

# DDoS attack detection using optimal scrutiny boosted graph convolutional and bidirectional long short-term memory

Huda Mohammed Ibadi, Asghar Asgharian Sardroud

Department of Computer Engineering, Urmia University, Urmia, Iran

## Article Info

### Article history:

Received Mar 3, 2025

Revised Jun 18, 2025

Accepted Aug 1, 2025

### Keywords:

Artificial intelligence

Deep learning

Distributed denial of service

Machine learning

Unknown attack

## ABSTRACT

The distributed denial of service (DDoS) attack occurs when massive traffic from numerous computers is directed to a server or network, causing crashes and disrupting functionality. Such attacks often shut down websites or applications temporarily and remain among the most critical cybersecurity challenges. Detecting DDoS is difficult and must occur before mitigation. Recently, machine learning and deep learning (ML/DL) have been employed for detection; however, architectural limitations restrict their effectiveness against evolving attack methods. This paper presents a novel framework, scrutiny boosted graph convolutional–bidirectional long short-term memory and vision transformer (SBGC-BiLSTM-ViT), which integrates graph convolutional, BiLSTM, and ViT models with machine learning classifiers such as support vector machine (SVM), Naïve Bayes (NB), random forest (RF), and K-nearest neighbors (KNN). The integration enables autonomous extraction of critical features, enhancing precision in detecting and classifying DDoS attacks. To further boost performance, a Bayesian optimization algorithm (BOA) is applied for hyperparameter tuning of SBGC and ML methods. Evaluation on benchmark datasets UNSW-NB15 and CICDDoS2019 demonstrates that the proposed approach achieves higher accuracy and effectively identifies new DDoS variants, outperforming conventional methods.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Asghar Asgharian Sardroud

Department of Computer Engineering, Urmia University

Urmia 5756151818, Iran

Email: a.asgharian@urmia.ac.ir

## 1. INTRODUCTION

The harmonious nature of future based on technology with the on-going technologies like artificial intelligence, joined devices and wide-open information development unleashes a new imagery of possibility. The services and operations provided by these connected devices continue to reshape all facets of society, including people's daily lives, healthcare, transportation, and homes. Internet of things (IoT) devices are projected to attain 38.6 billion by 2025 and expand to 50 billion by 2030. Quick joining devices shoot up and new models are sometimes left without updates, security and patches. The widespread adoption of such technologies has created opportunities for malicious actors to exploit vulnerabilities, potentially gaining control over these devices to launch attacks on critical infrastructure and websites [1]. In recent years, cyber-attacks have emerged as a major issue due to their heightened sophistication, including denial-of-service (DoS) methods and the escalating threat of zero-day attacks targeting industrial networks, government entities, and military operations. Conventional signature-based Diagnosis strategies may not be sufficient to tackle the variability in cyber-attacks seen nowadays. Zhao *et al.* [2] new anomaly diagnosis methods can be improved for better learning, adapting and also diagnosing threats on different network areas.

In a new simplified learning-knowledge discovery and data mining (NSL-KDD) version, examples of network layer attacks are divided into 4 basic classes in provide data, which is referred to as: i) denial of service (DoS), when the attacker tries to gain access to the hosts/service locked by the legitimate user; ii) probe, the attacker attempts to obtain information about the target network by performing scanning programs that install or network scans; iii) user to root (U2R), the attacker has leverage some of the most widely used attack strategies such as malware infection and stolen credentials to shift her access from low privilege to super/root user level access; and iv) remote to local (R2L), the attacker can perform a simulation of local users and acquire access to the target system [3]. The DoS attacks are the specific attack level that is common in over-the-net communicating between different nets that employ internet services for utility, storage and process. These attacks are characterized by network congestion caused by “zombie packets”, These are malicious data units that traverse the communication layers, generating significant disruption by overwhelming network channels and contaminating substantial volumes of legitimate traffic during transmission [4].

These attacks share similarities in their execution mechanisms, with variations primarily observed in the scale and intensity of the assault. In a DoS attack a single system and Internet connection attacks a victim. In contrast, distributed denial of service (DDoS) attack leverages more computers as well as internet connections to saturate the victim; it is often done by using botnets, which are networks of compromised machines [5]. Depending on which protocol is involved in an attack there are a range of different ways to execute any of such attacks. As identified by Mahjabin *et al.* [6] such ways as user datagram protocol (UDP) flood assault, transmission control protocol synchronize sequence numbers (TCP SYN), and hypertext transfer protocol (HTTP) flood. HTTP flood attacks feature stimulating requirements of GET/POST supplied across HTTP. When the user wants to retrieve data from a server, GET is used, and when the user wants to send data to a server, such as uploading a file, POST is used.

Transferring thousands of requests to a server or cluster will greatly increase its workload, potentially overwhelming the server and causing disruptions. This renders the servers unreachable to authorized users. By sending a SYN packet, getting a synchronize-acknowledgment (SYN-ACK) packet from the server, and completing the handshake with an ACK packet, TCP SYN attack exploits the three-way handshake procedure in a TCP connection. This kind of attack fakes SYN packet destination address. This results to entries persisting in the connection database of the server and SYN-ACK packets being delivered continually at the spoof address. As the items pile up, the server is unable to process legitimate requests anymore. UDP flood attack: A UDP flood attack includes sending an overwhelming amount of UDP packets with phony internet protocol (IP) addresses and arbitrary port numbers. When these packets reach the server, it checks for programs associated with the specified ports. If nothing matches, a “destination unreachable” packet is returned by the server as a response. Since more packets arrived than the server could handle, the server could not assist authorized users because of too much traffic. Due to the weak security capabilities of IoT devices, significant research has emerged in identifying and mitigating DoS and DDoS traffic. But machine learning (ML) methods are not used whole algorithms.

Among the ML-based intrusion detection systems (IDSs), supervised/semi-supervised learning techniques have become the mainstream ones in real world. However, the classical ML techniques failed to do early detection of emerging unknown attacks, hence unknown attacks diagnosis is still a challenging problem. Novel disease identification differs to be a challenging sign and has caught the eye of researchers to handle plenty of challenge to formulate various approaches from various sectors together with the IDS and face detection. In such fields, the data quickly toggles. Daily, new unfamiliar attacks and innovative familiar attack variants are emerging. In order to be able to continue to diagnose bad traffic, IDS must be able to adapt to switching areas.

The structure of the present paper is organized as follows: section 2 outlines the explanatory context and provides a summary of the related study. Section 3 discusses the conceptual foundations of the proposed schemes and presents an overview of the model employed for empirical evaluation, including implementation details. Section 4 presents the empirical results obtained from the testing procedures, and compares them to previous paper outcomes. Finally, section 5 provides the compact conclusion of the paper, and maps the unit for trajectories of insight paper.

## 2. RELATED WORK

The increasing complexity and frequency of DDoS attacks have prompted an extensive body of research into detection mechanisms, particularly within the context of software defined networking (SDN). While numerous approaches have emerged over the past five years, most studies fall into two primary categories: those employing traditional ML techniques and those leveraging advanced deep learning (DL) models. The purpose of this section is not just to list previous works, but to point out methodological deficiencies and performance limitations in these approaches, that motivates the design of a hybrid, optimized detection paradigm.

Section 2.1 presents the related works that use ML classifier like support vector machine (SVM), K-nearest neighbors (KNN) and decision tree (DT), to detect the anomalies in the traffic of SDN. However, in general, these methods may substantially under-generalize to new or changing threats as they may have (in the worst case) no capacity to effectively learn features. Subsection 2.2 further extends to more recent advances based on DL models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memorys (LSTMs) for enhanced adaptivity and feature extraction. However, they often suffer from computational inefficiency and are not interpretable enough for high-stakes applications.

From this overview, we come to two fundamental observations: i) traditional ML methods do not have deep enough cutting for temporal or semantic understanding and ii) existing DL methods encounter difficulty in terms of optimization or interpretability, and especially in online applications. To bridge this divide, the present study proposes a hybrid detection model that integrates scrutiny boosted graph convolution (SBGC), bidirectional LSTM (BiLSTM), and vision transformer (ViT) models—optimized using the bayesian optimization algorithm (BOA). By combining the representational strength of DL with the decision efficiency of ML, this study aims to develop a robust and adaptable IDS suitable for modern SDN-driven infrastructures.

