Heliyon 10 (2024) e29410

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

OOA-modified Bi-LSTM network: An effective intrusion detection framework for IoT systems

Siva Surya Narayana Chintapalli [a], Satya Prakash Singh [a], Jaroslav Frnda [b,c], Parameshachari Bidare Divakarachari [d,*], Vijaya Lakshmi Sarraju [a], Przemysław Falkowski-Gilski [e]

[a] Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, India
[b] Department of Quantitative Methods and Economic Informatics, Faculty of Operation and Economics of Transport and Communications, University of Zilina, 01026, Zilina, Slovakia
[c] Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VSB Technical University of Ostrava, 70800, Ostrava, Czech Republic
[d] Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru 560064, Visvesvaraya Technological University, Belagavi, India
[e] Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233, Gdansk, Poland

ARTICLE INFO

Index-values:
Bi-directional long short-term memory network
Exponential linear unit activation function
Internet of things
Intrusion detection system
Osprey optimization algorithm

ABSTRACT

Currently, the Internet of Things (IoT) generates a huge amount of traffic data in communication and information technology. The diversification and integration of IoT applications and terminals make IoT vulnerable to intrusion attacks. Therefore, it is necessary to develop an efficient Intrusion Detection System (IDS) that guarantees the reliability, integrity, and security of IoT systems. The detection of intrusion is considered a challenging task because of inappropriate features existing in the input data and the slow training process. In order to address these issues, an effective meta heuristic based feature selection and deep learning techniques are developed for enhancing the IDS. The Osprey Optimization Algorithm (OOA) based feature selection is proposed for selecting the highly informative features from the input which leads to an effective differentiation among the normal and attack traffic of network. Moreover, the traditional sigmoid and tangent activation functions are replaced with the Exponential Linear Unit (ELU) activation function to propose the modified Bi-directional Long Short Term Memory (Bi-LSTM). The modified Bi-LSTM is used for classifying the types of intrusion attacks. The ELU activation function makes gradients extremely large during back-propagation and leads to faster learning. This research is analysed in three different datasets such as N-BaIoT, Canadian Institute for Cybersecurity Intrusion Detection Dataset 2017 (CICIDS-2017), and ToN-IoT datasets. The empirical investigation states that the proposed framework obtains impressive detection accuracy of 99.98 %, 99.97 % and 99.88 % on the N-BaIoT, CICIDS-2017, and ToN-IoT datasets, respectively. Compared to peer frameworks, this framework obtains high detection accuracy with better interpretability and reduced processing time.

* Corresponding author.
E-mail addresses: cssuryanarayana@bitmesra.ac.in (S.S.N. Chintapalli), sp.singh@bitmesra.ac.in (S.P. Singh), jaroslav.frnda@uniza.sk (J. Frnda), paramesh@nmit.ac.in (P. Bidare Divakarachari), vijayalakshmi@bitmesra.ac.in (V.L. Sarraju), przemyslaw.falkowski@eti.pg.edu.pl (P. Falkowski-Gilski).

https://doi.org/10.1016/j.heliyon.2024.e29410
Received 21 September 2023; Received in revised form 16 March 2024; Accepted 8 April 2024
Available online 13 April 2024

1. Introduction

In the present scenario, IoT is extensively utilized in several industries such as energy management, water management, smart agriculture, environmental monitoring, retail, smart home automation, etc. [1–3]. IoT devices pose a severe threat from cyber-attacks like data leakage, spoofing, Distributed Denial-of-Service (DDoS) etc. These attacks affect one or more IoT devices which are used as 'platforms' or 'resources' [4,5]. As an outcome, it is vital to safeguard data and secure IoT devices by developing intrusion-resistant IoT systems [6]. An efficient IDS is developed for identifying intrusions in IoT systems and securing device communication. Several IDSs are developed in recent periods for securing internet communication [7,8]. The IDS precisely monitors IoT systems and alerts administrators while malicious behaviors are detected in the system. IoT devices are becoming more portable and compact, but are limited in battery capacity and size [9]. Additionally, IoT devices communicate with each other using light-weighted protocols; therefore require an energy efficient and light-weighted attack detection technology [10].

