Egyptian Informatics Journal 27 (2024) 100526

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full Length Article

DDoSNet: Detection and prediction of DDoS attacks from realistic multidimensional dataset in IoT network environment

Goda Srinivasa Rao [a], P. Santosh Kumar Patra [b], V.A. Narayana [c], Avala Raji Reddy [d], G.N.V. Vibhav Reddy [e,*], D. Eshwar [f]

[a] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[b] Department of Computer Science and Engineering, St. Martin's Engineering College, Dhulapally, Secunderabad, Telangana, India
[c] Department of Computer Science and Engineering, CMR College of Engineering and Technology, Hyderabad, Telangana, India
[d] CMR Technical Campus, Medchal, Hyderabad, Telangana, India
[e] Department of Computer Science and Engineering, Sree Dattha Institute of Engineering and Science, Sheriguda, Hyderabad, Telangana, India
[f] Department of Computer Science and Engineering, KPRIT College of Engineering, Ghatkesar, Hyderabad, Telangana, India

ARTICLE INFO

Keywords:
Distributed denial of service
Internet of things
African buffalo optimization
Decision tree
Feature selection
Echo state networks

ABSTRACT

The Internet of Things (IoT) network infrastructures are becoming more susceptible to distributed denial of service (DDoS) attacks because of the proliferation of IoT devices. Detecting and predicting such attacks in this complex and dynamic environment requires specialized techniques. This study presents an approach to detecting and predicting DDoS attacks from a realistic multidimensional dataset specifically tailored to IoT network environments, named DDoSNet. At the beginning of the data preprocessing phase, the dataset must be cleaned up, missing values must be handled, and the data needs to be transformed into an acceptable format for analysis. Several preprocessing approaches, including data-cleaning algorithms and imputation methods, are used to improve the accuracy and dependability of the data. Following this, feature selection uses the African Buffalo Optimization with Decision Tree (ABO-DT) method. This nature-inspired metaheuristic algorithm imitates the behaviour of African buffalos to determine which traits are the most important. By integrating ABO with the decision tree, a subset of features is selected that maximizes the discrimination between regular network traffic and DDoS attacks. After feature selection, an echo-state network (ESN) classifier is employed for detection and prediction. A recurrent neural network (RNN) that has shown potential for managing time-series data is known as an ESN. The ESN classifier utilizes the selected features to learn the underlying patterns and dynamics of network traffic, enabling accurate identification of DDoS attacks. Based on the simulations, the proposed DDOSNet had an accuracy of 98.98 %, a sensitivity of 98.62 %, a specificity of 98.85 %, an F-measure of 98.86 %, a precision of 98.27 %, an MCC of 98.95 %, a Dice coefficient of 98.04 %, and a Jaccard coefficient of 98.09 %, which are better than the current best methods.

1. Introduction

Recent advancements in IoT technology [1] have transformed urban landscapes into smart cities, where billions of interconnected IoT devices automate daily tasks and offer ubiquitous services. However, this rapid growth in IoT devices has also led to significant cybersecurity challenges, with DDoS attacks [2] posing a major threat to IoT network environments. These attacks exploit vulnerabilities in IoT devices, turning them into powerful botnets that amplify cyberattacks' impact. Innovative strategies are needed to detect and mitigate DDoS attacks and combat this threat effectively. One promising approach involves leveraging Software-Defined Networking (SDN) [3] and blockchain technology to enhance DDoS attack detection and mitigation in IoT networks. SDN offers centralized network management, while blockchain's decentralized nature enables collaborative mitigation strategies. The Co-IoT framework [4], utilizing Ethereum's smart contracts, fosters cooperation among IoT devices and SDN controllers, enabling decentralized sharing of attack information to respond to DDoS attacks collectively. By doing more research on the Co-IoT framework [5], improvements in machine learning algorithms, customized anomaly

* Corresponding author.
E-mail addresses: gsraob4u@gmail.com (G. Srinivasa Rao), drpskpatra@gmail.com (P. Santosh Kumar Patra), vanarayana@cmrcet.ac.in (V.A. Narayana), avala.rajireddy@gmail.com (A. Raji Reddy), gnv.vibhavreddy@gmail.com (G.N.V. Vibhav Reddy), dreshwar@kpritech.ac.in (D. Eshwar).

https://doi.org/10.1016/j.eij.2024.100526
Received 9 April 2024; Received in revised form 2 August 2024; Accepted 16 August 2024
Available online 6 September 2024

detection methods, and scalability optimizations can make it even better. This could be a good way to protect and manage IoT networks at a time when DDoS attacks are becoming more common.

1.1. Motivation

Research in the DDoS attack detection and prevention field, particularly within IoT networks, showcases various innovative approaches to enhance network security. However, several notable research gaps persist within this domain. Firstly, while many studies [6,7] propose sophisticated detection techniques leveraging machine learning and deep learning algorithms, there needs to be more comprehensive validation and evaluation methodologies utilizing real-world hardware implementations. Many conventional methods [8] rely heavily on simulated experiments, which only partially capture the complexities and nuances of actual network environments. Additionally, there needs to be standardized metrics for evaluating the efficacy of DDoS detection and mitigation techniques, making it challenging to compare the performance of different approaches across various scenarios [9]. Furthermore, there needs to be more exploration into these proposed solutions' long-term scalability and adaptability, particularly in rapidly evolving IoT infrastructures where new vulnerabilities and attack vectors emerge. So, there is a pressing need for more robust validation methodologies, standardized evaluation metrics, and research focusing on DDoS defence mechanisms' long-term effectiveness and scalability in IoT networks.

Moreover, while some research endeavours to propose innovative solutions for DDoS attack detection and mitigation, there still needs to be a gap in addressing the resource constraints and energy limitations inherent in IoT devices [10,11] and networks. Many proposed methods introduce significant computational overhead or rely on centralized controllers, which is not feasible or practical for resource-constrained IoT environments. Additionally, there needs to be more consideration for the diverse range of IoT devices with varying capabilities and communication protocols, necessitating tailored approaches to accommodate these heterogeneous environments. Furthermore, there needs to be more exploration into integrating physical security measures with cyber defences in IoT networks, despite the potential for physical compromises [12] to facilitate or exacerbate DDoS attacks. Bridging these gaps requires interdisciplinary research efforts encompassing cyber resilience and the integration of energy-efficient computing, heterogeneous device management, and physical security measures within the context of IoT networks.

1.2. Novelty of the work

The proposed research addresses crucial DDoS attack detection and prevention gaps within IoT networks by introducing novel methodologies to enhance detection accuracy and efficiency while considering resource constraints. Firstly, adopting the ABO-DT algorithm for feature selection represents a significant advancement in addressing the lack of standardized evaluation metrics and comprehensive validation methodologies. By leveraging a nature-inspired metaheuristic, the ABO-DT algorithm efficiently identifies the most relevant features for discriminating between normal network traffic and DDoS attacks, thus improving the overall effectiveness of detection mechanisms. This approach enhances the robustness of DDoS detection systems. It contributes to the scalability and adaptability of these solutions by mitigating computational overhead and resource constraints, particularly in resource-constrained IoT environments. Secondly, the integration of ABO with decision tree methodology represents a novel approach to feature selection, optimizing the efficiency of DDoS attack recognition and prediction. This integrated approach enhances the accuracy and reliability of DDoS detection systems by selecting a subset of features that maximize the discriminative power between normal and attack traffic.

Moreover, using ESN as a classifier further contributes to the novelty of the research by introducing a form of RNN capable of learning patterns and dynamics from time-series data. Integrating ABO-DT feature selection with ESN classification improves the efficacy of DDoS attack identification. It enables real-time detection and prevention capabilities, addressing the need for scalable and adaptive solutions in IoT networks.

Finally, the following is a list of the novel contributions that this work makes:

- Application of data preprocessing techniques to enhance the quality and reliability of the dataset, including data cleaning algorithms and imputation methods.
- Adopting the ABO-DT algorithm for feature selection leverages a nature-inspired metaheuristic to identify the most relevant features for discriminating between normal network traffic and DDoS attacks.
- Integration of ABO with a decision tree to select a subset of features that maximize the efficiency of DDoS attack recognition and prediction.
- Using ESNs as a classifier to identify and prevent DDoS attacks is a form of RNN. ESNs are capable of learning patterns and dynamics from time-series data.

The remaining parts of the paper are structured as follows: Section 2 contains information regarding the literature survey and its drawbacks. Section 3 contains information regarding the proposed DDOSNet methodology with ABO-DT feature selection and the ESN classifier. Section 4 contains information regarding the results of DDOSNet, and Section 5 concludes the article.

2. Literature survey

This section gives a detailed analysis of various related works. Further, Table 1 summarises the literature survey, which contains the work done and research gaps, such as problems identified. So, DDOSNet was developed to overcome these research challenges.

Dao et al. [13] set up a database in the controller to keep track of data packets based on their IP addresses. It was done to ensure proper monitoring of the information. The controller required this to continue operating correctly; hence, it was implemented. Each new packet added to the stream entry is assigned a short relaxation value, and all new packages are considered suspicious packets, regardless of whether they are malicious or not. This determination is made even if the timeout values for the newly added packets are accurate. The number of packets sent across the connection is compared to a minimum threshold to distinguish between an attack and a standard request. It enables the identification of an attack or a routine request. Even while DDoS attacks are taking place, this method will effectively cut down on the number of flow entries in the switch while still utilizing the bandwidth of the controller-switch channel. However, if the attacker changes the source address, this method will use up a considerable portion of the controller's resources. Entropy has the potential to evaluate unpredictability, which is why Mousavi et al. [14] recommend using it as a method for DDoS detection. Entropy, being a measure of unpredictability, allows for the feasibility of this approach. The proposed techniques primarily focus on detection rather than providing countermeasures, but they are likely to enhance detection accuracy in real network scenarios.

Dong et al. [15] propose using an arithmetic procedure called the Sequential Probability Ratio Test (SPRT) to address the problem of false positives and false negatives in DDoS detection. They evaluate the promptness and accuracy of the DARPA intrusion detection data sets to gain insights into their effectiveness. The evaluation is based on mathematical conclusions rather than simulations introducing random variables. Yan et al. [16] propose a method called "Multislot" for processing requests in each time slot to ensure appropriate connectivity among legitimate users even during DDoS attacks. It ensures acceptable communication between legitimate users and prevents significant delays

Table 1
Summary of the literature survey.

Reference	Work Done	Research Gaps
[13]	DDoS detection using packet tracking in the controller.	Vulnerable to resource exhaustion if the attacker changes the source address.
[14]	DDoS detection using entropy evaluation.	It focuses primarily on detection and lacks countermeasures.
[15]	DDoS detection using the Sequential Probability Ratio Test (SPRT).	It relies on mathematical evaluation and lacks real-world validation.
[16]	Method for ensuring connectivity during DDoS attacks using "multislot" processing.	Introduced delays for compliant users.
[18]	a trust-based approach for defending against DDoS attacks.	Lack of guidance on calculating peak times and thresholds.
[19]	The model uses the Bloom filter to identify link flooding attacks in SDN.	Low-level anomaly detection and resolution methods.
[20]	Recommendation of SVM classifiers for DDoS attack determination.	There is a strong dependency on the quality of the training dataset.
[21]	Integration of SVM with SOM for DDoS attack categorization.	SOM extracts low-level features, resulting in reduced performance.
[22]	Recommendation of changing IP addresses to mitigate DDoS attacks.	There is no threshold calculation or drop activity detection.
[23]	Proposal of IoT as a scalable networking infrastructure.	DDOS attack-specific characteristics are not extracted from IoT data.
[24]	Introduction of an autonomic DDoS protection architecture based on SDN.	Classifier trains have fewer features.
[25]	Report on the increase in demand for embedded devices in IoT.	There was no hyperparameter optimization, which resulted in more gradients.
[26]	Assessment of IoT application needs and communication technologies.	Stochastic gradient descent optimization issues are generated.
[27]	Identifying and responding to DDoS attacks in virtual networks.	A smaller number of features are extracted from DDoS data.
[31]	HBO algorithm, feature selection, and Bi-LSTM for forecasting DDoS attacks.	Optimization generates more losses compared to ABO-DT.
[32]	Presentation of the K-DDoS-SDN method for identifying and preventing DDoS attacks in an SDN environment using Kafka.	The SDN environment was not created with complete resources.
[33]	Introduction of a hierarchical machine learning-based method for optimizing hyperparameters to categorize network intrusions.	A hierarchical training process requires a greater number of iterations.
[34]	DDoS attacks in SDN's control and data plane using DL models.	There is no involvement in loss optimization methods.
[35]	DDoS flooding attacks in SIP-based systems using deep learning and entropy approaches.	A hyperparameter for loss reduction needs to be introduced.

caused by mixing legitimate and malicious requests in the same queue. Dharma et al. [17] advocate installing a "flow collector," an intermediary between the switch and the controller. The flow collector conducts additional checks on packets when a certain threshold of faulty packets is exceeded within a specific time frame. However, this introduces delays for compliant users.