## 2.1. Traditional machine learning approaches for DDoS detection in SDN environments

Sekar *et al.* [7] which focuses on the DDoS diagnosis and assault prevention domain in SDN reality. The specialized topic of technology in the research comprises machine learning approaches, specifically decision tree and techniques like extremely randomized trees (Extra Tree), and categorical boosting (Cat Boost) in domains of SDN to enhance network security. The primary challenge under review in this work, entrenched in SDN to DDoS attack vulnerability, promotes more robust detection and mitigation solutions creation. Despite the implementation of conventional security measures, their adequacy in addressing the increasingly complex and dynamic nature of DDoS attacks within SDNs remains uncertain.

Applying machine learning techniques, Sanapala *et al.* [8] proposes a unique technique of DDoS attacks detection and mitigation in the SDN architecture. The proposed system integrates dynamic processing and evaluation of real-world network traffic using SVM and DT classifiers. This approach achieves a high accuracy rate in identifying and mitigating potential DDoS attacks. It uses the machine learning approach to detect and mitigate DDoS assaults in SDNs in terms of network security. Tahirou *et al.* [9], who study the use of machine learning techniques. This research emphasizes the security challenges of SDN framework created by the separation of control and data plane. This paper's challenge indicates that well-targeted, accurate diagnosis and improved DDoS attack mitigation of SDN can be tough to address with essentiality being stress for high accuracy and low false positive rates. The proposed method now includes analysis based on traffic pattern, entropy, and ML based approaches. The classifiers used for constructing a classification model for the proposed method comprises KNN, SVM, and Naive Bayes. Alsudani and Saeaa [10] believe that ML-based solutions might be employed to research the topic of DDoS attacks detection in SDN architecture. The existing literature in this paper one of the security challenges of internet technology is discussed as DDoS assault which addresses the DDoS attack on SDN architecture or resource allocation is comparably less and covers key implications for future SDN technology. The purpose of the current research is to undertake a comparative performance evaluation to discover the best-performing machine-learning approaches for DDoS attack detection. Machine learning methods such as RF, logistic regression (LR), DT, KNN, and SVM are used to identify and detect DDoS attacks. The purpose is to determine the best technique to detect DDoS assaults in SDN systems. Feng *et al.* [11] focus on incorporating ML particularly for DDoS attack detection.

The centralized control plane of SDN makes it inherently vulnerable to DDoS attacks, which can render the entire network infrastructure inoperative. To enhance DDoS detection accuracy and reduce false positives and detection time, several ML-based solutions have been proposed. One study introduced a hybrid SVM-KNN model that processes packet header data for training and classification. Alashhab *et al.* [12] proposed an online ML-based LDDoS detection approach using explainable boosting machine (EBM) with stochastic gradient descent (SGD), outperforming traditional methods like SVM, NB, and DT. Almohagri *et al.* [13] explored SDN vulnerabilities under DDoS threats and emphasized the potential of ML techniques in attack diagnosis. Another study assessed the efficiency of DT, extreme gradient boosting (XGBoost), and random forest (RF) in SDN-based DDoS detection. Berei *et al.* [14] applied a dual-feature-reduction method to train nine models on SDN traffic and validated their performance using PyShark and CICFlowMeter for real-time analysis. Das *et al.* [15] introduced a hybrid ensemble ML framework combining supervised (for known attacks) and unsupervised (for zero-day attacks) classifiers using outlier detection. Almomani *et al.* [16] benchmarked several ML classifiers, including XGBoost, Gaussian NB, DT, RF, SVM, and LR, for DoS detection in IoT environments. Airlangga [17] conducted a comparative analysis of RF, DT, and LR models with synthetic minority over-sampling technique (SMOTE)-enhanced data to counter class imbalance. Nisa *et al.* [18] proposed a two-phase authentication method based on ML-driven packet filtering to improve SDN security without disrupting host operations. Table 1 summarizes the evaluated models, datasets, and their comparative advantages and limitations.

Table 1. Summary of ML approaches for DDoS detection

Ref	Year	Classifier	Dataset	Pros	Cons
[7]	2023	DT, Cat Boost, Extra Tree	Kaggle	Enhanced security	Dataset limitations
[8]	2023	SVM, DT	Network traffic data	Scalable, effective, real-time detection	Feature selection, potential data overfitting
[9]	2023	KNN, SVM, Gaussian Naïve Bayes (GNB)	CIC-2019, Mininet emulator	Effective DDoS detection and mitigation	Need for large training data
[10]	2023	SVM, RF, DT, LR, KNN	Mendeley data	Centralized control, efficient detection	Specific domain applicability
[11]	2023	SVM, KNN	Mininet emulator	Higher detection performance	Potential scalability challenges
[12]	2023	Online ML with SGD and EBM	Mininet emulator	Realtime time processing, interpretability	Simulated environment, limited evaluation
[13]	2023	XGBoost, RF, DT	CICDDoS2019	Enhanced network security	Limited real world
[14]	2024	RF, KNN, and SVM	CICFlowMeter and Pyshark	Improving overfitting, reducing the unnecessary complexity of the initial models, and decreasing both training and prediction times	Dataset limitations
[15]	2024	LR, DT, NB, NN, SVM, local outlier factor (LOF), isolation forest (ISOF), ensemble learning (ELE), OCSP	NSL-KDD, UNSW-NB15, and CICIDS2017	Correctly detecting DDoS attacks without many false alarms	Sensitive to the choice of kernel and its parameters
[16]	2024	Extreme gradient boosting (XGboost), Gaussian NB, LR, RF, SVM, and DT	UNSW-NB15	Among the classifiers that have been studied, the RF is the most effective in detecting DoS attacks	Assessing a few ML classifiers using particular attacks
[17]	2024	LR, RF, and DT	Kaggle	Improvement through scaling and data balancing	Specific domain applicability
[18]	2024	SVM, KNN	CICDoS 2017	Reduction of false positives, the minimization of central processing unit utilization and control channel bandwidth consumption, the improvement of packet delivery ratio, and the decrease in the number of flow requests submitted to the controller	Sensitivity to the choice of k in KNN, SVM does not perform very well when the data set has more noise

## 2.2. Deep learning-based techniques for DDoS detection in SDN contexts

To help in identifying and combating DDoS attacks in SDN domains, Alsudani *et al.* [19] provide the adversarial DBN-LSTM method employing generative adversarial network (GAN), deep belief networks and long short-term memory (DBN-LSTM) to minimize the sensitivity of the system to adversarial attacks and expedite feature extraction.

Yungaicela-Naula *et al.* [20] present an SDN-based method to automate detection and mitigation of slow-rate DDoS attacks. Reinforcement learning (RL) is applied in the framework to reduce attacks, while deep learning (DL) assists to identify them. Furthermore, a moving target defense (MTD) mechanism based on network function virtualization (NFV) is presented to improve the effectiveness and adaptability of the system. DL techniques that Mousa and Abdullah [21] are utilized for diagnostics of DDoS attacks as well as checkpoint networks, a fault-tolerant methodology for extended operations. Ali *et al.* [22] explore a number of classification algorithms (e.g., SVMs, KNNs, DTs, multilayer perceptrons (MLPs) and CNNs). The approach presented by Mansoor *et al.* [23] consists of three steps: data preprocessing, cross-feature selection (which aims to find essential features for DDoS detection) and detection with the use of the RNNs model. Wang *et al.* [24] presented six models: DNN, CNN, RNN, LSTM, CNN + RNN, CNN + LSTM to identify whether network traffic was a malicious attack. Consider the CNN and BiLSTM utilizing an attention technique, Said and Askerzade [25] intrusion diagnostic multiple models. The model comprises of three primary components: CNN, BiLSTM, and attention layer. The CNN layer focuses on various network traffic data to derive local features. The BiLSTM layer learns the temporal relationships of local features. The attention layer picks up the most relevant properties from the respective BiLSTM result for each type of intrusion for DL, Rao and Subbarao [26] employ the SVM, MLP, and LSTM algorithms.