Developing an efficient framework accurately detects intrusion attacks in real time scenarios, but it is challenging while dealing with a vast amount of IoT data [11,12]. By reviewing existing literature, it is evident that most IDSs detect only specific attacks, because these systems are trained only on specific attack types [13,14]. IoT devices with limited battery resources, memory, and computing ability are not able to perform computational tasks, because they generate a vast communication and computation load. Therefore, it is necessary to develop an efficient security tool to strike a balance between performance and security [15]. Recently, deep learning models have gained increased attention from researchers across several domains to overcome diverse problems [16,17]. Moreover, the data from IoT is highly challenging because of large amount of information. A large amount of feature makes the evaluation and attack detection a complex task [18]. Hence, the feature selection i.e., part of dimension reduction risk is essential in selecting the optimum feature subset to depict the overall dataset [19]. The feature reduction using the metaheuristic algorithm provides improved results because it offers the best optimum results [20].

The primary motivation of this paper is to develop a modified version of a deep learning model for precise intrusion detection in IoT systems with limited processing time.

The contributions of this paper are illustrated below:

- The OOA is chosen for feature selection due to its ability of avoiding the local optima risk and obtaining the global optimum solution that helps to discover the highly informative features. The OOA selects the dominant and discriminative features from the normalized IoT data which leads to ensure an effective differentiation among normal and attack traffic.
- The Bi-LSTM is modified by incorporating the ELU activation function for performing effective detection. Specifically, the Bi-LSTM is chosen because it is goof at obtaining the long-term dependencies. In the modified Bi-LSTM, an ELU activation function is incorporated with the conventional Bi-LSTM network to prevent vanishing gradients and overfitting problems by efficiently recognizing the sequences and patterns of IoT system traffic.

This paper is prepared in this manner: A literature review related to the topic "intrusion detection" is conducted in section 2. Theoretical and mathematical explanations about the proposed framework, OOA based modified Bi-LSTM network are presented in section 3. The empirical investigation and the conclusion of the proposed IDS are outlined in sections 4 and 5, respectively.

2. Literature survey

Keshk et al. [21] developed an efficient IDS for IoT systems based on the LSTM network. This LSTM uses a different set of input features (partial dependence plot, individual conditional expectation, permutation feature importance, and Shapley additive explanations) for identifying cyberattacks. The results demonstrated that the developed IDS obtained higher interpretability, processing time, and detection accuracy than the peer systems. The developed IDS had the potential in assisting decision makers and administrators to understand the behaviour of attacks. However, generating traffic data as sequences by the LSTM was a time consuming and complex process.

Altunay and Albayrak [22] presented a hybrid IDS for ensuring security in Industrial IoT (IIoT) systems. The presented IDS integrated LSTM with Convolutional Neural Network (CNN) for effective identification of intrusion attacks in IIoT systems. The integration of LSTM and CNN offered several benefits in intrusion detection such as, the extraction of useful temporal and spatial patterns in IIoT system's traffic data. In contrast, the hybrid model (LSTM + CNN) was computationally intensive, because it required high memory and processing power. Furthermore, the integration of LSTM and CNN resulted in a complex network structure, therefore, expertise was needed to fine tune and manage complex models.

Gebretsadik et al. [23] introduced an Enhanced Bloom Filter (EBF) for precise detection of intrusions in IoT systems. The experiment conducted on a real time intrusion dataset revealed that the EBF was accurate, faster, and memory efficient compared to other filters like the cuckoo filter and the traditional bloom filter. However, the sub-optimal use of hash functions in the EBF increased the false positives during intrusion detection.

Alsirhani et al. [24] designed an intelligent IDS by integrating feature-based and deep learning-based techniques. Initially, a min-max normalization technique was applied for transforming the numeric datasets into a predefined range of zero to one. Secondly, the features, namely, autoregressive data, data percentiles, correlation coefficient, mutual information, information gain, standard deviation, mode, median, and mean were extracted from the rescaled datasets. Thirdly, discriminative features were selected by employing the African vulture optimization algorithm. Finally, these discriminative features were passed into a hybrid model (LSTM +

Deep Belief Network (DBN)) to categorize the attack and normal packets. The outcomes obtained on real time intrusion datasets revealed that the presented framework had a reliable potential in cybersecurity applications. The integration of LSTM and DBN offered enhanced performance in intrusion detection by capturing complex patterns in IoT data, but it also faced challenges related to interpretability, computational cost, and complexity.

Keserwani et al. [25] initially integrated the Particle Swarm Optimization (PSO) algorithm and the Grey Wolf Optimization (GWO) algorithm to eliminate inappropriate, irrelevant, and unnecessary features in intrusion datasets that resulted in high accuracy and low detection time. Furthermore, the obtained features were input into the random forest for categorizing intrusion attacks. Similarly,

Table 1
Literature table.