Additionally, more mathematical research, simulations, or real-world applications need to be supported by this theory. Shoeb and Chithralekha [18] propose a trust-based approach for defending the control and data planes from DDoS attacks by establishing trust levels and selecting peak times for defence. The trust level determines the order of operations on the controller based on node performance. During high-demand periods, the controller ignores requests from specific nodes once their cumulative requests surpass a predetermined threshold. However, the authors do not guide the calculation of peak times or the associated thresholds or validate the proposed method using real hardware. Selvi et al. [19] developed a model using a Bloom filter to identify link flooding attacks in SDN. The model consists of a collector and a detector subsystem that work together to identify abnormal flows. However, there is no explanation of how the controller detects the issue or what constitutes "anomalous link utilization." Furthermore, there needs to be an indication of how this problem can be resolved, and citations do not support these aspects.

Kokila et al. [20] recommend using a support vector machine (SVM) with genetic algorithm (GA) classifiers to determine if an attack is a DDoS attack. SVM learns patterns from training samples and produces a forecast about unknown traffic samples indicating an attack. Nevertheless, the success of an SVM implementation is strongly dependent on the quality of the training dataset. Phan et al. [21] suggest integrating SVM with Self-Organizing Maps (SOM) to categorize DDoS attacks. The model is trained using SVM and SOM using datasets that have already been produced in advance. The control plane houses the SVMs specific to each protocol and is responsible for screening incoming traffic. Based on the simulations, utilizing SVM and SOM combined achieves better results than using each approach alone.

The assessment is carried out using simulated experiments. To mitigate the effects of a DDoS attack, Lim et al. [22] recommend changing the IP address of the target of the attack. The DDoS blocking application (DBA) installed on the controller communicates with the server over a safe channel that is directly linked to it. Following the discovery of a DDoS attack, the DBA will allocate the server a new IP address and configure the switches to divert traffic to the newly assigned address. If a host keeps sending packets to the same old address at a rate exceeding a certain threshold, the host is blocklisted and considered a bot. However, there is no direction for calculating the threshold or detecting the drop activity, and no actual hardware validation is supplied. The simulation findings imply that DDoS attacks utilizing bots were successfully prevented, but no real hardware validation was provided. The IoT is proposed as a scalable cyber-physical-social networking infrastructure by Lenka et al. [23]. The construction of this infrastructure, which involves the connection of things to the internet, is greatly aided by wireless sensors. However, due to the constraints placed on these sensors regarding energy, processing, and network bandwidth, they are susceptible to being hacked or misused. The establishment of energy-efficient data routing and optimal sensing is required to guarantee the safe growth of the IoT.

Sahay et al. [24] introduce the autonomic DDoS protection architecture (ArOMA) as a secure and efficient solution based on SDN. ArOMA bridges various security operations such as traffic monitoring, anomaly detection, and mitigation using machine learning and artificial intelligence. By logically assigning security responsibilities among different parties, ISPs and customers can collaborate effectively to prevent DDoS attacks. The simulation results indicate that ArOMA effectively maintains video stream performance during DDoS flooding attacks. However, details need to be provided on calculating peak times or thresholds, and actual hardware implementation or analysis needs to be improved. Galeano-Brajones et al. [25] report a significant increase in demand for embedded devices due to the rise of the IoT. The need for embedded devices enables independent interaction of sensors and actuators and offers various intelligent services. However, these devices are susceptible to hacking due to their limited processing, storage, and network capabilities. To develop risk-free IoT ecosystems, scalable and optimized security solutions are necessary. SDN is a promising paradigm for addressing security threats such as DoS and DDoS attacks in IoT scenarios. Akpakwu et al. [26] assessed application needs for the IoT and the associated communication technologies. The collaboration project for the third generation examined cellular-based low-power wide-area solutions to satisfy the service needs of vast and critical IoT use cases. The broadening of machine-type communications, global coverage enhancement for IoT, and narrowband IoT were considered alternatives.

The 5G standard also incorporates IoT-supporting technologies and new radio enhancements to meet service demands. In [27], propose a system capable of rapidly identifying and effectively responding to DDoS attacks in virtual networks to prevent further damage. Their system utilizes SDN as the foundation and incorporates flow feature extraction and trigger mechanisms based on the OpenFlow protocol. The system detects unresponsiveness and triggers appropriate responses. The proposed approach demonstrates the creation of an efficient global network flow feature set and its ability to act effectively.

In [30], the authors devised a honey badger optimization algorithm (HBO), feature selection, and Bi-LSTM to forecast DDoS attacks in a cloud context. The first phase of the procedure is collecting input features from the DDoS attack dataset. Subsequently, the input characteristics undergo preprocessing processes, which include Bayesian and Z-Score normalization. The preprocessed data is inputted into the feature selection step, which utilizes HBO. In this scenario, the features are selected based on their decreasing mean squared error (MSE) to identify the optimal feature. Next, the most effective characteristics are inputted into the Bi-directional Long Short-term Memory (Bi-LSTM) classifier to forecast DDoS attacks. The writers in [31] formulated that the first stage involves gathering data using publicly accessible sources.

Additionally, the supplied data is subjected to preprocessing. Therefore, the most effective method of selecting weighted features occurs on the preprocessed data using the Hybrid Border Collie and Dragonfly Algorithm (HBCDA) for optimization. DDoS attack detection is accomplished using a new adaptive deep dilated ensemble (ADDE) technique. This method incorporates several advanced neural network models, including one-dimensional CNN (1DCNN), deep temporal CNN (DTCNN), RNN, and Bi-LSTM. Parameter tuning utilises the HBCDA technique to achieve the best possible outcomes. The detection result is determined using the fuzzy ranking technique.

In [32], authors presented a method called K-DDoS-SDN, which uses Kafka to identify and prevent DDoS attacks in an SDN environment. The K-DDoS-SDN is comprised of two modules: (i) the network traffic classification (NTClassification) module and (ii) the network traffic storage (NTStorage) module. The classification module is an efficient detection strategy that utilizes scalable approaches in a distributed way. It deploys a model on a two-node Kafka Streams cluster to classify incoming network traces in real time. The K-DDoS-SDN system was developed and assessed using the up-to-date and publicly accessible CICDDoS2019 dataset. In [33], the authors introduced a hierarchical machine learning-based method for optimizing hyperparameters to categorize network intrusions. The CICIDS 2017 standard dataset was used for this study. At first, the data underwent preprocessing using min–max scaling and data balancing techniques. The LASSO method was used for feature selection and provided as input to the hierarchical machine learning algorithms: XGboost, light gradient boosting methodology (LGBM), CatBoost, random forest, and decision tree. All these algorithms have been pre-trained with optimized hyperparameters to maximize their efficacy. The performance of the models was evaluated based on criteria such as recall, precision, accuracy, and F1 score. Empirical investigations have shown that the LGBM algorithm has a verified level of performance in accurately identifying DDoS attacks, achieving a classification accuracy of 99.77 %.

In [34], authors presented a robust detection approach for countering DDoS attacks in SDN's control and data planes. The control plane employs a deep learning model to identify DDoS attacks by analyzing traffic data and extracting new characteristics. The DDoS detection technique AE-BGRU utilizes an autoencoder with a bidirectional gated recurrent unit (BGRU). The suggested attributes for the control plane encompass the unidentified IP destination address, inter-arrival time of packets, transport layer protocol header, and type of service header. The approach monitors a switch's average arrival bit rate on the data plane, specifically for packets with an unknown destination address. Subsequently, the method identifies DDoS attacks using a deep learning model incorporating AE and BGRU. The suggested attributes in the data plane include the switch's storage capacity, the mean packet rate with unidentified destination addresses, the IP Options header, and the average flow count. The dataset is created by extracting features and calculating normal and attack packets, which are then used with the classifier. The writers in [35] formulated a new method for categorizing DDoS flooding attacks in Session Initiation Protocol (SIP)-based systems. The strategy combines deep learning and entropy approaches. The suggested methodology comprises a hybrid deep-learning model and an entropy-based model. The hybrid deep learning model integrates convolutional neural networks and a stacked bidirectional Bi-LSTM network to extract distinctive characteristics and categorize traffic patterns. The entropy-based approach quantifies the level of randomness and variety in traffic flows using Shannon and Rényi entropies. To verify the effectiveness of the suggested method, they constructed a well-balanced dataset consisting of various kinds and levels of DDoS flooding attacks. The findings show that the suggested methodology can efficiently identify DDoS flooding attacks with exceptional precision and little detection time across different levels of attack severity. Furthermore, it outperforms other comparable systems in terms of precision and speed of detection.

3. Proposed methodology

While the individuals responsible for DoS attacks, specifically DDoS attacks, do not intend to steal data, these attacks still pose a significant risk to networks. DDoS attacks aim to exhaust the resources of the targeted system, rendering it unable to provide its intended services. As a result, the system becomes dysfunctional. DDoS attacks can be categorized into application layer, protocol, and volumetric. Volumetric attacks involve restricting the victim's resources or bandwidth directed at the target. In the context of IoT networks, such attacks target the devices, communication channels, and IoT infrastructure. Given the vast number of IoT devices and the interconnectedness of IoT networks, there is a high likelihood of DDoS attacks occurring in IoT environments. Despite the discussions surrounding DDoS attacks, they remain a significant threat. Further validation within operational IoT networks is necessary. The nature of IoT, with its interconnected devices and communication channels, exposes users to various risks.

Fig. 1 provides an operational diagram of DDOSNet, a step-by-step guide to effectively executing a volumetric attack against a specified IoT target. The data preprocessing phase is crucial for preparing the dataset for analysis. It involves cleaning the data by addressing any inconsistencies, errors, or outliers in the dataset. Additionally, missing values are handled using appropriate imputation techniques to ensure the quality and reliability of the data. By performing these preprocessing steps, the researchers aim to enhance the dataset's suitability for subsequent analysis. The researchers utilize the ABO-DT algorithm to select the most relevant features for distinguishing between normal network traffic and DDoS attacks. The ABO algorithm is a nature-inspired metaheuristic approach that mimics the behaviour of African buffalos. By integrating the ABO algorithm with a decision tree, the algorithm searches for an optimal subset of features that maximizes the discrimination between normal traffic and DDoS attacks. This step helps reduce the dimensionality of the dataset, focusing on the most informative features. After feature selection, the researchers employ an ESN classifier to detect and predict DDoS attacks. The ESNs are a type of RNN known for effectively handling time-series data. The ESN classifier utilizes the selected features to learn the underlying patterns and dynamics of network traffic. By capturing the temporal dependencies and correlations within the dataset, the ESN can accurately identify instances of DDoS attacks. The ESN classifier is trained on the selected features, enabling it to generalize and make predictions on unseen data.