The DL model that is proposed learns and generates binary and multiclass classification models to differentiate between routine traffic and network attack functions models and datasets are analyzed to detect anomalies and indicators of potential attacks. Rigorous and precise analysis is conducted within the deep

learning model. During diagnostics, the system differentiates between malicious activity and normal network traffic. Additionally, traffic patterns and protocol behavior are examined to enhance detection accuracy. Traffic filtering was then done to remove dubious traffic or attack signatures. Salih and Abdulrazaq improved a innovative DL framework called Cybernet [27]. Recognizing and diagnosing the sorts of DDoS attacks accurately, and understanding some of the most fundamental tactics in the field of cyber protection were the objectives. The key architectural component implemented is that the output from the LSTM and the 1D-CNN are added prior to feeding the hidden output to the next layer. Non-time series data is also applied in producing the features in the LSTM layers, and it helps the model to learn relevant attributes automatically with the inclusion of layers such as convolutional layers. Benmohamed *et al.* [28] proposed a new technique called Encoder-Stacked deep neural networks that leverage Stacked/bagged MLP. The suggested solution employs an encoder to identify relevant elements from a processed data set for precise attack identification, the comparison of various deep learning-based strategies for DDoS detection, along with their respective classifiers, datasets, and evaluation aspects, is summarized in Table 2.

Table 2. Summary of DL approaches for DDoS detection

Ref	Year	Classifier	Dataset	Pros	Cons
[19]	2023	GAN and DBN-LSTM	CICDDoS 2019	Effective DDoS detection	Real-time performance, reproducibility
[20]	2023	LSTM and RL	CICDoS2017	Efficient detection, minimal complexity	Validatethe one of the proposed framework
[21]	2023	LSTM and CNN	Mendeley Data UNSW_2018_I oT_Botnet	Single dataset used	Real-time testing, diverse datasets
[22]	2023	SVM, KNN, DT, MLP, CN	CIDS2017 and CICDDoS2019	Enhanced detection, scalability	Limited real-time assessment
[23]	2023	RNN	Mendeley Data	Low FPR, informative features	Limited to communication channels
[24]	2023	CNN + RNN, DNN, CNN, RNN, LSTM, and CNN + LSTM	CSE-CIC-IDS2018	Enhanced detecting capabilities over current systems	Additional validation is required in various network contexts
[25]	2023	CNN-BiLSTM	InSDN	Enhanced intrusion detection	Reliance on specific datasets
[26]	2024	SVM, MLP, and LSTM	NSL-KDD D	It reduces the impact of DoS/DDoS attacks on networks	
[27]	2024	LSTM, CNN	CIC-DDoS2019	superior detection and recognition capabilities, particularly when applied to unseen data	The lack of in-depth exploration into the interpretability of the proposed models
[28]	2024	Stacked deep neural networks (E-SDNN)	CICDS2017 and CICDDoS2019	the effectiveness of the E-SDNN model	the model's complexity and runtime analysis necessitate additional training time

### 3. METHOD

The present paper provided the optimized similarity-based graph clustering (O-SBGC), similarity-based graph clustering (SBGC) developed by the butterfly optimization algorithm (BOA) for tuning hyperparameters for DDoS attack diagnosis and early prevention. This section is primarily comprised of eight subsections: dataset preparation, preprocessing, splitting data, balancing data, BOA for optimizing hyperparameters, Feature extraction with O-SBGC-BiLSTM and ViT, ML optimization with Bayesian optimization algorithm (SVM, KNN, NB, RF), and Evaluation of models based on accuracy, recall, false positive rate (FPR), false negative rate (FNR), true negative rate (TNR), precision, F1-score, receiver operating characteristic (ROC), area under the curve (AUC). The overall workflow of the proposed DDoS detection system, incorporating O-SBGC-BiLSTM and ViT models, is illustrated in Figure 1.

#### 3.1. Bayesian optimization algorithm

Bayesian optimization refers to the method of optimization given the ML that is normally being applied for optimizing objective black-box tasks. The mechanism of Bayesian optimization applies probabilistic model integration to help optimization processes and techniques to model and sample Bayesian networks. The mechanism presents instance solutions population applying Bayesian networks. Bayesian optimization technique is being applied for solving broad issues' range. One of the most common issues that such a mechanism is applied for optimization refers to hyper-parameter ML models tuning for developing prediction performance. This achieves info from the last instances and updates optimization function sample points below. The task of utility is applied to choosing the next instance points for increasing the function output above.

BOA as the method of optimization starts with creating a random basic population of promising solutions known as strings, with a unique share over whole feasible strings. After that, such population would be updated over several iterations including four stages. Such four basic stages are repeated till promising actual variables. Basically, by performing the method of choice for the present population, satisfactory responses would be selected. Next, Bayesian network is made for fitting satisfactory promising solution population.

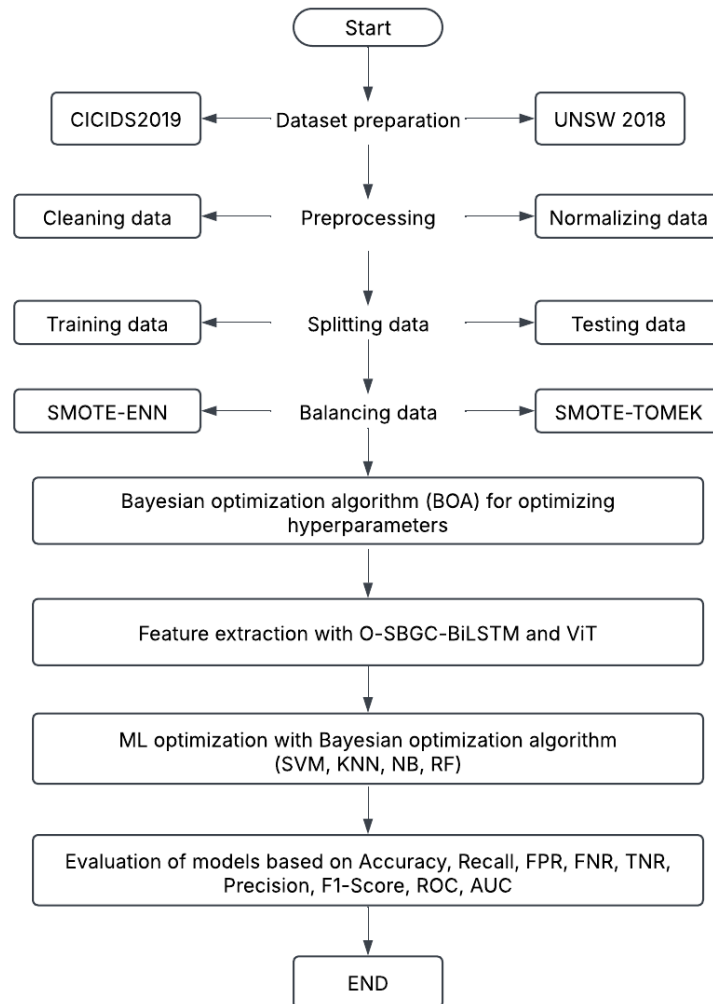


Figure 1. Flowchart proposed method

### 3.2. Feature extraction

The proposed model incorporates a dual-stream feature extraction process, combining the capabilities of SBGC and BiLSTM to capture spatial and temporal patterns from network traffic data. This process is further enhanced through integration with the ViT, which contributes deep attention-based embeddings to the overall representation.

The initial stage begins with the SBGC module, which transforms the raw input features into graph-structured data. In this context, each input feature vector corresponds to a node, and the edges between nodes are dynamically determined based on similarity metrics or correlation coefficients. This structure allows SBGC to accumulate and transfer context between features, capturing non-linear spatial information that is not preserved in flat representations. The SBGC block is composed of an input projection layer, a GCN block, and a dropout mechanism. This scheme enables the network to capture complex inter-feature relations while keeping the generality and without overfitting.

Then, the enhanced features generated by SBGC are fused with frame-difference vectors  $V_{ti}$  that reflect the temporal difference between adjacent frames. Such differences are crucial for distinguishing subtle behavioral changes of the traffic that reflect DDoS attacks. The spatial feature  $f_{ti}$  and temporal feature  $V_{ti}$  are

stacked and adopted to the BiLSTM network for time-sequence modeling. BiLSTM layer (with 128 hidden units for both directions) processes the concatenated input signals and models the bidirectional temporal dependencies. This architecture is favorable for dynamics intrusion detection, the model can take the context of a particular traffic windows, both the past and future observations into account.