Author	Methodology	Strength	Research gap
Keshk et al. [21]	LSTM network was developed for enhancing the IDS.	This LSTM based IDS had the capacity in supporting the decision makers and administrators for knowing the attack behaviour.	The generation of traffic as sequences using LSTM was a time consuming and complex process.
Altunay and Albayrak [22]	The combination of LSTM and CNN was developed for detecting the attacks in IIoT.	The LSTM + CNN extracted beneficial temporal and spatial patterns.	The developed LSTM + CNN required high memory and processing power as well as, resulted in complex architecture.
Gebretsadik et al. [23]	The EBF was developed for precise identification of intrusion.	This EBF performed accurate, memory efficient and faster prediction in real time intrusion dataset.	The sub-optimal utilization of hash functions in the EBF increased the false positives
Alsirhani et al. [24]	The African vulture optimization was used to choose the discriminative features. Next, a hybrid LSTM + DBN was used for classification.	The acquisition of complex patterns was used to enhance the detection using LSTM with DBN.	The developed LSTM with DBN faced challenges of interpretability, computational cost, and complexity during the detection.
Keserwani et al. [25]	The combination of PSO and GWO was used to choose the relevant features.	The elimination of inappropriate, irrelevant, and unnecessary features was used to enhance the accuracy and minimize the detection time.	The developed work was mainly concentrated on developing the feature selection approach.
Hassan et al. [26]	The random forest was integrated with MRFO for developing an IDS.	The MRFO was used to remove the irrelevant features from the overall feature.	The random forest was ineffective in handling imbalanced data which caused biased predictions.
Anushiya and Lavanya [27]	The AFSA was developed for discovering the important properties in the data. Next, the faster recurrent CNN with genetic algorithm was used for classification.	The AFSA based important characteristic discovery was used to enhance prediction.	The integration of multiple convolutional and recurrent layers increased the model's complexity.
Alweshah et al. [28]	The EPC optimization was developed for selecting the features to enhance the classification using KNN.	The selection of appropriate features using EPC was used to enhance the IDS.	The KNN was sensitive to outliers and data points that superiorly affected the classification.
Alweshah et al. [29]	The features were selected using SSOA and KNN based classification was done for improving the prediction.	The SSOA based feature selection was used to remove the irrelevant features.	The KNN based classification was affected, because it was sensitive to outliers and data points.
Li et al. [30]	The ANN was used to detect the normal and abnormal behaviour of medical IoT. Here, the BOA was used to select the discriminative features.	The selection of discriminative features helped to enhance the accuracy of ANN.	However, the developed deep ANN was prone to overfitting problems.
Kumar et al. [31]	The IBRO was used to ensure an optimum feature selection for IDS. Next, the recurrent kernel CNN with MMBO was used for detection.	The IBRO based optimum feature selection was used to minimize the complexity of IDS.	The complex patterns among the patterns were required to be considered during the prediction.
Dahou et al. [32]	A modified RSA with CNN was developed for IDS.	Highly relevant and informative features were chosen by modified RSA in IDS.	The developed recurrent kernel CNN required huge amount of computations in detection.
Li and Yi [33]	The Bi-LSTM was integrated with CNN for performing IDS in IIoT.	The training was enhanced by using the batch normalization.	The integration of two deep learning models i.e., Bi-LSTM with CNN increased the complexity.
Elsayed et al. [34]	The improved LSTM was developed for differentiating the benign and attack traffics.	The improved LSTM based secured automatic two level IDS designed for improving classification.	The transformation of intrusion data into sequences using improved LSTM was a time consuming and complex process.
Elaziz et al. [35]	The CapSA and CNN were developed for selecting features and attack detection in IDS.	The accuracy was improved based on the chosen features from CapSA.	The contextual and temporal dependencies from the features were required to be considered for further improving the IDS.
Gharehchopogh et al. [37]	An influential feature was discovered by using the MODHHO.	The detection of influential feature was used to avoid misclassification.	This work was mainly concentrated on the feature selection, but an effective classifier was required for enhancing the prediction.
Asgharzadeh et al. [38]	The CNN and BMECapSA were used in IDS for enhancing the detection.	The local and global features were extracted by CNN with hybrid layers for enhancing the detection.	The complex patterns among the features were required to be considered by the classifier for further enhancing the IDS.