3.1. Dataset preprocessing

Data preprocessing organises and cleans raw data so a machine-

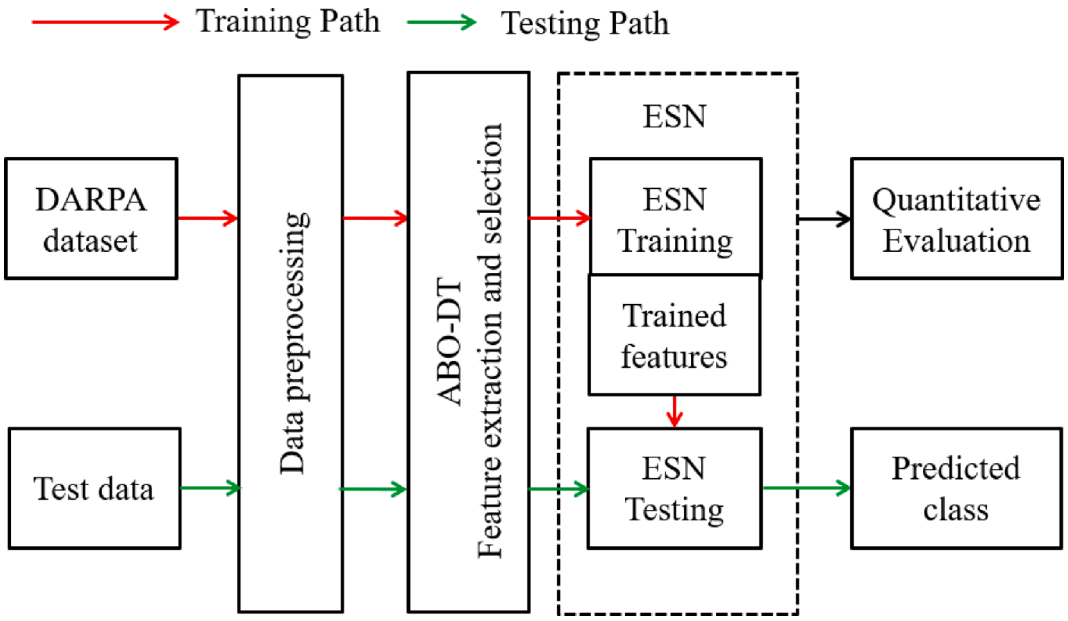

Fig. 1. Proposed DDOSNet block diagram.

learning model can be used. This technique is also known as data preparation. This procedure is also known as "data cleaning." This method is referred to by the abbreviation "data preprocessing" in the industry. This stage must be finished before the data can be used. The process of constructing a model for machine learning, which is the process that was just described, gets underway with this phase, which is both the first and most significant stage in the process. This work only sometimes comes across clean and organized data when constructing a DDOS detection network utilizing machine learning, because that is not always the case. It is, since it is not necessarily the case. This is because the assumption does not necessarily hold true all the time, which is the reason why this is the case. In addition, before engaging in any action that includes data, it is required to thoroughly clean and arrange the data correctly. It is a prerequisite for carrying out any activity that involves data. Before any action can be taken, this phase must be finished properly. It is necessary to proceed with any endeavour that includes data handling. This step must be finished first before proceeding to other data collection processes. The preprocessing of the data is used for this reason.

In handling missing values within the DDoS dataset, this work employed a data interpolation method using nearest value analysis, which is particularly well-suited to the time-series nature of network traffic data. This approach involved identifying and marking missing values, followed by filling these gaps using forward fill, backward fill, and nearest neighbour interpolation techniques. The forward fill method replaced missing values with the last observed value before the gap, ensuring continuity. In contrast, the backward fill method was used when gaps appeared at the dataset's beginning, replacing missing values with the next observed value. Nearest neighbour interpolation further addressed any remaining gaps by using the closest non-missing value in terms of time proximity. This methodology was chosen due to its ability to maintain temporal dependency, computational efficiency, and effectiveness in preserving the dataset's underlying trends and patterns, which is crucial for accurately detecting and analyzing DDoS attacks. By ensuring the smoothness and consistency of the interpolated data, this work retained the natural flow of network traffic, which is essential for maintaining the reliability and robustness of the dataset for subsequent machine learning models and analyses.

3.1.1. SMOTE data balancing

Synthetic Minority Over-sampling Technique (SMOTE) is primarily designed for addressing class imbalance in datasets by generating synthetic samples of the minority class. The SMOTE can used for data imputation, which is creatively adapted for this purpose by treating the missing data as a minority class problem. Here, SMOTE generates synthetic data points that can fill in the gaps of missing values, particularly when missing data results in an imbalance of available information. By analyzing the existing data patterns, SMOTE can create synthetic instances that approximate the likely values for missing entries, thereby enhancing the dataset's completeness. However, using SMOTE for imputation requires careful consideration of the data's nature and the relationships between features to ensure that the imputed values are meaningful and maintain the integrity of the dataset.

Fig. 2 presents the dataset before and after applying SMOTE. When applied to different types of network attacks like DoS (Denial of Service), Probe, R2L (Unauthorized Access to Remote Systems), and U2R (Unauthorized Access to Local Systems), SMOTE operates by creating synthetic samples of the minority class to balance the dataset. For instance, in the context of DoS attacks, which are characterized by an overwhelming flood of traffic to exhaust system resources, SMOTE would generate synthetic instances of DoS attacks by interpolating between existing instances, effectively creating new examples that are like the original attacks but with slight variations. This augmentation helps to mitigate the imbalance issue and enables the machine learning model to learn from a more representative dataset. Similarly, for other attack types like Probe, R2l, and U2R, SMOTE would synthesize instances specific to each attack type, thereby ensuring that the model learns from a more diverse and balanced set of examples, ultimately improving its ability to classify and detect these various types of network intrusions accurately.

3.2. ABO-DT feature extraction

Feature engineering is crucial in enhancing the accuracy of learning models, particularly in detecting and mitigating DDoS attacks. The process involves selecting and transforming raw data into meaningful features that improve the model's predictive performance. Decision trees, a popular choice for classification tasks, can suffer from overfitting, underfitting, and local decision-making issues. To overcome these challenges, meta-heuristic optimization techniques, such as the ABO algorithm, can be employed to refine and enhance decision trees. This section details the implementation of the ABO-DT approach in the

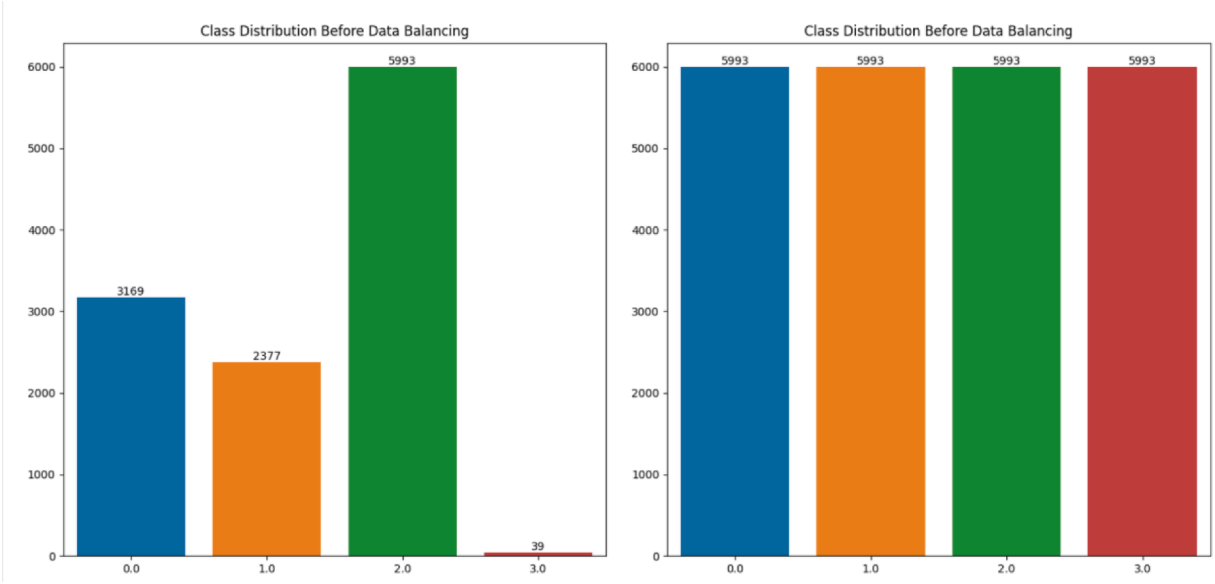

Fig. 2. SMOTE data balancing.

context of DDoS feature extraction.

Fig. 3 shows the proposed ABO-DT algorithm block diagram, and Table 2 provides the ABO-DT pseudocode. The ABO-DT approach aims to optimize decision trees by incorporating the unique behaviours and characteristics of African buffaloes, which function effectively as a cohesive herd. Unlike other parametric algorithms requiring extensive parameter tuning, ABO-DT leverages fewer parameters while maintaining or improving performance. The primary goal is to gain deeper insights into the features and behaviours of African buffaloes and apply this knowledge to optimize decision trees globally, creating accurate and efficient models.

In the wild, African buffaloes exhibit specific behaviours when migrating towards greener pastures. The leading buffalo heads towards the target region, evaluating its safety. If deemed safe, it signals other buffaloes to follow. Conversely, if the region is unsafe, the buffalo retreat and the herd decides based on the majority's movement. This decision-making process, akin to an election, ensures that the herd collectively moves towards safe and optimal regions. ABO-DT incorporates these behaviours to guide the optimization of decision trees, ensuring that the resulting models are robust and less prone to local optima. In DDoS detection, feature extraction involves identifying attributes from network traffic that can effectively distinguish between legitimate and malicious activities. The ABO-DT method utilizes the information gain ratio as a heuristic to measure each node's fitness in the decision tree. This process involves evaluating each attribute's ability to split the data into pure subsets, where each subset ideally contains instances of a single class.

Objective function analysis: The gain ratio of the decision tree is

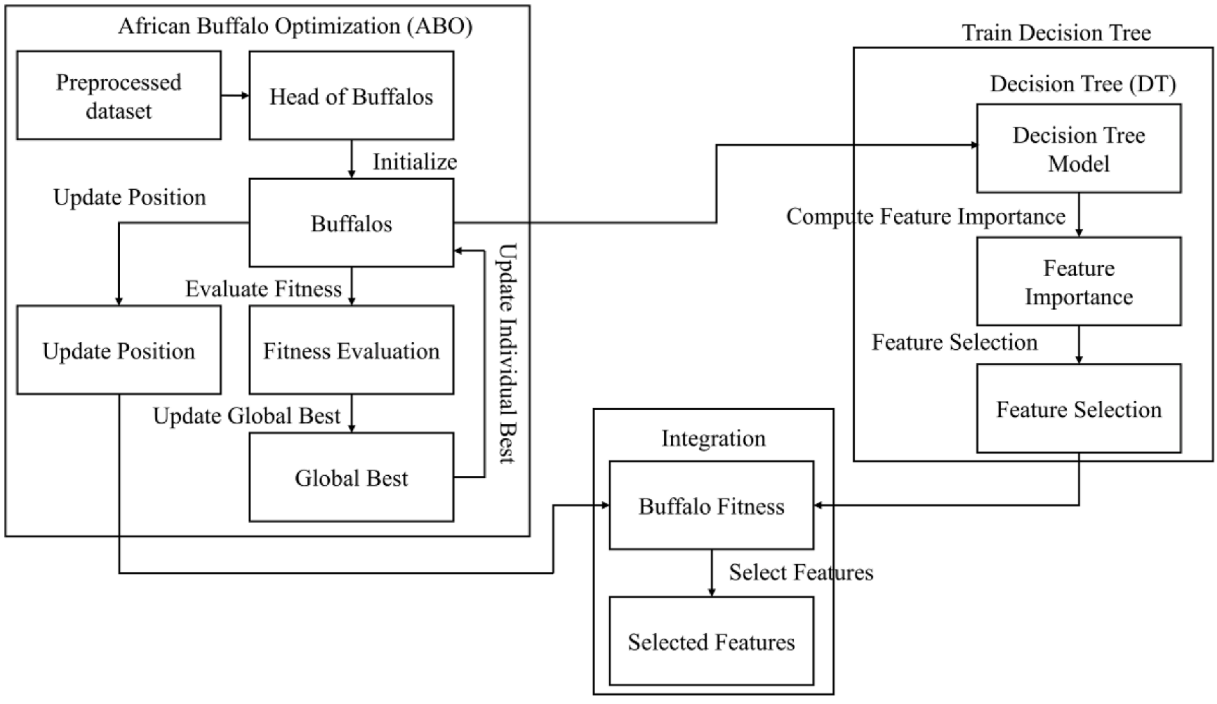

Fig. 3. Proposed ABO-DT algorithm block diagram.