Mathematically, the BiLSTM processing can be expressed as:

$$E_{ti} = f_{Bilstm}(\text{concat}(f_{ti}, V_{ti})) = f_{Bilstm}(\text{concat}(Af_{ti}, f_{(t-1)})) \quad (1)$$

The ViT processes the input data in a similar way by dividing it into fixed-size patch tokens. These tokens are embedded and enriched with positional encodings, then fed to a stack of Transformer encoder layers. The attention mechanism in the ViT allows for the model to learn long-range dependencies and to assign the varying weights to different parts of the input. The output of the ViT which is a high-dimensional feature vector capturing both global and local information.

After the per-frame extraction, the SBGC-BiLSTM and ViT pipelines are concatenated using a features concatenation layer, generating an integrated feature vector that preserves spatial, temporal, and attention-based context. This fusion guarantees that the model can take advantage of the complementary strengths from graph-based modeling, sequential learning and self-attention mechanisms.

The resulting vector is then provided to the classification stage where well-known machine learning algorithms as SVM, KNN, NB (classifications) and RF are employed. These classifiers operate on the fused representation to accurately detect and classify DDoS attack instances within the network traffic data [29].

### 3.3. Machine learning optimization with Bayesian optimization algorithm

Mechanisms of ML apply the extracted attributes from every layer of the DL model's feature for making classifications. For obtaining the highest feasible accuracy in classification, this is essential for recognizing agents that impact mechanisms' accuracy and selecting the best amounts given the chosen attributes. Here, some papers exist about DL models and ML mechanisms' parameter optimization. Here, every ML algorithm hyperparameters were chosen via Bayesian optimization and standard eight-fold cross-validation way in the processing of training. Various ML mechanisms have uniform hyperparameters that could be optimized. For instance, KNN has hyperparameters such as distance weight, metric, and neighbors' number; SVM has hyperparameters such as box kernel kind, multiclass method, and limitation level; RF has hyperparameters such as max splits number and split variable; NB has hyperparameters such as kind of kernel as well as name of share.

NB algorithm and kernel kind applied in its impact how appropriate predictions are four options exist: Triangle, Gaussian, Box, and Epanechnikov. Every kernel has some weaknesses and strengths. GNB models apply Gaussian share for computing every level of feasibilities when kernel Naive Bayes models apply selected kernel share. The SVM algorithm has three hyperparameters: multiclass technique, kernel kind, and box limitation class. Kernel kind decides how data is transformed before classification and could be linear, cubic, quadratic, Gaussian radial basis function (RBF). KNN algorithm has 3 basic adjustments for being optimized to obtain the best outcomes: distance weight and metric, and neighbors' number. Neighbors' number shows how many nearby neighbors must be considered to group every location. The metric of distance computes distance among data points, various accessible options exist such as City block, Euclidean, and so on. Random forest refers to a set of big decision trees' amounts. Every tree is built by selecting k random features and applying a training set of data. Choose n decision trees number (n estimators) to be created in the forest. A more considerable tree number normally causes the developed accuracy however at the developed cost of computation.

Algorithm 1 outlines a comprehensive pipeline for detecting DDoS attacks using the O-SBGC-BiLSTM-ML framework. Initially, the necessary components, including model hyperparameters and training parameters, are configured. Preprocessing begins with loading the DDoS datasets, followed by splitting them into training, validation, and test subsets. The BOA is then employed to optimize the hyperparameters of the O-SBGC-BiLSTM model, ensuring improved accuracy. This optimized model extracts high-level temporal features from the network traffic data, which are subsequently refined and encoded by a ViT through attention-based feature enhancement. The features from both models are concatenated to form a hybrid representation. In the classification phase, traditional ML classifiers—SVM, NB, KNN, and RF—are applied to the hybrid feature set. The model is trained, validated, and then tested, with performance evaluated using metrics like precision, recall, F1-score, and accuracy. Threshold-based binary decisions are made during testing to classify traffic as benign or DDoS. Finally, ROC and AUC metrics are computed to assess the model's overall detection capability, and suspicious samples are logged for further analysis.

---

**Algorithm 1. Algorithm for DDoS detection and classification using O-SBGC-BiLSTM-ML model**

---

Input: Dataset of DDoS with corresponding labels  
 Training parameters  
 Test dataset for evaluation.  
 Output: DDoS detection and classification Evaluation metrics (accuracy, precision, and recall)

- 1: Load and preprocess the dataset.
- 2: Split the dataset into train, validation, and test sets.
- 3: Bayesian optimization algorithm
  - i. Implement BOA for optimizing hyperparameters in SBGC-BiLSTM.
  - ii. Implement BOA for optimizing hyperparameters in ML algorithms.
- 4: O-SBGC-BiLSTM Feature Extraction:
  - i. Implement an O-SBGC-BiLSTM for feature extraction.
  - ii. Train the O-SBGC-BiLSTM using the training dataset.
  - iii. Extract high-level features using the trained O-SBGC-BiLSTM.
- 5: Vision Transformer (ViT) Encoding:
  - i. Implement a ViT to extract features.
  - ii. Fine-tune the ViT using the training dataset.
  - iii. Encode the extracted features with the trained ViT.
- 6: Concatenate the extracted features of O-SBGC-BiLSTM and ViT.
- 7: Implement ML algorithms as classification heads (SVM, NB, KNN, RF).
- 8: Train the classification head using the fused features from step 5.
- 9: Model Evaluation:
  - i. Evaluate the O-SBGC-BiLSTM-ML model using the validation set.
  - ii. Fine-tune hyperparameters if necessary.
- 10: Model Testing:
  - i. Test the trained model on the testing dataset.
  - ii. Compute classification metrics (e.g., Accuracy, Recall, FPR, FNR, TNR, Precision, F1-score, ROC, AUC).

---

## 4. RESULTS AND DISCUSSION

In this section, we evaluate the proposed IDS experimentally. The performance and effectiveness of the system were evaluated on two benchmark datasets: UNSW-NB15 and CICIDS2019. Detection accuracy and robustness were measured using standard evaluation metrics. All experiments were implemented in Python, on a machine with an Intel Core i7-7700 processor and 32 GB of RAM.

### 4.1. Collection and characteristics of dataset

In order to carefully review performance and robustness of the proposed DDoS detection model. Two publicly accessible benchmark datasets were used: UNSW-NB15 and CICDDoS2019. These are realistic network-wide DDoS attack datasets containing legitimate and attack traffic for training and validation of intrusion detection model, especially in SDN and an advanced DDoS attack case.

#### 4.1.1. UNSW-NB15 dataset

The UNSW-NB15 dataset was originally constructed by the Australian Centre for Cyber Security (ACCS) at the University of New South Wales. The raw network packets were generated using the IXIA PerfectStorm tool, which simulates a hybrid environment of normal and malicious activities to mimic real-world traffic conditions. The dataset includes both legitimate traffic and nine categories of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms [30].

The Kaggle-hosted version of the dataset [31] provides the preprocessed and labeled data in CSV format, consisting of 175,341 samples in the training set and 82,332 samples in the testing set. Each instance in the dataset is characterized by a set of 49 features derived from raw packet captures, encompassing flow-based, content-based, and time-based attributes. The dataset's class distribution reflects real-world imbalance, with a high proportion of normal traffic, thus offering an appropriate challenge for intrusion detection systems to achieve high detection accuracy without sacrificing generalizability.

#### 4.1.2. CICIDS2019 dataset

The CICDDoS2019 dataset was created by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. It is among the last and the most complete datasets especially developed to analyze modern DDoS attacks. The traffic set contains intermediary traffic and a wide scope of recent DDoS variants: port mapping (PortMap), network basic input/output system (NetBIOS), lightweight directory access protocol (LDAP), Microsoft SQL server (MSSQL), user datagram protocol (UDP), UDP latency attack (UDP-Lag), SYN, network time protocol (NTP), domain name system (DNS), and simple network management protocol (SNMP) flood attacks are included. These attacks were implemented with open-source attack packages in a testbed to emulate volumetric and reflective flooding situations aiming at generating test data that closely represents a real-world scenario [32].



CICDDoS2019 [33] dataset is available in a structured format to be used for ML, including more than 80 flow features captured via CICFlowMeter. The data is divided in between two days of collection, in order to keep performance tested in temporal differences. Due to such rich feature space and high diversity of attack vectors, it is an attractive baseline for model's adaptability performance and detection accuracy.

#### 4.2. Usefulness of the dataset

This research focuses on the detection and classification of DDoS attacks—cyber threats characterized by overwhelming a target system using multiple compromised devices acting in coordination. This differs from DoS attacks, which originate from a single attacking device to exhaust the resources of a target.