Hassan et al. [26] integrated random forest with an improved Manta Ray Foraging Optimization (MRFO) algorithm for accurate classification of intrusion attacks. However, compared to other advanced classification models, random forest was ineffective in handling imbalanced intrusion data that led to biased predictions.

Anushiya and Lavanya [27] introduced a novel algorithm named the assimilated Artificial Fish Swarm Algorithm (AFSA) to find important properties in IoT based intrusion datasets related to the problem statement. After gathering important properties of the datasets, classification was accomplished by combining a faster recurrent CNN model with a genetic algorithm. In this context, the faster recurrent CNN model integrated numerous convolutional and recurrent layers that significantly increased the model's complexity, and required longer training time and more computational resources.

Alweshah et al. [28] and Alweshah et al. [29] integrated the Emperor Penguin Colony (EPC) optimization algorithm and the Shuffled Shepherd Optimization Algorithm (SSOA) with a K-Nearest Neighbor (KNN) classifier for accurate classification of intrusion attacks in IoT systems. The traditional KNN classifier was sensitive to outliers and data points that superiorly affected the classification performance. Additionally, Li et al. [30] used an Artificial Neural Network (ANN) for detecting normal and abnormal behaviors in medical IoT systems. However, the detection accuracy of ANN depended completely on the selected features, so the selection of discriminative and important features from the traffic data was vital in this literature. In the developed framework, the Butterfly Optimization Algorithm (BOA) was applied for selecting discriminative features. The developed framework, BOA-ANN obtained higher results in intrusion detection, but the deep ANN was prone to overfitting problems.

Kumar et al. [31] presented an automated IDS for identifying attacks in IoT systems. The presented automated IDS included three phases: (i) data preprocessing using the min-max normalization technique, (ii) optimal feature selection using the Improved Battle Royale Optimization (IBRO) algorithm, and (iii) intrusion attack detection by a recurrent kernel CNN model with the Modified Monarch Butterfly Optimization (MMBO) algorithm. Firstly, the min-max normalization technique improved the data quality by decreasing the introduced noise, while IBRO selected the most optimal features that reduced the computational complexity of this IDS. Furthermore, the incorporation of the MMBO algorithm with the recurrent kernel CNN model reduced overfitting problems and improved the performance of the classifier. Additionally, Dahou et al. [32] combined the modified Reptile Search Algorithm (RSA) with a CNN model for intrusion detection in IoT systems. This framework obtained competitive classification performance related to other well-known frameworks. However, while dealing with large intrusion datasets, the recurrent kernel CNN and traditional CNN models proved to be computationally intensive.

Li and Yi [33] integrated the Bi-LSTM network with a CNN model for intrusion detection in IoT systems. As discussed in the literature above, the integration of two deep learning models maximized the complexity of the classifier and needed more computational resources. Elsayed et al. [34] introduced an improved LSTM network for identifying attack categories, differentiating between benign and attack traffics, and defining sub-attack types. The improved LSTM network was trained and validated using two realistic datasets for proving its effectiveness over other IDSs. The transformation of intrusion data into sequences by the improved LSTM network was a time consuming and complex process.

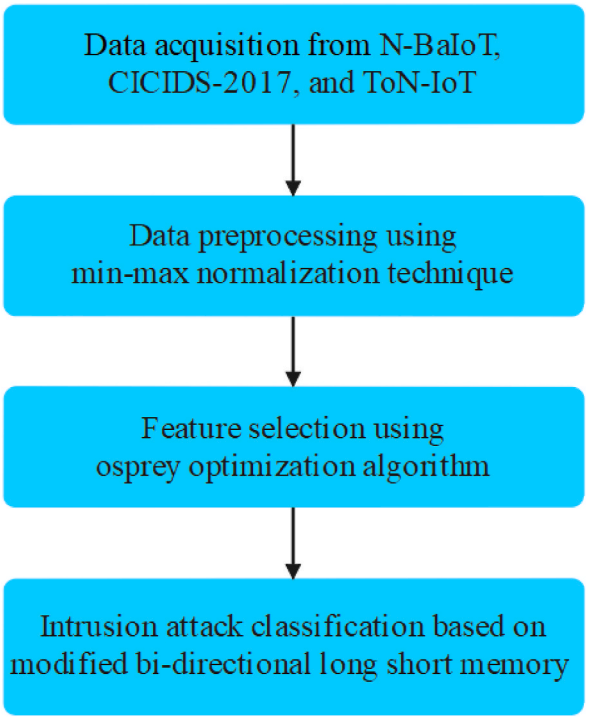

Fig. 1. Workflow of the OOA based modified Bi-LSTM network.