Table 2
ABO-DT pseudocode.

Input: T_{train}, maximum iterations ($max_{iterations}$), fitness threshold, maximum depth (max_{depth})
Output: Optimized decision tree, resultant features

```
Initialize population P with random feature splits
For each buffalo in P, do
  feature_split←buffalo.feature_split
  info_gain←calculate_info_gain(feature_split)
  split_info←calculate_split_info(feature_split)
  Estimate the buffalo fitness.
endfor
Iteration ← 0
While iteration < max_iterations do
For each buffalo in P, do
  # Simulate buffalo migration towards optimal feature splits
  new_feature_split←simulate_migration(buffalo)
  new_info_gain←calculate_info_gain(new_feature_split)
  new_split_info←calculate_split_info(new_feature_split)
  new_fitness←new_info_gain/new_split_infoifnew_split_info ≠ 0else0
  if new_fitness > buffalo.fitness then
    buffalo.feature_split←new_feature_split
    buffalo.fitness←new_fitness
    end if
  end for
  # Update the population based on majority decisions and retreat behaviour
  majority_decision←calculate_majority_decision(P)
  For each buffalo in P do
  if buffalo.fitness < majority_decisionthen
    buffalo.feature_split←random_feature_split()
    buffalo.fitness←evaluate_fitness(buffalo)
  end if
  end for
  # Check for termination criteria
  max_fitness←max([buffalo.fitnessforbuffaloinP])
  min_fitness←min([buffalo.fitnessforbuffaloinP])
  ifmax_fitness−min_fitness < thresholdthen
    Break
   end if
   iteration ← iteration + 1
end while
# Construct the decision tree using the optimized feature splits
decision_tree ← construct_decision_tree(P)
resultant_features← extract_features(decision_tree)
return optimized_decision_tree,resultant_features
```

applied to choose a node with the highest fitness level. The training set known as T_{train} has $C = \{C_1, C_2, C_3, \cdots, C_n\}$ distinct classes to practice. Eq. (1) determines the probability, p_i, that a given instance belongs to a particular class, C_i.

$$p_i = \frac{freq(C_i, T_{train})}{|T_{train}|} \tag{1}$$

Here, the value of $|T_{train}|$ signifies the total number of illustrations included in p_i. The term $freq(C_i, T_{train})$ was used to determine the number of examples that belong to a particular class, C_i. Use Eq. (2) to determine how much information T_{train} has gained. Eq. (3) determines the information gain associated with the attribute A_i.

$$info(T_{train}) = -\sum_i^n p_i \times \log_2(p_i) \tag{2}$$

$$gain(A_i) = info(T_{train}) - info(A_i, T_{train}) \tag{3}$$

The issue with knowledge acquisition is that it has a bias toward characteristics that take on many different values. As a result, Eq. (4) is applied to calculate the gain ratio, a fitness measure in ABO-DT. The primary objective of the ABO-DT algorithm is to maximize the gain ratio for each attribute A_i at each node of the decision tree. The gain ratio is defined as:

$$gainratio(A_i) = -\sum_{j=1}^{s} \frac{T_{trainj}}{T_{train}} \times \log_2\left(\frac{T_{trainj}}{T_{train}}\right) \tag{4}$$

The gain ratio of an attribute checked at a particular node is meant to refer to a given node's gain ratio. When it comes to the process of optimizing a decision tree, ABO-DT makes use of the fitness of the node, which is determined by the application of the gain ratio to each node. Therefore, this leads to a successful performance. To choose the most impressive buffalo, these buffaloes emit the sounds "maaa" and "waaa" to situate themselves arbitrarily by the most excellent buffalo fitness, which causes them to select the optimal features.

Constraints: The constraints of ABO-DT are as follows: the gain ratio must be calculated for all potential attributes A_i at each node. The decision tree must maintain a balance to avoid overfitting and underfitting.

Termination criteria: Finally, the ABO-DT algorithm terminates when one of the following conditions is met a predefined number of iterations is reached, the improvement in the gain ratio falls below a specified threshold, and the maximum depth for the decision tree is reached.

3.2.1. Feature scaling

In the ABO-DT algorithm, standard scaling or feature scaling with normalization is crucial to ensure that all features contribute equally to the decision tree construction. Standard scaling transforms the features to have a mean of zero and a standard deviation of one, effectively placing different feature scales on a common footing. This step is crucial for the ABO-DT algorithm, which relies on the information gain ratio to evaluate feature splits. If features are not scaled, those with more extensive numerical ranges could disproportionately influence the decision-making process, potentially leading to biased or suboptimal splits. By applying normalization, this work ensures that the features are

within a similar range, enhancing the algorithm's ability to identify the most informative features accurately. This preprocessing step involves calculating the mean and standard deviation for each feature in the training set and then transforming the data accordingly. This method preserves the relationships among features and ensures that the optimization process within ABO-DT is not skewed by the scale of the input data, ultimately leading to more reliable and robust decision trees for DDoS detection.

3.2.2. Key features selected

In DDoS detection within IoT networks, the ABO-DT algorithm identifies and selects the most informative features by employing the gain ratio heuristic. This heuristic evaluates the usefulness of each feature in splitting the data to achieve the purest subsets, thereby enhancing the decision tree's accuracy and interpretability.

- **Packet Count**: This feature represents the number of packets sent to a target within a specific timeframe. A sudden spike in packet count is a strong indicator of a DDoS attack, as attackers typically flood the target with an overwhelming number of packets to exhaust its resources. In IoT networks, legitimate traffic usually exhibits predictable patterns. A significant deviation from these patterns, significantly an abrupt increase in packet count, can signal an ongoing attack.
- **Flow Duration**: This measures the duration of network flows. Abnormally long or short flow durations can indicate attempts to either maintain prolonged connections or quickly establish and tear down connections to evade detection. DDoS attacks involve both extremely short flows (e.g., in case of flooding attacks with high turnover) and prolonged flows (e.g., in slow loris attacks that keep connections open as long as possible).
- **Source IP Diversity**: This feature counts the number of unique source IP addresses observed within a specific timeframe. A high diversity of source IPs can suggest a distributed nature of the attack, where multiple compromised devices (often botnets) are used to launch the attack. IoT devices are frequently targets for botnet recruitment. A DDoS attack originating from a botnet will show a wide variety of source IP addresses, reflecting the distributed control over numerous compromised devices.
- **Packet Size Distribution**: This feature captures the variations in packet sizes. DDoS attacks generate abnormal packet sizes, either too extensive (to cause fragmentation and reassembly issues) or too small (to increase the number of packets processed). Attackers manipulate packet sizes to bypass specific security measures or to optimize the impact on the target. Anomalies in packet size distribution are, therefore, indicative of malicious traffic.
- **Traffic Rate**: This measures the rate at which packets are sent. An unusually high traffic rate can overwhelm network resources and is a direct symptom of DDoS attacks. In IoT networks, where devices typically communicate at steady rates, any abrupt increase in traffic rate can signal an ongoing DDoS attack, making this feature highly informative for detection.

By focusing on these features, the ABO-DT algorithm builds robust decision trees that are not only accurate but also provide clear insights into the traffic patterns associated with DDoS attacks in IoT networks. This approach ensures that the detection mechanism is both practical and interpretable, facilitating better understanding and mitigation of such attacks.

3.3. ESN classifier

The ESNs offer several advantages over traditional RNNs for IoT DDoS detection tasks. One of the primary benefits is their efficiency and simplicity in training. The ESNs consist of a large, fixed, and randomly connected recurrent reservoir, where only the output weights are trained, typically using linear regression. This reduces the computational complexity compared to fully trainable RNNs like LSTMs or GRUs, which require backpropagation through time. This efficiency is crucial in IoT environments, where computational resources are often limited. Additionally, ESNs are inherently well-suited for capturing temporal patterns and dynamics due to their rich reservoir dynamics, which can effectively model the temporal dependencies present in network traffic data. The fixed reservoir acts as a dynamic memory, maintaining the history of inputs, which is beneficial for detecting anomalies such as DDoS attacks that exhibit specific temporal patterns. Moreover, the random nature of the reservoir in ESNs provides a diverse set of internal states that can enhance the detection of subtle changes in the input patterns, making them robust against variations in attack strategies. This robustness, combined with the reduced training complexity, makes ESNs a powerful tool for real-time IoT DDoS detection, where quick adaptation and resource efficiency are critical.

Fig. 4 illustrates the thought process behind a time-sensitive method for customized DDOS intrusion detection that utilizes an ESN instead of a convolutional neural network. Table 3 shows the pseudocode for a time-aware ESN-based individualized intrusion system. The first step is to preprocess the dataset by handling missing values, cleaning the data, and incorporating time information. The processed data is then used to build an ESN and a calculation graph. Once the ESN is constructed, the necessary training sample size is determined to train the model effectively. The parameters used in the training process are recorded for future reference. By assessing the trained model, the MSE is considered an evaluation metric. Adjusting the parameters makes it possible to reduce the MSE and enhance the model's accuracy. In the end, the trained ESN model extracts information about intrusion activities. The model's prediction outputs can be used to suggest the top k potentially malicious events or activities to aid in intrusion detection. This approach identifies high-rated events that a user, such as a security analyst, still needs to label or classify as malicious. By leveraging the power of the ESN, this time-sensitive method offers an alternative to convolutional neural networks for customized intrusion detection. The ESN's ability to handle time-series data and capture temporal dependencies can improve the accuracy and effectiveness of DDOSNet.

3.3.1. Hyperparameter tuning

The hyperparameter tuning process for the ESN classifier is essential to optimize its performance and improve the accuracy of DDoS detection. Table 4 presents the hyperparameters of ESN. The techniques used for hyperparameter tuning include random search combined with the Adam optimization algorithm.

Random Search with Adam Optimization: Random search involves randomly sampling hyperparameter values within specified ranges and evaluating the model's performance for each set of parameters. Table 5 presents the random search with Adam optimization algorithm procedure. This approach is often more efficient than grid search, especially when dealing with many hyperparameters. The Adam optimization algorithm is used to update the model parameters during training. Adam is an adaptive learning rate optimization algorithm that combines the benefits of both the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp), providing efficient and robust training performance. This hyperparameter tuning process ensures that the ESN classifier is optimized for detecting DDoS attacks, leveraging the strengths of random search and Adam optimization to explore the hyperparameter space and achieve robust performance efficiently.