While CICDDoS2019\_Kaggle directly aligns with the scope of this study by providing labeled data for various real-world DDoS attack types, UNSW-NB15\_Kaggle also plays a valuable role. Despite being focused on DoS and other traditional attacks, it offers rich, diverse features and complex multi-class scenarios that can enhance the model's ability to learn generalized attack patterns. This is particularly important for pre-training stages, enabling the detection framework to differentiate between benign and malicious behaviors under different attack surfaces. Moreover, using both datasets improves the system's cross-domain adaptability and resilience to evolving attack techniques, strengthening its applicability in real-world SDN environments.

#### 4.3. Preprocessing

These include recognizing important components that are important for later analysis, normalizing features to unique scales for stability, and cleaning raw data to remove noise and redundancy. Such a step optimizes data from particular scenarios and lays the foundation for the model to learn efficiently. Z-score normalization was applied to standardize the dataset, ensuring that features had a mean of zero and a standard deviation of one. Z-score normalization involves transforming values by sharing by standard deviation and subtracting the mean [34]. The z-score normalization formula is:

$$V'_i = \frac{V_i - \bar{E}}{std(E)} \quad (2)$$

where  $V_i$  is the value of row E of the i-th column and  $V'_i$  = z-score normalized values.

$$std(E) = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^n (V_i - \bar{E})^2} \quad (3)$$

and mean value,

$$\bar{E} = \frac{1}{n} \sum_{i=1}^n V_i \quad (4)$$

Data normalization can enhance model performance, avoid overfitting, and promote convergence. Scaling the data to a range between 0 and 1 is the method employed in data normalization. All feature values are then converted to a common scale.

#### 4.4. Splitting the dataset

Using an 80% training and 20% testing rate, the structured data set is divided into training and testing subsets. This was achieved by implementing the default train-test split function available in Python's model selection module, ensuring an appropriate division of data for model evaluation. To safeguard each assault level proportionately represented in a basic collection of data, the order not only shuffles a set of data but also creates a stratified strategy.

#### 4.5. Balancing techniques

Two distinct training data preparation methods were applied, specifically tailored for nominal data. These methods included undersampling, oversampling, and a combination of various resampling strategies. The primary objective was to balance the datasets and enhance the performance of the developed models.

SMOTE-edited nearest neighbors (SMOTE-ENN) appears as SMOTE and ENN methods' fusion, efficiently dealing with imbalanced data classification concerns. By integrating SMOTE's minority-level oversampling with ENN's majority-class undersampling, the technique harmonizes the share of the dataset. Such a method was formulated by [35] as a strong strategy to control class imbalance.

Likely, SMOTE-Tomek links (SMOTE-TOMEK) is the other amalgamation method developed in imbalanced data classification scenarios. This is a potent approach renowned for its efficacy in considering imbalanced sets of data, specifically while meeting noisy/overlapping samples. By combining SMOTE and Tomek links, such a method efficiently navigates imbalanced scenarios for developing model performance.

#### 4.6. Performance metrics

The presented performance of the intrusion model was evaluated using the variables of recall, accuracy, F-score, and precision. The classification metrics overview is presented here. Appropriately grouped data rate out of whole grouped data is how appropriately something is grouped. The precision determines the accuracy of optimistic predictions; a lower false positive ratio denotes more precision. The percentage of samples that are accurately classified as positive is known as recall. Recall and precision are combined in the F-score to create a fair assessment metric. This is feasible to describe that as a precision and recall medium. Whole metrics are given the confusion matrix. This matrix rows and columns are labeled with certain levels (ground truth) and predicted levels in turn. Table 3 shows the normal confusion matrix for a binary classification issue.

Table 3. Confusion matrix for a binary classification

Actual/predicted	Predicted positive (attack)	Predicted negative (normal)
Actual positive (attack)	TP	FN
Actual negative (normal)	FP	TN

Additionally, true positive, true negative, false positive, and false negative are denoted as  $TP$ ,  $TN$ ,  $FP$ , and  $FN$ , respectively.

True positive (TP) refers to attack data volume in network traffic that is labeled as an attack.

True negative (TN) shows network data volume in the network traffic that is considered normal.

False positive (FP) is normal instances in network traffic that are misclassified as attack instances.

False negative (FN) relates to attack instances in network traffic That are misclassified as normal.

The classification metrics assessing an IDS system are described:

Classification rate or accuracy (CR): this is the accuracy rate on diagnosing unusual/usual manner. This is usually applied while a set of data is balanced to prevent a good model performance from false sense.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Recall (R) or true positive rate (TPR) or Sensitivity: this caused by sharing number of accurately estimated attacks by entire attacks' number. This is classifier ability to diagnose whole positive cases (attacks).

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

Precision (P): shows attack diagnosis confidence. This is effective but that is not recommended to apply Accuracy due to the set of data is not balanced.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

F1-score: this is the harmonic mean among precision and recall (TPR), Achieving a value close to the lower of the two class distributions, particularly in cases of class imbalance, indicates improved classification balance. A value approaching 1 suggests that the classifier is performing optimally in distinguishing between the classes.

$$F1 - Score = \frac{2PR}{P+R} \quad (8)$$

#### 4.7. Training and testing

The dataset is partitioned into 80% for training and 20% for testing for the evaluation of the proposed method. To reduce the model error, the training process is performed for 100 epochs, using a learning rate of 0.001 for quick convergence. Figures 2 and 3 show trends of loss and accuracy during both training and testing stages in two datasets. The loss trends are depicted in Figures 2(a) and 3(a), while the accuracy trends are shown in Figures 2(b) and 3(b). The loss values were between 0.10 and 0.50. This evaluation was on the benchmark datasets but the performance of the proposed model suggests its capability in identifying attacks not specifically tested in the evaluation. In addition, as shown in Figures 1 and 2, the value of training loss starts high and then decreases slowly in the training process. A significant slowing down in the reduction of errors is usually observed after about 20 epochs.

The accuracy of training and testing steadily increased as the number of epochs increased and finally maintain at the higher levels as performed in Figure 2(a) indicates the model has a good generalization property in transferring the learning model as a mapping from the training data to unseen samples. The corresponding

loss values for training and testing are visualized in Figure 2(b) instead. Here we have a loss which starts from a high point but decrease monotonically over the epochs and it reflect the ability of your model to minimize the error in prediction. The reduction of error slows down after 20 epochs, we assume that this is due to convergence of learning.

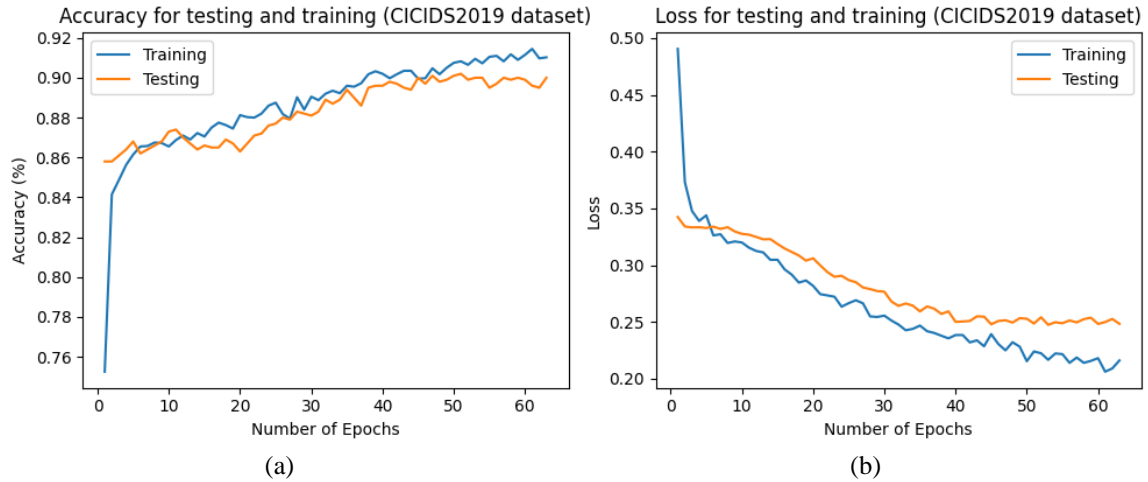


Figure 2. Training and testing: (a) accuracy and (b) loss on CICIDS2019 dataset

This figure demonstrates the training procedure of the model on UNSW-NB15 dataset for 100 epochs: Figure 3(a) shows the evolution of accuracy on training and testing data, showing an increasing trend in accuracy for training and a high level of accuracy in validation, which indicates the powerful learning ability of the model. We present the loss trend in Figure 3(b), where we observe that the training and test loss gradually decrease, demonstrating that the optimization calculation is successfully conducted and the prediction error decreases while training.