Elaziz et al. [35] presented the combination of swarm intelligence algorithm and CNN for developing an effective IDS for IoT-cloud. Initially, a CNN was used to acquire an optimal feature from the data. Next, a Capuchin Search Algorithm (CapSA) was used to obtain optimum features for enhancing the accuracy. However, the contextual and temporal dependencies from the features were required to be considered during the detection. Yi et al. [36] categorized the IDS approaches developed in fog environment. Gharehchopogh et al. [37] developed the Multi-Objective Dynamic Harris Howks Optimization (MODHHO) for discovering the influential features while performing Botnet detection in the IoT data. Next, the classifiers of Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Multilayer Perceptron (MLP) were used for precise detection. This work was mainly concentrated on the MODHHO based selecting features, but an effective classifier was required for improving the IDS.

Asgharzadeh et al. [38] presented the CNN and Binary Multi-objective Enhanced CapSA (BMECapSA) for improving the IDS. The local and global features were extracted by using the CNN with hybrid layers and BMECapSA was used to select the features. This work was mainly concentrated on feature selection, but the complex patterns among the feature were additionally needed to be considered during classification.

The strengths and research gaps of the existing researches are given in the below Table 1.

Generally, transforming intrusion data as sequences by the improved LSTM [34] is a time consuming and complex process. Moreover, irrelevant features existing in the data affect the process of classification. To highlight the afore-stated concerns and enhance the performance of intrusion detection in IoT systems, a novel framework, OOA based modified Bi-LSTM network is proposed in this paper. The OOA based feature selection is developed for selecting the optimum features to perform an effective differentiation during the detection.

3. Methods

In the context of intrusion detection, the proposed framework comprises four phases. In the initial phase, IoT data is collected from the N-BaIoT, CICIDS-2017, and ToN-IoT datasets. Secondly, data preprocessing is conducted utilizing the min-max normalization technique, and thirdly, significant features are chosen by implementing OOA. In the final phase, a modified Bi-LSTM network is designed to classify the normal and attack types. The workflow of the OOA based modified Bi-LSTM network is depicted in Fig. 1.

3.1. Dataset description

The proposed framework, OOA based modified Bi-LSTM network performance is tested on three online benchmark datasets. The N-BaIoT, CICIDS-2017, and ToN-IoT datasets include real world IoT data obtained from various sources confirming the IDS is trained and tested on realistic data which represents the authentic IoT behaviour. Moreover, these datasets include a comprehensive set of features comprising characteristics of network traffic, communication protocols and device communications. These features support in evaluating and displaying the behaviour of IoT network that leads to the design of precise IDS. N-BaIoT dataset is one of the multivariate and sequential datasets, comprising 115 real attributes, two attacks (Bashlite and Mirai), and a total of 7,062,606 instances. This dataset contains traffic data acquired from nine IoT devices, which are infected by Bashlite and Mirai attacks [39]. The nine IoT devices are as follows: Samsung SNH 1011 N web camera, simple home XCS7-1002-WHT security camera, simple home XCS7-1003-WHT security camera, Philips B120 N/10 baby monitor, ecobee thermostat, provision PT-737E security camera, provision PT-838 security camera, ennio doorbell, and danmini doorbell [40].

Furthermore, CICIDS-2017 dataset comprises real-time network data recorded at various periods. This dataset includes several types of attacks like portscan, infiltration, Bot, heartbleed, golden-eye, etc., where these attacks are not found in other datasets [41]. The features of the CICIDS-2017 dataset are categorized into different types by means of their reliable and realistic benchmarking. Eleven criteria are used for benchmarking to ensure the evaluation's reliability [42]. The summary of the CICIDS-2017 dataset is provided in Table 2, which includes details about the types of classes, file names, and the number of records in each class.

Additionally, in the ToN-IoT dataset, data records are acquired from various sources such as records of windows operating systems,

Table 2
Summary of the CICIDS-2017 dataset.