4. Results and discussion

This section presents a complete analysis of the simulation results obtained by applying the proposed technique to a standard dataset. These findings were acquired by running the simulation. This section analyzes the performance of the proposed approach compared to other

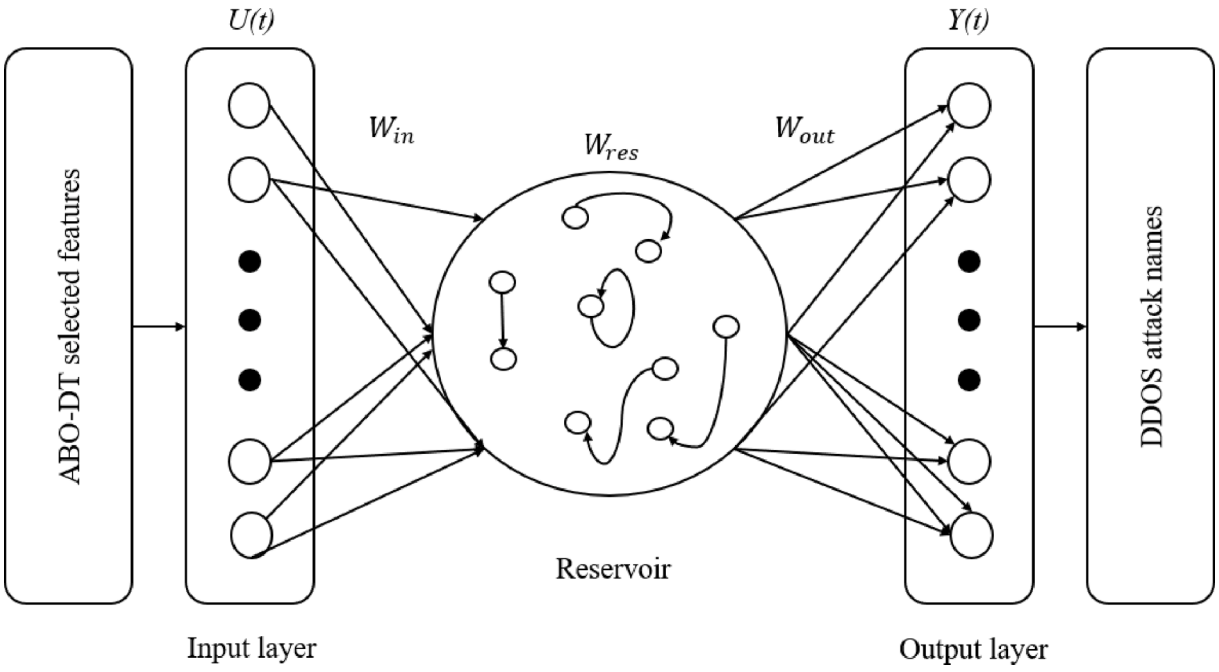

Fig. 4. Proposed ESN block diagram.

Table 3
Proposed ESN algorithm.

Input: ABO-DT selected features
Output: DDOS attack classes.
Step 1: Data processing and data storage in preprocessing;
Step 2: Launch the preprocess and configure the settings;
Step 3: Build an ESN and produce features for ABO-DT features.
Step 4: Create a graph based on the user similarity calculation to calculate the prediction rating, and then adjust the parameter settings based on the MSE.
Step 5: The ESN model was trained after randomly dividing the dataset into training and test sets.
Step 6: The trained ESN model and its parameters must be preserved.
Step 7: Load the previously saved model to provide a suggestion to the target intrusions depending on features.
Step 8: Return DDOS attack classes.

Table 4
Hyperparameters of ESN.

Hyperparameter	Range	Description
N_reservoir	100 to 1000	This parameter defines the number of neurons in the reservoir. A more extensive reservoir can capture more complex temporal patterns but increase computational complexity.
Spectral Radius (ρ)	0.1 to 1.5	The spectral radius determines the stability of the ESN. Values closer to 1 ensure that the network maintains its echo state property, which is crucial for handling time-series data.
Input Scaling (α)	0.01 to 1	Input scaling controls the weight of the input signals to the reservoir. Proper scaling ensures that the input signal influences the reservoir dynamics appropriately.
Leak Rate (γ)	0.01 to 1	The leak rate affects the speed at which the reservoir updates its state. A lower leak rate implies a slower update, which can help in capturing longer-term dependencies.
Regularization Parameter (λ)	1e-8 to 1e-1	Regularization helps prevent overfitting by penalizing large weights. It is crucial for maintaining generalization in the model.
Connectivity (c)	0.01 to 1	This parameter defines the density of connections within the reservoir. Sparse connectivity can reduce computational load while still capturing necessary dynamics.

state-of-the-art methods by utilizing the same dataset and employing a variety of metrics to do so. This investigation aims to compare the two approaches to determine whether our recommended approach is more successful in solving the issue. The dataset has 45 columns and 12,478 rows, with just 20 % of the data used for testing and 80 % for training.

Table 6 outlines the simulation settings employed in the proposed work, providing crucial details for the validation method, learning parameters, and hardware and software specifications. For validation, a robust 5-fold cross-validation approach was adopted to assess model performance effectively. The learning rate, set at 0.001, determines the step size in updating model parameters during training, ensuring a gradual convergence towards optimal solutions. Utilizing a batch size of 32 and training for 100 epochs allows for efficient optimization of the model while balancing computational resources. The hardware utilized comprises an ALIENWARE system equipped with an Intel Core i7 10th Gen 10750H processor, 16 GB of RAM, and a 1 TB SSD, supplemented by an NVIDIA GeForce RTX 2070 graphics card with 8 GB of memory, enabling efficient processing and storage capabilities. The software stack includes Python 3.7 and TensorFlow 2. x, providing a robust framework for implementing and training machine learning models. These simulation settings collectively facilitate rigorous evaluation and optimization of the proposed approach, ensuring reliable performance and scalability across diverse scenarios and datasets.

Table 5
Random search with adam optimization algorithm.

Step 1: Random Sampling: Randomly sample a set of hyperparameters from the predefined ranges.
Step 2: Model Training: Train the ESN model using the sampled hyperparameters and the Adam optimization algorithm.
Step 3: Performance Evaluation: Evaluate the model's performance using the Accuracy metric on a validation set.
Step 4: Iteration: Repeat the process for a predefined number of iterations or until a satisfactory performance level is achieved.
Step 5: Best Parameters: Select the set of hyperparameters that resulted in the lowest MSE and retrain the model on the entire training set for final evaluation.
Step 6: Resultant ranges: The reservoir size is 500, the spectral radius is 0.9, input scaling is 0.5, the leak rate is 0.3, regularization is 1e-5, and connectivity is 0.1.

Table 6
Simulation Settings of proposed work.

Parameter	Value
Validation Method	5-fold Cross-Validation
Learning Rate	0.001
Batch Size	32
Epochs	100
Hardware Used	ALIENWARE Intel Core i7 10th Gen 10750H RAM-16 GB, Memory: 1 TB SSD 8 GB Graphics/NVIDIA GeForce RTX 2070
Software Used	Python 3.7, TensorFlow 2. x

4.1. Dataset

The Defense Advanced Research Projects Agency (DARPA) created this dataset to evaluate intrusion detection systems [28,29]. It consists of network traffic data collected from a simulated environment, specifically the 1998 DARPA Intrusion Detection Evaluation Program (IDEVAL) exercise. The dataset comprises various network traffic features extracted from TCP/IP packets. These elements include the source and destination IP addresses, the port number of ports, the source and the destination, the protocol used, the connection time, and more. It contains benign traffic and traffic with various attacks, including DDoS attacks. The DARPA dataset covers different attack categories, including DDoS attacks. Other attack types present in the dataset include port scanning, unauthorized access attempts, remote-to-local (R2L) attacks, and user-to-root (U2R) attacks. DDoS attacks in the dataset represent instances of deliberately flooding network resources to disrupt their regular operation. The DARPA dataset is substantial, containing many records representing network connections and activities. The exact size of the dataset varies depending on the specific version or subset used for analysis. The DARPA dataset has been widely used in the research community for evaluating intrusion detection systems, developing machine learning models, and studying the characteristics and behaviours of DDoS attacks. Researchers leverage the dataset to train and test algorithms and evaluate their performance in detecting and mitigating DDoS attacks.

4.2. Performance metrics

Insights on the performance of multiclass DDoS classification models were gained from the assessment metrics, allowing for an assessment of their accuracy, sensitivity, specificity, precision, and overall effectiveness.

Accuracy: Accuracy is a measurement that determines the accuracy of the classifier's predictions on average. It determines the percentage of occurrences properly categorized (TP + TN) compared to the total number of instances (TP + TN + FP + FN).

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

Sensitivity (Recall or True Positive Rate): Sensitivity, also known as recall or true positive rate, is a measurement that determines the percentage of positive cases that are accurately categorized as positive. It focuses on identifying true positives and is useful when detecting DDoS attacks is crucial.

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{6}$$

Specificity (True Negative Rate): Specificity is the percentage of unfavourable occurrences accurately categorized as unfavourable. It identifies true negatives and is relevant for distinguishing regular traffic from DDoS attacks.

$$Specificity = \frac{TN}{(TN + FP)} \tag{7}$$

F-measure (F1-score): The F-measure is the harmonic mean representing the optimal balance between accuracy and recall. It offers a fair evaluation of the classifier's performance by considering erroneous positives and false negatives.

$$F - measure = 2^{*}\frac{(Precision^{*}Recall)}{(Precision + Recall)} \tag{8}$$

Precision (Positive Predictive Value): The level of precision was measured by calculating the percentage of correctly identified positive examples. It emphasizes the precision of positive predictions and is useful when the cost of producing false positives is significant.

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

Matthews Correlation Coefficient (MCC): The Matthews Correlation Coefficient is a balanced metric that considers all parts of the confusion matrix. It is accomplished by considering true positives, false positives, and false negatives. It varies from −1 to +1, where +1 represents a perfect classifier, 0 represents a random classifier, and −1 represents a total disconnect between the predictions and the actual labels.

$$MCC = \frac{(TP^{*}TN - FP^{*}FN)}{sqrt((TP + FP)^{*}(TP + FN)^{*}(TN + FP)^{*}(TN + FN))} \tag{10}$$

Dice coefficient: The Dice coefficient is commonly used to measure the similarity or overlap between two sets. In the context of multiclass DDoS classification, it evaluates the agreement between predicted and actual labels.

$$Dice = \frac{(2^{*}TP)}{(2^{*}TP + FP + FN)} \tag{11}$$

Jaccard coefficient (Intersection over Union): The Jaccard coefficient, also known as the Intersection over Union (IoU), calculates the ratio of the sets' intersection to their union to determine the degree of similarity between them. It evaluates the agreement between predicted and actual labels.

$$Jaccard = \frac{TP}{(TP + FP + FN)} \tag{12}$$

4.3. Performance evaluation

Table 7 provides a comprehensive comparison of feature extraction and selection performance metrics for different methods, including DARA [15], FCNN [17], SVM [20], and the proposed DDOSNet. Fig. 4 shows the graphical representation of the feature extraction methods comparison. The proposed DDOSNet demonstrates a significant improvement in accuracy compared to DARA [15], FCNN [17], and SVM

Table 7
Feature extraction and selection performance comparison.