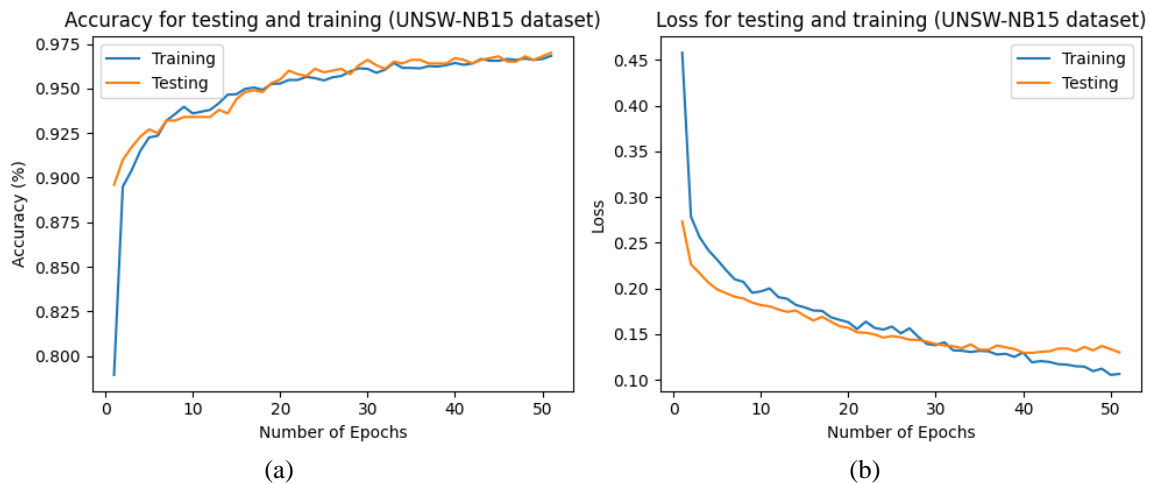


Figure 3. Training and testing: (a) accuracy and (b) loss on UNSW-NB15 dataset

#### 4.8. Performance evaluation on the CICIDS2019 dataset

The CICIDS2019 dataset has undergone a number of tests to evaluate the efficacy of the proposed methodology. Table 4 displays the classification outcome of the suggested method. The performance of the proposed technique in identifying DDoS attacks on the CICIDS2019 dataset is shown in Table 4. The model's 98.98% accuracy shows that the recommended approach can identify DDoS assaults from regular traffic with very little error. This high accuracy implies that the model can discriminate between classes (attacks and normal traffic). Furthermore, the recall score of 98.92% implies that the model can detect positive samples (DDoS

assaults) among all accessible positive samples. This criterion indicates that the model rarely misses attack samples. The precision value is also 98.95%, indicating that a large number of the samples identified by the model as assaults were attacks. Finally, the F1-score is 98.98%, showing an excellent balance of accuracy and recovery. This score indicates that the model was able to detect attacks with high accuracy but also misclassified a small percentage of samples. Overall, the findings demonstrate the proposed method's great performance in accurately and efficiently detecting DDoS attacks, confirming its superiority over numerous current methods.

Table 4. Classification of the proposed approach on the CICIDS2019 dataset

Technique	Accuracy	Recall	Precision	F1-score
Proposed method	98.98	98.92	98.95	98.98

#### 4.9. Performance evaluation on the UNSW-NB15 dataset

Several tests have been conducted on the UNSW-NB15 dataset to evaluate the effectiveness of the proposed methodology. The categorization results for the suggested method are shown in Table 5. The performance of the proposed method on the UNSW-NB15 dataset is shown in Table 5. With an accuracy of 99.79%, the results show that the model does a very good job of recognizing and differentiating attack samples from regular traffic. The model's predictions have the least amount of error at this level of accuracy. With a 99.74% recall rate, the model is able to identify nearly all positive cases (attacks) in the data. According to this score, assault samples are rarely missed by the model. Additionally, the accuracy of the model is reported to be 99.71%, which shows that the error rate of model predictions for attack samples is quite low. This indicates that samples that are labeled as attacks are almost never incorrectly classified. The precision and recovery weighted average, or F1-score, is 99.73%, suggesting that these two factors are fairly matched. This figure suggests that although the model correctly recognized the attack samples, a very small percentage of the samples were incorrectly classified. On the UNSW-NB15 dataset, the proposed method performs remarkably well overall. These findings demonstrate the model's high accuracy and dependability, suggesting that it might be used as a useful tool for identifying online dangers.

Table 5. Classification of the proposed approach on the UNSW-NB15 dataset

Technique	Accuracy	Recall	Precision	F1-score
Proposed method	99.79	99.74	99.71	99.73

#### 4.10. Comparison proposed method

In Table 6, the proposed method's performance on the UNSW-NB15 dataset is compared to two existing approaches. The approach [15] has an accuracy of 98.5%, a recall of 99.6%, a prediction accuracy of 96.1%, and an F1 value of 97.8%. Although this method has a high recall, its prediction accuracy is lower than previous methods, indicating a decline in performance in properly detecting categories. The approach [16] outperforms the method [15], providing 99.4% precision, 99.6% recall, 99.2% prediction accuracy, and an F1 value of 99.4%. This method is more efficient in identifying assaults in the UNSW-NB15 dataset, with improvements in all criteria compared to [15]. In this comparison, the proposed method outperforms the other methods with 99.79% accuracy, 99.74% recall, 99.71% prediction accuracy, and 99.73% F1. These outstanding findings demonstrate the suggested method's great efficiency in accurately and comprehensively detecting attacks. The proposed method exhibits close precision, recall, and F1 values, indicating it reduces false positive and negative mistakes while maintaining a high balance in evaluation criteria.

Overall, the results show that the proposed method, which employs advanced methodologies and proper optimization, is more accurate and comprehensive than earlier methods and serves as an effective tool for detecting attacks in UNSW-NB15 networks.

Table 6. Comparing the performance of the proposed method with other methods on the UNSW-NB15 dataset

Technique	Dataset	Accuracy	Recall	Precision	F1-score
[15]	UNSW-NB15	98.5	99.6	96.1	97.8
[16]	UNSW-NB15	99.4	99.6	99.2	99.4
Proposed method	UNSW-NB15	99.79	99.74	99.71	99.73

Table 7 compares the results of different DDoS attack detection algorithms using the CICIDS2019 dataset. The results demonstrate that the suggested method performed significantly better than other methods, with 98.98% precision, 98.92% recall, 98.95% accuracy, and 98.98% F1-score. While other approaches, such as [19] and [28], performed well (accuracy 91.23% and 98.86%, respectively), none of them provided more complete information, such as recall, precision, or F1-score. The suggested method outperforms existing approaches in all evaluation metrics (accuracy, recall, precision, and F1-score) and appears to be a better approach for intrusion detection in the CICIDS 2019 dataset.

Table 7. Comparing the performance of the proposed method with other methods on the CICIDS2019 dataset

Technique	Dataset	Accuracy	Recall	Precision	F1-score
[19]	CICIDS2019	91.23	-	-	-
[28]	CICIDS2019	98.86	-	-	-
[22]	CICIDS2019	90.09	-	-	84
Proposed method	CICIDS2019	98.98	98.92	98.95	98.98

#### 4.11. Discussion

The SBGC-BiLSTM-ViT framework demonstrated high performance in detecting DDoS attacks on UNSW-NB15 and CICIDS2019 datasets. The integration of SBGC, BiLSTM, and ViT, along with Bayesian optimization, significantly enhanced detection accuracy and generalization. It achieved 98.98% accuracy on CICIDS2019 and 99.79% on UNSW-NB15, outperforming state-of-the-art methods.

The model benefits from temporal learning BiLSTM, structural aggregation (SBGC), and contextual modeling (ViT). SMOTE and normalization further improved robustness to class imbalance. Compared to Amazon web services (AWS) web application firewall (WAF)/Shield, the proposed model adapts dynamically to unseen threats, offering superior flexibility and accuracy.

