidirectional recurrent neural networks (BRNN)," *The International Journal of Advanced Manufacturing Technology*, vol. 125, no. 9, pp. 4027–4045, 2023.
[45] O. Pandithurai, C. Venkataiah, S. Tiwari, and N. Ramanjaneyulu, "DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment," *Expert Systems with Applications*, vol. 241, p. 122544, 2024.
[46] D. Kilichev, D. Turimov, and W. Kim, "Next–Generation Intrusion Detection for IoT EVCS: Integrating CNN, LSTM, and GRU Models," *Mathematics*, vol. 12, no. 4, p. 571, 2024.
[47] H. Whitworth, S. Al-Rubaye, A. Tsourdos, and J. Jiggins, "5G Aviation Networks Using Novel AI Approach for DDoS Detection," *IEEE Access*, vol. 11, pp. 77518–77542, 2023.

BIOGRAPHIES OF AUTHORS

Manjula Maraiah received bachelor of engineering, Masters of Technology degrees in Computer Science and Engineering from Visvesvaraya Technological University (VTU), India in 2004 and 2011 respectively. She is pursuing the Ph.D. degree in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering (IoT and Cybersecruity including Blockchain), B.M.S. College of Engineering, Bengaluru, Karnataka, India. Her current research interests include computer networking, cybersecurity, data science. As a dedicated and emerging researcher, she holds a Student Member Account with the Institute of Electrical and Electronics Engineers (IEEE) and is a proud member of the International Society for Technology in Education (ISTE). She can be contacted at email: manjulam.csi@bmsce.ac.in.

Venkatesh is currently working as an associate professor in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bengaluru-560001. Obtained his Bachelor of Engineering, Masters of Technology, and Ph.D. in Computer Science and Engineering in 2000, 2004, and 2018 respectively. His research interests include wireless sensor networks, Ad-hoc networks, data Science, and web recommendation system, cybersecurity, SIoT. He can be contacted at email: venkateshm.uvce@bub.ernet.in.