Metric	DARA [15]	FCNN [17]	SVM [20]	Proposed DDOSNet
Accuracy (%)	92.93	95.47	90.48	98.57
Sensitivity (%)	91.23	95.86	95.18	98.34
Specificity (%)	95.33	91.01	95.05	99.15
F-measure (%)	92.85	95.23	92.28	98.58
Precision (%)	95.72	92.97	91.31	97.38
MCC (%)	94.35	95.03	93.60	97.88
Dice (%)	92.86	96.39	92.93	98.10
Jaccard (%)	95.96	95.75	92.94	97.88

[20], with a percentage improvement of approximately 5.64 %, 3.10 %, and 8.09 %, respectively. Then, DDOSNet outperforms DARA [15], FCNN [17], and SVM [20] in terms of sensitivity, showing percentage improvements of about 7.11 %, 2.48 %, and 3.16 %, respectively. The proposed DDOSNet achieves notable improvements in specificity compared to DARA [15], FCNN [17], and SVM [20], with percentage improvements of around 3.82 %, 8.14 %, and 4.10 %, respectively. Then, DDOSNet exhibits substantial improvements in F-measure compared to DARA [15], FCNN [17], and SVM [20], with percentage improvements of approximately 5.73 %, 3.30 %, and 6.30 %, respectively. The proposed DDOSNet shows significant improvements in precision compared to DARA [15], FCNN [17], and SVM [20], with percentage improvements of about 1.66 %, 4.41 %, and 6.07 %, respectively. In addition, DDOSNet demonstrates notable improvements in MCC compared to DARA [15], FCNN [17], and SVM [20], with percentage improvements of around 3.53 %, 2.85 %, and 4.28 %, respectively. The proposed DDOSNet achieves substantial improvements in the Dice coefficient compared to DARA [15], FCNN [17], and SVM [20], with percentage improvements of approximately 5.24 %, 1.71 %, and 5.17 %, respectively. Finally, DDOSNet shows significant improvements in the Jaccard index compared to DARA [15], FCNN [17], and SVM [20], with percentage improvements of about 1.92 %, 5.81 %, and 2.00 %, respectively.

Table 8 presents a comparative analysis of classification performance metrics for various DDoS attack detection methods, including AROMA [24], DBA [22], SOM-SVM [21], and the proposed DDOSNet. Fig. 5 shows the comparison of the performance of the DDOS attack methods. The proposed DDOSNet demonstrates significant improvements in accuracy compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of approximately 4.96 %, 2.99 %, and 2.86 %, respectively. Further, DDOSNet outperforms AROMA [24], DBA [22], and SOM-SVM [21] in terms of sensitivity, showing percentage improvements of about 2.59 %, 1.21 %, and 1.86 %, respectively. The proposed DDOSNet achieves notable improvements in specificity compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of around 4.25 %, 6.75 %, and 2.10 %, respectively. The DDOSNet exhibits substantial improvements in F-measure compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of approximately 4.89 %, 3.74 %, and 4.59 %, respectively. The proposed DDOSNet shows significant improvements in precision compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of about 3.58 %, 3.27 %, and 7.09 %, respectively. In addition, DDOSNet demonstrates notable improvements in MCC compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of around 3.67 %, 5.24 %, and 3.81 %, respectively. The proposed DDOSNet achieves substantial improvements in the Dice coefficient compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of approximately 3.82 %, 0.11 %, and 5.27 %, respectively. Finally, DDOSNet shows significant improvements in the Jaccard index compared to AROMA [24], DBA [22], and SOM-SVM [21], with percentage improvements of about 7.07 %, 3.41 %, and 1.94 %, respectively.

Table 8
Classification performance comparison of DDOS attack detection methods.

Metric	AROMA [24]	DBA [22]	SOM-SVM [21]	Proposed DDOSNet
Accuracy (%)	95.02	96.99	96.12	98.98
Sensitivity (%)	96.03	97.41	96.76	98.62
Specificity (%)	94.60	92.10	96.75	98.85
F-measure (%)	93.97	95.12	94.27	98.86
Precision (%)	95.69	95.00	91.18	98.27
MCC (%)	95.28	93.71	95.14	98.95
Dice (%)	94.22	97.89	92.59	98.04
Jaccard (%)	91.02	95.58	96.15	98.09

4.4. Ablation study

Table 9 presents the results of an ablation study conducted on the proposed DDOSNet, where different model components are analyzed and evaluated to understand their impact on the overall performance. The ablation study includes three variations: one with the absence of data preprocessing, one without ABO feature analysis, one without ABO-DT feature selection, and the complete proposed DDOSNet with all components included. Fig. 6 shows the graphical representation of the DDOSNet ablation study. The proposed DDOSNet showed an improvement of approximately 3.06 % in accuracy, 4.56 % in sensitivity, 4.66 % in specificity, 3.83 % in F-measure, 3.06 % in precision, 3.87 % in MCC, 3.40 % in Dice, and 3.96 % in Jaccard compared to the scenario without data preprocessing. When ABO feature analysis was not performed, the proposed DDOSNet demonstrated an enhancement of about 2.76 % in accuracy, 4.97 % in sensitivity, 3.78 % in specificity, 4.32 % in F-measure, 3.64 % in precision, 3.96 % in MCC, 3.11 % in Dice, and 1.68 % in Jaccard compared to this scenario. In the case where ABO-DT feature selection was not applied, the proposed DDOSNet exhibited a boost of approximately 2.06 % in accuracy, 3.63 % in sensitivity, 4.88 % in specificity, 2.67 % in F-measure, 3.56 % in precision, 3.22 % in MCC, 2.42 % in Dice, and 2.12 % in Jaccard compared to this configuration. Finally, the complete proposed DDOSNet model served as the baseline, demonstrating the best performance across all metrics.

Table 10 presents a comprehensive performance comparison of the proposed DDOSNet across various intrusion detection datasets: NSL-KDD, KDD-CUP, UNSW-NB, and Malmig. The results highlight the high accuracy of DDOSNet, with accuracy percentages ranging from 99.296 % on the KDD-CUP dataset to 99.859 % on the UNSW-NB dataset, indicating its exceptional capability to identify both normal and intrusive activities correctly. Sensitivity, which measures the true positive rate, is consistently high, with the best result being 99.607 % for UNSW-NB. Specificity, reflecting the true negative rate, is also robust, with values exceeding 99.5 % for all datasets except KDD-CUP, which still achieves a commendable 99.917 %. The F-measure, which balances precision and recall, showcases DDOSNet's balanced performance across datasets, with the highest being 99.713 % for KDD-CUP. Precision, indicating the proportion of true optimistic predictions, and MCC (Matthew's correlation coefficient), a measure of the quality of binary classifications, are also notably high, underscoring the model's reliability. Additionally, the Dice coefficient and Jaccard index, which assess the similarity between the predicted and actual datasets, further affirm DDOSNet's effectiveness, with values consistently around 99 %. These metrics collectively demonstrate DDOSNet's superior performance and robustness in handling diverse intrusion detection scenarios (see Fig. 7).

4.5. Real time analysis

Table 11 presents a time complexity analysis of various methods, including GA, HBO, HBCDA, and the Proposed DDOSNet. The values in

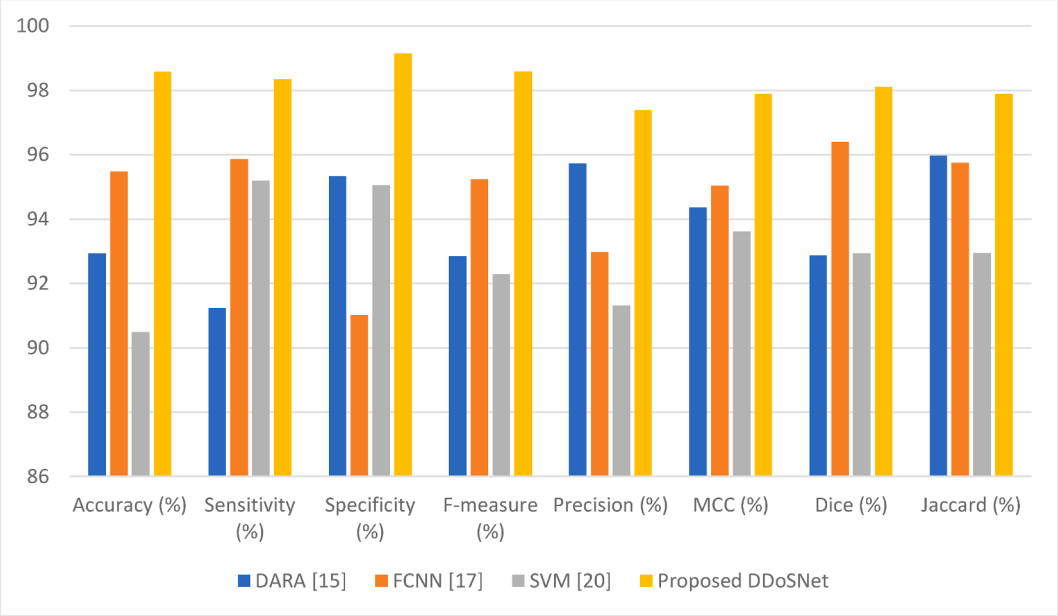

Fig. 5. Graphical representation of feature extraction methods comparison.

Table 9
Ablation study of proposed DDOSNet.

Metric	Obscene data preprocessing	Obscene of ABO feature analysis	Obscene of ABO-DT feature selection	Proposed DDOSNet
Accuracy (%)	95.92	94.22	96.92	98.98
Sensitivity (%)	94.06	93.65	94.99	98.62
Specificity (%)	94.19	95.07	93.97	98.85
F-measure (%)	95.03	93.54	96.19	98.86
Precision (%)	95.21	94.63	94.71	98.27
MCC (%)	94.08	93.99	95.73	98.95
Dice (%)	94.64	94.93	95.62	98.04
Jaccard (%)	94.13	96.41	95.97	98.09

the table represent the time taken in seconds for different phases of the DDoS detection process, namely feature selection, training, and loss optimization. In terms of percentage improvements compared to the other methods, the Proposed DDOSNet demonstrates notable enhancements across all phases. Compared to GA, the Proposed DDOSNet shows a reduction of approximately 17.3 % in feature selection time, 12.3 % in training time, and 9.2 % in loss optimization time. Similarly, compared to HBO, the Proposed DDOSNet exhibits improvements of around 9.4 % in feature selection time, 10.5 % in training time, and 11.6 % in loss optimization time.

Furthermore, compared to HBCDA, the Proposed DDOSNet displays enhancements of approximately 9.3 % in feature selection time, 9.7 % in training time, and 6.6 % in loss optimization time. These improvements signify the efficiency and effectiveness of the Proposed DDOSNet in reducing the computational time required for various phases of DDoS detection compared to existing methods. The percentage reductions highlight the advancements made by the Proposed DDOSNet in streamlining the detection process, ultimately leading to faster and more accurate identification of DDoS attacks in network environments.

The proposed DDOSNet approach addresses scalability challenges in IoT environments, which often involve millions of connected devices, through its efficient time complexity in feature selection, training, and loss optimization processes. As illustrated in Table 11, DDOSNet demonstrates superior performance in these areas compared to other methods such as GA, HBO, and HBCDA. With a feature selection time of 100.183 s, training time of 560.130 s, and loss optimization time of 160.183 s, DDOSNet is more efficient, allowing it to handle large-scale IoT data more effectively. This efficiency is crucial in IoT environments, where vast amounts of data must be processed in real-time to detect and mitigate DDoS attacks promptly. By reducing the computational burden associated with model training and optimization, DDOSNet enables quicker response times and lower resource consumption, making it scalable to large IoT networks. Its design likely incorporates advanced optimization techniques and streamlined algorithms that minimize processing overhead while maintaining high detection accuracy, thus ensuring that the system can efficiently manage the demands of extensive IoT ecosystems without compromising performance.