Overall, the results validate SBGC-BiLSTM-ViT as a scalable, adaptive IDS suitable for edge computing and SDN environments. Future work should explore real-time deployment and resilience against adversarial attacks.

## 5. CYBERSECURITY IMPACT OF THE PROPOSED IDS FRAMEWORK

The proposed hybrid framework—SBGC-BiLSTM-ViT—introduces a novel approach to DDoS attack detection by integrating advanced deep learning architectures with Bayesian optimization. While its detection performance demonstrates superiority over existing academic approaches, its significance in the broader cybersecurity landscape is better understood through its comparative value against real-world DDoS countermeasure solutions. This section discusses the method's practical implications, its advantages over commercial tools, and its role in improving network defense.

### 5.1. Detection capabilities beyond academic models

The experimental results obtained on the UNSW-NB15 and CICDDoS2019 datasets demonstrate that the proposed method outperforms conventional models in terms of accuracy, precision, recall, and F1-score. Through the integration of SBGC, BiLSTM, and ViT, the system captures intricate spatial, sequential, and attention-based representations of network traffic. Bayesian optimization further refines the learning process by tuning hyperparameters for both deep learning and machine learning components. These improvements allow the model to detect a wide range of attacks, including low-volume and zero-day DDoS variants, which many traditional methods fail to identify.

### 5.2. Comparative evaluation against industrial DDoS countermeasures

To contextualize its cybersecurity impact, the proposed method must be evaluated not only against prior research but also relative to widely used industry tools such as AWS WAF and AWS Shield Standard. These platforms are commonly deployed for defending web applications and cloud infrastructure from DDoS threats.

AWS WAF applies rate-based rules to detect and mitigate SYN flood attacks and other application-layer threats through customizable filtering [36]. AWS Shield Standard, on the other hand, provides automatic protection against volumetric DDoS threats, including UDP flood and SYN flood attacks, by monitoring and absorbing malicious traffic based on pre-defined heuristics [37]. However, these systems primarily rely on static thresholding, rule-based signatures, and predefined traffic patterns.

In contrast, the SBGC-BiLSTM-ViT framework introduces a dynamic and context-aware detection mechanism. The SBGC component builds graph-structured relationships between features, while the BiLSTM

captures temporal attack signatures. ViT further enhances classification accuracy through self-attention-driven feature prioritization. This synergy enables the model to detect not only known attack variants but also evolving and stealthy zero-day threats, which often evade threshold-based systems like WAFs. A comparative summary is provided in Table 8.

Table 8. Comparison of SBGC-BiLSTM-ViT with AWS WAF and shield for DDoS mitigation

Feature	Proposed method (SBGC-BiLSTM-ViT)	AWS WAF	AWS Shield Standard
Attack types detected	DDoS (SYN, UDP, zero-day)	SYN (rate-based rules)	UDP, SYN, volumetric attacks
Learning capability	Adaptive deep learning	Static rule-based	Threshold/heuristic-based
Detection of zero-day attacks	Yes	No	No
Temporal pattern analysis	BiLSTM-based	No	No
Graph-based contextual learning	SBGC integrated	No	No
Attention-driven feature focus	Vision transformer	No	No
Edge/SDN deployability	Lightweight model	Cloud-only	Cloud-only

This comparison highlights the proactive and intelligent nature of the proposed system. Instead of relying on volume-triggered heuristics, the model learns dynamic behaviors, detecting early-stage anomalies prior to their escalation into large-scale DDoS attacks enables proactive threat mitigation and enhances network resilience.

### 5.3. Expected impact on real-world cybersecurity

The proposed framework can function as a complementary layer to infrastructure-level defenses such as WAF and Shield by embedding it into SDN controllers, network gateways, or edge computing devices. Unlike traditional countermeasures that focus on bulk traffic patterns, this model emphasizes feature semantics, behavioral deviations, and sequential dependencies—providing a fine-grained and interpretable security layer for real-time defense.

In modern cybersecurity frameworks, such capabilities are aligned with zero trust architecture (ZTA) and behavioral anomaly detection principles [38]. The system’s ability to generalize across datasets and adapt to new attack vectors makes it ideal for deployment in critical infrastructure, IoT ecosystems, and enterprise networks. By reducing false positives, enabling early detection, and adapting to unseen threats, the SBGC-BiLSTM-ViT framework serves as a strategic enhancement to existing DDoS protection infrastructures, contributing to a more resilient, intelligent, and proactive cybersecurity posture.

## 6. CONCLUSION

This study introduced a hybrid intrusion detection framework, SBGC-BiLSTM-ViT, which integrates SBGC, BiLSTM, and ViT models, optimized through the BOA. The proposed approach effectively addresses the challenges of detecting and classifying DDoS attacks by combining graph-based spatial feature extraction, sequential pattern recognition, and attention-driven contextual analysis. Experimental results on benchmark datasets demonstrated that the integration of ViT for feature embedding and BiLSTM for temporal modeling significantly enhanced the system’s detection accuracy and generalization ability. The BOA further contributed to performance improvement by optimizing hyperparameters across both the deep learning and classification stages. Beyond achieving high performance in detection tasks, this work contributes to the field of cybersecurity by offering a scalable and adaptive detection model that can be integrated into real-time, resource-constrained environments such as software-defined networks and edge computing systems. Future research should explore deploying the framework in live environments, evaluating its resilience under adversarial conditions, and extending its applicability to a broader range of cyber threats, including multi-vector and cross-protocol attacks.

## FUNDING INFORMATION

Authors state no funding involved.

## 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
Huda Mohammed Ibadi	✓	✓	✓	✓		✓	✓		✓		✓			
Asghar Asgharian Sardroud	✓	✓		✓		✓				✓		✓	✓	

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review &amp; Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

## CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

## DATA AVAILABILITY

The data that support the findings of this study are openly available in *Kaggle* [33], at <https://www.kaggle.com/datasets/dhoogla/cicddos2019>.




## REFERENCES

- [1] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "A deep learning ensemble approach to detecting unknown network attacks," *Journal of Information Security and Applications*, vol. 67, Jun. 2022, doi: 10.1016/j.jisa.2022.103196.
- [2] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *EURASIP Journal on Information Security*, vol. 2019, no. 1, Dec. 2019, doi: 10.1186/s13635-019-0084-4.
- [3] Swati, S. Roy, J. Singh, and J. Mathew, "Securing IIoT systems against DDoS attacks with adaptive moving target defense strategies," *Scientific Reports*, vol. 15, no. 1, Mar. 2025, doi: 10.1038/s41598-025-93138-7.
- [4] R. Gopi *et al.*, "Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things," *Multimedia Tools and Applications*, vol. 81, no. 19, pp. 26739–26757, Aug. 2022, doi: 10.1007/s11042-021-10640-6.
- [5] 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*, vol. 3, Dec. 2025, doi: 10.1016/j.csa.2025.100085.
- [6] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, Dec. 2017, doi: 10.1177/1550147717741463.
- [7] R. R. Sekar, A. M. Jenny, D. Sreshta, M. Vikas, D. B. N. Ajay, and M. Ganesh, "Prediction of distributed denial of service attacks in SDN using machine learning techniques," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Jun. 2023, pp. 1–5, doi: 10.1109/CONIT59222.2023.10205887.
- [8] S. Sanapala, D. D. Reddy, G. L. Chowdary, and K. S. Vikyath, "Machine learning based DDoS attack detection in software defined networks (SDN)," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, Jul. 2023, pp. 1124–1126, doi: 10.1109/ICECAA58104.2023.10212147.
- [9] A. K. Tahirou, K. Konate, and M. M. Soidridine, "Detection and mitigation of DDoS attacks in SDN using machine learning (ML)," in *2023 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA)*, May 2023, pp. 52–59, doi: 10.1109/ICDATA58816.2023.00019.
- [10] S. Alsudani and M. N. Saeed, "Enhancing thyroid disease diagnosis through emperor penguin optimization algorithm," *Wasit Journal for Pure sciences*, vol. 2, no. 4, pp. 66–79, Dec. 2023, doi: 10.31185/wjps.230.
- [11] S. Feng, G. Yang, and W. Man, "Research on DDoS attack detection based on machine learning in SDN environment," in *2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Sep. 2023, pp. 821–825, doi: 10.1109/ITOEC57671.2023.10291822.
- [12] A. A. Alashhab, M. S. M. Zahid, M. Alashhab, and S. Alashhab, "Online machine learning approach to detect and mitigate low-rate DDoS attacks in SDN-based networks," in *2023 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, Sep. 2023, pp. 152–157, doi: 10.1109/IICAIET59451.2023.10291787.
- [13] B. A. Almohagri, M. A. Saeed, H. M. Alazaby, and A. I. Mohammed, "Machine learning approach for distributed denial of service attack detection in SDNs," in *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, Oct. 2023, pp. 01–07, doi: 10.1109/eSmarTA59349.2023.10293527.
- [14] E. Berei, M. A. Khan, and A. Oun, "Machine learning algorithms for DoS and DDoS cyberattacks detection in real-time environment," in *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, Jan. 2024, pp. 1048–1049, doi: 10.1109/CCNC51664.2024.10454755.
- [15] S. Das, M. Ashrafuzzaman, F. T. Sheldon, and S. Shiva, "Ensembling supervised and unsupervised machine learning algorithms for detecting distributed denial of service attacks," *Algorithms*, vol. 17, no. 3, Feb. 2024, doi: 10.3390/a17030099.
- [16] O. Almomani, A. Alsaaidah, A. A. A. Shareha, A. Alzaqebah, and M. Almomani, "Performance evaluation of machine learning classifiers for predicting denial-of-service attack in internet of things," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 1, 2024, doi: 10.14569/IJACSA.2024.0150125.
- [17] G. Airlangga, "Analysis and comparison of machine learning techniques for DDoS attack classification in network environments," *Jurnal Informatika Ekonomi Bisnis*, pp. 38–46, Mar. 2024, doi: 10.37034/infkeb.v6i1.795.
- [18] N. Nisa, A. S. Khan, Z. Ahmad, and J. Abdullah, "TPAAD: Two-phase authentication system for denial of service attack detection and mitigation using machine learning in software-defined network," *International Journal of Network Management*, vol. 34, no. 3, May 2024, doi: 10.1002/nem.2258.