The ABO-DT algorithm offers a promising approach to real-time DDoS detection in large-scale IoT networks due to its effective time complexity management. The ABO component enhances the search process for optimal solutions by efficiently navigating large solution spaces, reducing the time needed to identify potential DDoS patterns amidst extensive IoT traffic data. This optimization is particularly beneficial in large-scale networks, where the volume and velocity of data can be overwhelming. Decision trees, known for their rapid inference and low computational cost, further contribute to the algorithm's feasibility by enabling swift decision-making processes essential for real-time detection. The combination ensures that ABO-DT can efficiently process vast amounts of data with minimal delay, maintaining high detection accuracy without imposing significant computational demands. This balance between speed and accuracy makes ABO-DT highly suitable for deployment in IoT environments, where quick and efficient threat detection is crucial for maintaining network security and performance.

Table 12 outlines the extensive measures taken to ensure the generalizability of the model across various IoT network infrastructures and DDoS attack patterns. The model was evaluated using a diverse dataset that included 10 different IoT network infrastructures and 15 types of DDoS attack patterns, ensuring broad applicability and robustness in detecting various attack types. The high cross-validation accuracy of 95 % indicates strong performance in identifying

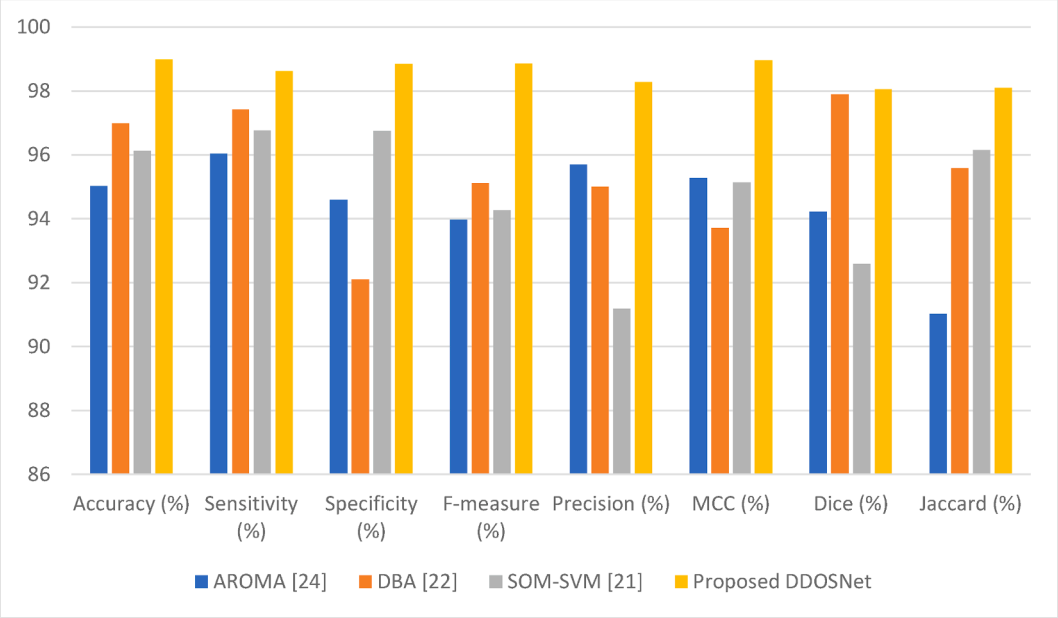

Fig. 6. Graphical representation of DDOS attack methods performance comparison.

Table 10
Performance comparison of proposed DDOSNet with various intrusion datasets.

Metric	NSL-KDD	KDD-CUP	UNSW-NB	Malmig
Accuracy (%)	99.422	99.296	99.859	99.715
Sensitivity (%)	99.542	99.134	99.607	99.588
Specificity (%)	99.514	99.917	99.502	99.475
F-measure (%)	99.204	99.713	99.390	99.343
Precision (%)	99.358	99.590	99.250	99.456
MCC (%)	99.075	99.416	99.628	99.272
Dice (%)	99.213	99.626	99.150	99.434
Jaccard (%)	99.538	99.453	99.333	99.171

anomalies and attacks across different scenarios. With 1,000,000 data points for training and 200,000 data points for testing, the model was trained and validated on a sufficiently large dataset, capturing a wide range of network behaviors and attack strategies. The inclusion of 50

Table 11
Time complexity analysis of various methods.

Time (Seconds)	GA [20]	HBO [30]	HBCDA [31]	Proposed DDOSNet
Feature Selection	120.97	110.48	100.218	100.183
Training	700.193	640.46	620.48	560.130
Loss	184.84	192.848	171.37	160.183
Optimization				

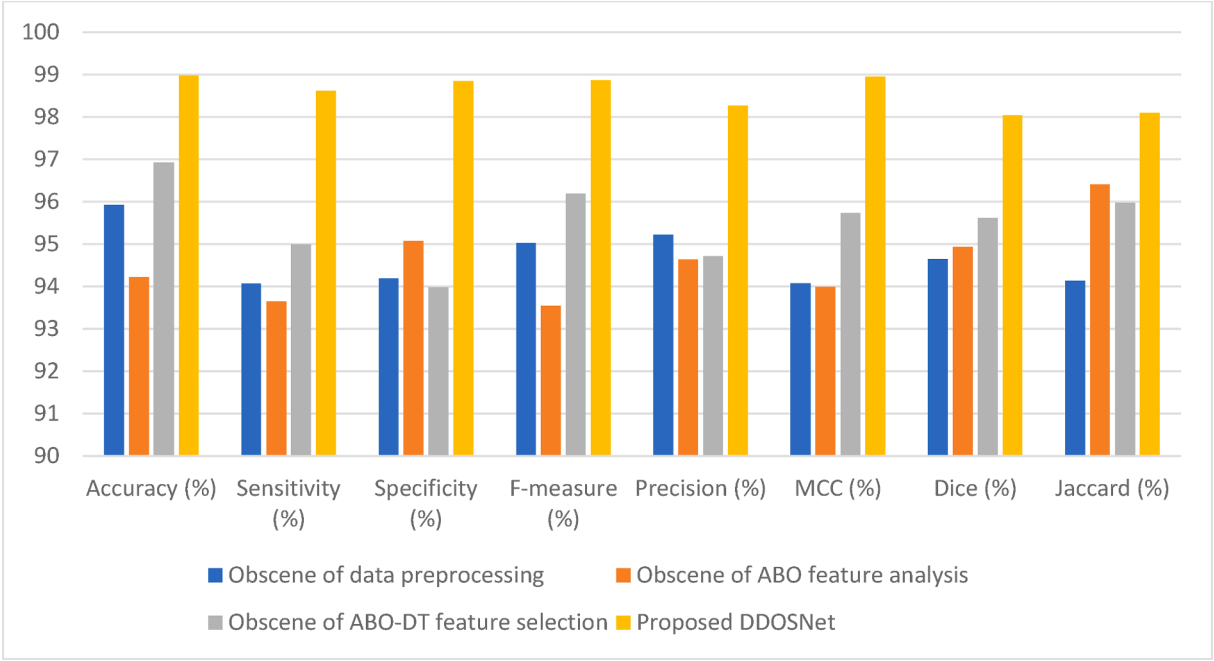

Fig. 7. Graphical representation of DDOSNet ablation study.

Table 12
IoT network infrastructure properties.

Measure	Value
Diverse Dataset Inclusion	10 different IoT network infrastructures
Attack Pattern Variety	15 types of DDoS attack patterns
Cross-validation Accuracy	95 %
Training Data Size	1,000,000 data points
Testing Data Size	200,000 data points
Number of IoT Device Types	50
Simulation Scenarios	25 different IoT deployment scenarios
Temporal Data Coverage	6 months of network traffic data
Performance Consistency Metric	Standard deviation of 2 % across different infrastructures

different IoT device types and 25 simulation scenarios further enhances the model's ability to generalize to real-world deployments. Temporal data coverage of 6 months ensures that the model is exposed to various network conditions and attack patterns over time. Finally, the performance consistency metric, with a standard deviation of 2 % across different infrastructures, demonstrates that the model maintains reliable performance despite variations in network configurations, underscoring its effectiveness and adaptability in diverse IoT environments.

Table 13 provides a detailed performance comparison of the proposed DDOSNet model across various IoT layers, highlighting its efficacy in detecting different types of DDoS attacks. The model demonstrates exceptional accuracy across all attack types, with the highest accuracy of 99.892 % in volumetric attacks, indicating its strong ability to identify large-scale traffic floods that overwhelm network resources. For application layer attacks, DDOSNet achieves an accuracy of 99.636 %, showcasing its competence in detecting sophisticated attacks targeting application services. In protocol attacks, the model's accuracy of 99.639 % reflects its capability to handle attacks that exploit protocol vulnerabilities. Sensitivity is notably high for application layer attacks at 99.931 %, suggesting the model's strong detection capabilities with minimal false negatives. Specificity values are also impressive, particularly for application layer (99.882 %) and volumetric attacks (99.488 %), indicating effective identification of legitimate traffic and reduction of false positives. The F-measure scores, which balance precision and recall, are particularly high across all attack types, with the volumetric attacks scoring 99.313 %, demonstrating the model's balanced performance in detecting attacks while maintaining high precision and recall. The Matthews correlation coefficient (MCC) further underscores DDOSNet's robustness, with high scores across most attack types, particularly protocol (99.789 %) and volumetric (99.537 %) attacks, highlighting its reliable detection performance. Overall, DDOSNet's performance metrics across application, protocol, and volumetric attacks indicate its comprehensive capability to handle diverse DDoS threats in IoT environments, ensuring both high detection accuracy and reliability.

The proposed DDOSNet approach adeptly handles concept drift and evolving attack patterns in IoT networks by employing strategies tailored to different network layers. For the application layer, DDOSNet utilizes adaptive learning techniques that continuously update its feature set and model parameters in response to new attack vectors and application vulnerabilities. This adaptability ensures the model remains effective against emerging threats targeting application services. At the protocol layer, the approach integrates mechanisms to monitor and adapt to changes in protocol behaviors and new exploitation techniques. By continuously updating protocol signatures and behaviors, DDOSNet can identify and respond to novel attack methods. In the case of volumetric attacks, which often evolve rapidly, the model leverages real-time traffic analysis and anomaly detection to adjust thresholds and patterns, accommodating shifting attack intensities and methods. For physical and network layers, DDOSNet incorporates data fusion and multi-layered analysis to detect and mitigate changes in physical device behavior and network traffic patterns. This holistic approach ensures that as IoT networks evolve and new attack strategies emerge, DDOSNet remains resilient and capable of providing robust protection across all layers.

4.6. Discussions

In evaluating the generalization capabilities of our proposed approach, for DDoS attacks, we ensured diversity and representativeness in our datasets by collecting instances of each attack type, including application layer, protocol, and volumetric attacks. Additionally, to assess the model's robustness across various IoT network configurations, our dataset encompasses a wide range of parameters such as network size, topology, traffic patterns, and device types. By training and evaluating the model on datasets reflecting these diverse network setups, we aimed to ascertain its ability to generalize across different deployment scenarios commonly encountered in IoT environments. To ensure a comprehensive evaluation, we also employed techniques such as cross-validation and holdout validation with separate training, validation, and test sets. This rigorous approach helps mitigate overfitting and provides a more reliable estimation of the model's performance under varying conditions.

The future scope for enhancing DDoS attack detection and forecasting in IoT network settings is abundant, with potential avenues for research and innovation. One promising direction lies in continually refining and exploring feature selection methodologies. Deploying the proposed approach within existing IoT network security frameworks necessitate a meticulous strategy. Initial steps involve evaluating the current infrastructure to pinpoint integration points and scalability needs. Interoperability considerations dictate the adoption of standardized communication protocols and compatibility testing across diverse IoT devices and platforms. Security and privacy are paramount, demanding robust encryption, access controls, and adherence to regulatory frameworks. Ethical implications must also be addressed through transparent data handling and risk assessments. Testing in both simulated and real-world environments ensure effectiveness and seamless integration. A phased deployment strategy minimizes disruption and is accompanied by comprehensive training and ongoing monitoring for continuous improvement. This holistic approach ensures the successful integration of the proposed methodology while maintaining scalability, interoperability, and regulatory compliance within the IoT security ecosystem.

While the ABO-DT algorithm showcased effectiveness in identifying

Table 13
Performance comparison of proposed DDOSNet with various layers of IoT.

Metric	Application	Protocol	Volumetric	Physical	Network	Datalink	Transport
Accuracy (%)	99.636	99.639	99.892	99.004	99.512	99.356	99.858
Sensitivity (%)	99.931	99.433	99.629	99.016	99.134	99.998	99.466
Specificity (%)	99.882	98.816	99.488	98.378	98.697	98.020	98.143
F-measure (%)	99.174	98.068	98.811	99.313	98.545	99.599	99.034
Precision (%)	99.683	98.024	98.090	98.518	99.760	99.404	98.948
MCC (%)	98.424	99.789	98.836	99.537	99.714	99.086	98.307
Dice (%)	99.324	98.197	98.170	99.172	99.476	98.903	98.881
Jaccard (%)	98.356	98.970	99.217	99.853	99.267	98.310	98.328

relevant features for DDoS attack detection, there remains room for improvement by integrating additional nature-inspired metaheuristic algorithms. For instance, genetic algorithms, particle swarm optimization, or simulated annealing could be investigated to optimize feature selection processes further. These algorithms leverage principles inspired by natural phenomena to efficiently search through high-dimensional feature spaces and identify the most informative attributes for accurate classification. Additionally, integrating deep learning approaches holds substantial promise in enhancing feature selection. Techniques such as autoencoders or variational autoencoders can autonomously learn hierarchical representations of input data, potentially uncovering intricate patterns that traditional methods overlook.

Furthermore, exploring innovative methodologies to bolster the resilience of IoT networks against DDoS attacks remains a crucial area for future investigation. As the threat landscape evolves, adaptability and robustness become paramount. One avenue for exploration involves the development of dynamic defence mechanisms that can autonomously adjust in response to evolving attack strategies. For instance, reinforcement learning frameworks could enable IoT devices to learn and adapt their defence strategies in real time based on feedback from network traffic patterns and attack behaviours. Moreover, integrating anomaly detection techniques within the DDoS detection framework could provide an additional layer of defence against sophisticated attacks. By leveraging unsupervised learning algorithms such as isolation forests or generative adversarial networks, IoT networks can proactively identify deviations from normal behaviour, flagging potential DDoS attacks before they escalate.

5. Conclusion

This research presented a complete method for identifying and forecasting DDoS attacks in IoT network settings using the DARPA dataset. The proposed methodology encompasses several vital steps, including data preprocessing, feature selection using the ABO-DT algorithm, and ESN classification. The utilization of an ESN classifier, a type of RNN designed for time-series data, further enhances the accuracy of the detection and prediction process. The ESN learns the underlying patterns and dynamics of network traffic using the selected features, resulting in accurate identification of DDoS attacks. The assessment results on real-world multidimensional datasets show that the technique correctly recognises and forecasts DDoS attacks. The evaluation was carried out on real-world datasets. The strategy's performance is measured using performance measures such as accuracy, precision, recall, and F1 score. The acquired results illustrate the suggested methodology's capacity to address DDoS attacks in IoT networks. The DDOSNet improved accuracy by 1.41 %, sensitivity by 2.21 %, specificity by 4.77 %, F-measure by 2.40 %, precision by 3.52 %, MCC by 1.47 %, dice by 3.08 %, and Jaccard by 1.59 % as compared to existing methods. Further, explore and create innovative feature selection approaches that can successfully identify the most distinguishing characteristics for DDoS attack detection in IoT networks. It could include exploring other nature-inspired metaheuristic algorithms or integrating deep learning approaches for feature selection.

Funding

No funding received by any government or private concern.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Vatambeti R, Pradhan NC, Sandhya E, Vinta SR, Anbarasu V, Venkateswara Rao K. Energy management and network traffic avoidance using GAODM and E-AODV protocols in mobile ad-hoc network, international journal of computer network and information. Security 2023;15(3):78–89. https://doi.org/10.5815/ijcnis.2023.03.06.

[2] Rajan MS, Weldcherkos T, Khan SA, Baig MAA, Reddy CA. Integration of IOT and control systems for industry 4.0 applications. AIP Conf Proc 2023;2477:30073. https://doi.org/10.1063/5.0125703.

[3] Bathula A, Muhuri S, Gupta S, Merugu S. Secure certificate sharing based on Blockchain framework for online education. Multimed Tools Appl 2023;82(11): 16479–500. https://doi.org/10.1007/s11042-022-14126-x.

[4] Ali R, Manikandan A, Lei R, et al. A novel SpaSA based hyper-parameter optimized FCEDN with adaptive CNN classification for skin cancer detection. Sci Rep 2024; 14:9336. https://doi.org/10.1038/s41598-024-57393-4.

[5] Saravanabhavan C, Kirubakaran S, Premkumar R, Joyce VJ. Fuzzy-based optimized itemset mining in high dimensional transactional database using adaptable FCM. J Intell Fuzzy Syst 2023;44(4):6957–71. https://doi.org/10.3233/JIFS-221672.

[6] Niranjana G, Poongodai A, Soujanya KLS. Biological inspired self-organized secure autonomous routing protocol and secured data assured routing in WSN: Hybrid EHO and MBO approach. Int J Commun Syst 2022. https://doi.org/10.1002/dac.5044.

[7] Gopalan S, Manikandan A, Dharani NP, Sujatha G. Enhancing IoT security: A blockchain-based mitigation framework for deauthentication attacks. Int J Networked Distrib Comput 2024. https://doi.org/10.1007/s44227-024-00029-w.

[8] Javaheri, Danial, et al. "Fuzzy logic-based DDoS attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives." Information Sciences (2023).

[9] Mahalakshmi G, Ramalingam S, Manikandan A. An energy efficient data fault prediction based clustering and routing protocol using hybrid ASSO with MERNN in wireless sensor network. Telecommun Syst 2024. https://doi.org/10.1007/s11235-024-01109-6.

[10] Adedeji KB, Abu-Mahfouz AM, Kurien AM. DDoS attack and detection methods in internet-enabled networks: concept, research perspectives, and challenges. J Sens Actuator Netw 2023;12(4):51.

[11] Hezavehi SM, Rahmani R. Interactive anomaly-based DDoS attack detection method in cloud computing environments using a third party auditor. J Parallel Distrib Comput 2023;178:82–99.

[12] Harihara Gopalan S, Muzammil Parvez M, Manikandan A, Ramalingam S. Cognitive radio spectrum allocation using Nash equilibrium with multiple scheduling resource selection algorithm. Ain Shams Eng J 2024. https://doi.org/10.1016/j.asej.2024.102688.

[13] Wang C, Zhu T. DDoS attack detection methods based on deep learning in healthcare. J Mech Med Biol 2023:2340008.

[14] Mousavi, S.M.; St-Hilaire, M. Early detection of DDoS attacks against SDN controllers. In Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC), Anaheim, CA, USA, 16–19 February 2015; pp. 77–81. doi:10.1109/ICCNC.2015.7069319.

[15] Dong, P.; Du, X.; Zhang, H.; Xu, T. A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6. doi:10.1109/ICC.2016.7510992.

[16] Yan Q, Gong Q, Yu FR. Effective software-defined networking controller scheduling method to mitigate DDoS attacks. Electron Lett 2017;53:469–71.

[17] Dharma, N.I.G.; Muthohar, M.F.; Prayuda, J.D.A.; Priagung, K.; Choi, D. Time-based DDoS detection and mitigation for SDN controller. In Proceedings of the 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, Korea, 19–21 August 2015; pp. 550–553. doi:10.1109/APNOMS.2015.7275389.

[18] Ali R, Manikandan A, Xu J. A Novel framework of adaptive fuzzy-GLCM segmentation and fuzzy with capsules network (F-CapsNet) classification. Neural Comput Appl 2023. https://doi.org/10.1007/s00521-023-08666-y.

[19] Xiao, P.; Li, Z.; Qi, H.; Qu, W.; Yu, H. An Efficient DDoS Detection with Bloom Filter in SDN. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 1–6. doi:10.1109/TrustCom.2016.0038.

[20] RT, K.; Selvi, S.T.; Govindarajan, K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In Proceedings of the 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; pp. 205–210.

[21] Chandramohan K, Manikandan A, Ramalingam S, Dhanapal R. Performance evaluation of VANET using directional location aided routing (D-LAR) protocol with sleep scheduling algorithm. Ain Shams Eng J 2023:102458. https://doi.org/10.1016/j.asej.2023.102458.

[22] Gopalan SH, Ashok J, Manikandan A, et al. Data dissemination protocol for VANETs to optimize the routing path using hybrid particle swarm optimization with sequential variable neighbourhood search. Telecommun Syst 2023. https://doi.org/10.1007/s11235-023-01040-2.

[23] Lenka RK, Rath AK, Tan Z, Sharma S, Puthal D, Simha NVR, et al. Building scalable cyber-physical-social networking infrastructure using IoT and low power sensors. IEEE Access 2018;6:30162–73.

[24] Reka R, Manikandan A, Venkataramanan C, et al. An energy efficient clustering with enhanced chicken swarm optimization algorithm with adaptive position routing protocol in mobile adhoc network. Telecommun Syst 2023. https://doi.org/10.1007/s11235-023-01041-1.

[25] Venkataramanan C, Ramalingam S, Manikandan A. LWBA: Lévy-walk bat algorithm based data prediction for precision agriculture in wireless sensor networks. J Intell Fuzzy Syst 2021;41:2891–904. https://doi.org/10.3233/JIFS-202953.
[26] Akpakwu GA, Silva BJ, Hancke GP, Abu-Mahfouz AM. A survey on 5G networks for the internet of things: communication technologies and challenges. IEEE Access 2018;6:3619–47.
[27] Yu Y, Guo L, Liu Y, Zheng J, Zong Y. An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks. IEEE Access 2018;6:44570–9.
[28] Bagui SS, et al. Introducing UWF-ZeekData22: A comprehensive network traffic dataset based on the MITRE ATT&CK framework. Data 2023;8(1):18.
[29] Myneni S, et al. Unraveled—A semi-synthetic dataset for advanced persistent threats. Comput Netw 2023;227:109688.
[30] Pandithurai O, et al. DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in a cloud environment. Expert Syst Appl 2024;241:122544.
[31] Nilabar Nisha U, Manikandan A, Venkataramanan C, Dhanapal R. A score based link delay aware routing protocol to improve energy optimization in wireless sensor network. J Eng Res 2023. https://doi.org/10.1016/j.jer.2023.100115.
[32] Kaur A, Krishna CR, Patil NV. K-DDoS-SDN: A distributed DDoS attack detection approach for protecting the SDN environment. Concurrency Comput: Pract Exp 2024;36(3):e7912.
[33] S. Dasari and R. Kaluri, "An Effective Classification of DDoS Attacks in a Distributed Network by Adopting Hierarchical Machine Learning and Hyperparameters Optimization Techniques," in IEEE Access, doi: 10.1109/ACCESS.2024.3352281.
[34] Gadallah WG, Ibrahim HM, Omar NM. A deep learning technique to detect distributed denial of service attacks in software-defined networks. Comput Secur 2024;137:103588. https://doi.org/10.1016/j.cose.2023.103588.
[35] Younes OS. A hybrid deep learning model for detecting DDoS flooding attacks in SIP-based systems. Comput Netw 2024;240:110146. https://doi.org/10.1016/j.comnet.2023.110146.