- [19] S. Alsudani, H. Nasrawi, M. Shattawi, and A. Ghazikhani, "Enhancing spam detection: A crow-optimized FFNN with LSTM for email security," *Wasit Journal of Computer and Mathematics Science*, vol. 3, no. 1, pp. 28–39, Mar. 2024, doi: 10.31185/wjcms.199.
- [20] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Pérez-Díaz, "SDN/NFV-based framework for autonomous defense against slow-rate DDoS attacks by using reinforcement learning," *Future Generation Computer Systems*, vol. 149, pp. 637–649, Dec. 2023, doi: 10.1016/j.future.2023.08.007.
- [21] A. K. Mousa and M. N. Abdullah, "An improved deep learning model for DDoS detection based on hybrid stacked autoencoder and checkpoint network," *Future Internet*, vol. 15, no. 8, Aug. 2023, doi: 10.3390/fi15080278.
- [22] T. E. Ali, Y.-W. Chong, and S. Manickam, "Comparison of ML/DL approaches for detecting DDoS attacks in SDN," *Applied Sciences*, vol. 13, no. 5, Feb. 2023, doi: 10.3390/app13053033.
- [23] A. Mansoor, M. Anbar, A. Bahashwan, B. Alabsi, and S. Rihan, "Deep learning-based approach for detecting DDoS attack on software-defined networking controller," *Systems*, vol. 11, no. 6, Jun. 2023, doi: 10.3390/systems11060296.
- [24] Y.-C. Wang, Y.-C. Hwang, H.-X. Chen, and S.-M. Tseng, "Network anomaly intrusion detection based on deep learning approach," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042171.
- [25] R. B. Said and I. Askerzade, "Attention-based CNN-BiLSTM deep learning approach for network intrusion detection system in software defined networks," in *2023 5th International Conference on Problems of Cybernetics and Informatics (PCI)*, Aug. 2023, pp. 1–5, doi: 10.1109/PCI60110.2023.10325985.
- [26] G. S. Rao and P. K. Subbarao, "A novel framework for detection of DoS/DDoS attack using deep learning techniques, and an approach to mitigate the impact of DoS/DDoS attack in network environment," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 1, pp. 450–466, 2024.
- [27] A. A. Salih and M. B. Abdulrazaq, "Cybernet model: A new deep learning model for cyber DDoS attacks detection and recognition," *Computers, Materials & Continua*, vol. 78, no. 1, pp. 1275–1295, 2024, doi: 10.32604/cmc.2023.046101.
- [28] E. Benmohamed, A. Thaljaoui, S. Elkhediri, S. Aladhadh, and M. Alohal, "E-SDNN: encoder-stacked deep neural networks for DDOS attack detection," *Neural Computing and Applications*, vol. 36, no. 18, pp. 10431–10443, Jun. 2024, doi: 10.1007/s00521-024-09622-0.
- [29] A. Hirs, L. Audah, A. Salh, M. A. Alhartomi and S. Ahmed, "Detecting DDoS Threats Using Supervised Machine Learning for Traffic Classification in Software Defined Networking," in *IEEE Access*, vol. 12, pp. 166675–166702, 2024, doi: 10.1109/ACCESS.2024.3486034.
- [30] "UNSW-NB15 Dataset," *UNSW-NB15 Dataset – The University of New South Wales*. [Online]. Available: <https://research.unsw.edu.au/projects/unsw-nb15-dataset> (accessed May 30, 2025).
- [31] Kaggle, "UNSW-NB15 network traffic Dataset." <https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15> (accessed May 30, 2025).
- [32] "CICDDoS2019 Dataset," *Canadian Institute for Cybersecurity*, 2019. [Online]. Available: <https://www.unb.ca/cic/datasets/ddos-2019.html> (accessed May 30, 2025).
- [33] "CICDDoS2019 Dataset," *Kaggle*, 2019. [Online]. Available: <https://www.kaggle.com/datasets/dhoogla/cicddos2019> (accessed May 30, 2025).
- [34] A. Lazaris and V. K. Prasanna, "An LSTM Framework for Software-Defined Measurement," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 855–869, Mar. 2021, doi: 10.1109/TNSM.2020.3040157.
- [35] I. Ortet Lopes, D. Zou, F. A. Ruambo, S. Akbar, and B. Yuan, "Towards Effective Detection of Recent DDoS Attacks: A Deep Learning Approach," *Security and Communication Networks*, vol. 2021, pp. 1–14, Nov. 2021, doi: 10.1155/2021/5710028.
- [36] "AWS WAF – Web Application Firewall," *Amazon Web Services*. [Online]. Available: <https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html> (accessed May 30, 2025).
- [37] "AWS Shield – DDoS Protection," *Amazon Web Services*. [Online]. Available: <https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html> (accessed May 30, 2025).
- [38] Ebuka Mmaduekwe Paul, Ebuka Mmaduekwe Paul, Joseph Darko Kessie, and Mukhtar Dolapo Salawudeen, "Zero trust architecture and AI: A synergistic approach to next-generation cybersecurity frameworks," *International Journal of Science and Research Archive*, vol. 13, no. 2, pp. 4159–4169, Dec. 2024, doi: 10.30574/ijrsra.2024.13.2.2583.

## BIOGRAPHIES OF AUTHORS



**Huda Mohammed Ibadi**    received her B.Sc. degree in Computer Science from the University of Kerbala in 2014. She then pursued her M.Sc. degree in Computer Science at Imam Reza International University, graduating in 2021. Currently, she is a Ph.D. student in Computer Engineering at Urmia University, Iran. Her research interests include artificial intelligence, machine learning, deep learning, computer vision, and optimization algorithms. She is particularly focused on developing innovative computational techniques for real-world problems and enhancing the efficiency of AI-driven systems. She can be contacted at email: [dr.hudaibadi@gmail.com](mailto:dr.hudaibadi@gmail.com).



**Asghar Asgharian Sardroud**    received the B.Sc. degree in Computer Engineering - Software from Urmia University in 2006, the M.Sc. degree in Computer Engineering - Software from Sharif University of Technology in 2009, and the Ph.D. degree in Computer Engineering - Software from Amirkabir University of Technology in 2015. He is currently an Assistant Professor in the Department of Computer Engineering at Urmia University. His research interests include software engineering, artificial intelligence, and computational systems. He can be contacted at email: [a.asgharian@urmia.ac.ir](mailto:a.asgharian@urmia.ac.ir).