Files	Classes	Records
Friday-DDoS.pcap_ISCX.csv	DDoS and benign	128,027 and 97,718
Friday-portscan.pcap_ISCX.csv	Portscan and benign	158,930 and 127,537
Friday-pcap_ISCX.csv	Bot and benign	1966 and 189,067
Thursday-infilteration.pcap_.csv	Infiltration and benign	36 and 288,566
Thursday-webattacks.pcap_ISCX.csv	Web attack-Cross Site Scripting (XSS), web attack-Structured Query Language (SQL) injection, web attack-brute force, and benign	652, 21, 1,507, and 168,186
Wednesday-pcap_ISCX.csv	Heartbleed, DoS slow-loris, DoS slow-http-test, DoS-hulk, golden-eye, and benign	11, 5,796, 5,499, 231,073, 10,293, and 440,031
Tuesday-hours.pcap_ISCX.csv	File Transfer Protocol (FTP)-patator, benign, and Secure Shell (SSH)-patator	7,938, 432,074, and 5897
Monday-hours.pcap_ISCX.csv	Benign states normal human activities	529,918

records of Linux operating systems, telemetry data records from different connected devices, network traffic records from Industrial IoT (IIoT) systems, and more [43]. This dataset is available in Comma-Separated Values (CSV) format and includes a label column that represents normal or attack behavior. The subtypes of the attack category are as follows: Man in the Middle (MITM), XSS, back door, data injection, DDoS, DoS, scanning, password attack, and ransomware. These nine types of attacks are collected from IIoT networks using various IIoT and IoT sensors [44]. This dataset comprises 300,000 normal records and 161,043 attack records.

3.2. Data pre-processing

After acquiring IoT based intrusion data from N-BaIoT, CICIDS-2017, and ToN-IoT datasets, data pre-processing is performed by eliminating errors such as inconsistent, incomplete and missing value. Furthermore, data rescaling is carried out utilizing the min-max normalization technique, which converts intrusion data into a particular range (0–1). This process preserves the order and relationships of values within every feature [45,46]. Additionally, in the context of intrusion detection, the min-max normalization technique is more robust to variations and outliers present in the intrusion data. The classification model becomes less sensitive to extreme values by constraining features to a particular range (0–1). This technique ensures that all features have the same scale and is very crucial in intrusion detection, because features have different measurement scales and units. This min-max normalization helps to prevent the impact of outliers during the detection by normalizing the data to the particular range. The formula utilized to compute this technique is presented in equation (1).

$$y_{scale} = \frac{y - y_{min}}{y_{max} - y_{min}} \tag{1}$$

where, y_{scale} indicates the mapped value of the attribute, y_{max} represents the maximum value of the attribute, y_{min} states the minimum value of the attribute, and y specifies the current value of the attribute. The output of min-max normalization technique is normalized/rescaled values ranging between 0 and 1 that reflect the relative position of every data point in the range of the original intrusion data. The rescaled data are passed into the feature selection technique for selecting the information features on the N-BaIoT, CICIDS-2017, and ToN-IoT datasets.

3.3. Feature selection

The normalized IoT data of N-BaIoT, CICIDS-2017, and ToN-IoT datasets are passed into the OOA for informative feature selection. The OOA mimics the intelligent natural behaviors of hunting and carrying of fish to a suited location to eat. Currently, OOA is one of the efficient population based optimization algorithms, which provides an appropriate solution based on the problem statement. The capacity of discarding the local optima risk and achieving the global optimum solution of OOA is utilized for discovering the most discriminative features. In this optimization algorithm, every osprey is a candidate solution which is modelled using a matrix X [47,48]. The position of every osprey $x_{i,j}$ is randomly initialized in the search space at the beginning of OOA implementation using equation (2).

$$x_{i,j} = lb_j + r_{ij} \times \left(ub_j - lb_j\right), i = 1, 2, \ldots..., N, j = 1, 2, \ldots.m \tag{2}$$

where, lb_j represents the lower bound of the j^{th} variable, r_{ij} states the random number which ranges between the interval of zero to one, ub_j represents the upper bound of the j^{th} variable, m denotes the number of variables, and N indicates the number of ospreys.

In this scenario, the objective function is evaluated based on osprey because every osprey is a candidate solution, and it is represented in a vector format V. The quality of the candidate solution is a main criterion to evaluate the values for the objective function. The best candidate solution represents that the best value is achieved for the objective function. Respectively, the worst candidate solution indicates that the worst value is obtained for the objective function. The best candidate solution is updated in every iteration by considering the maximum accuracy in the search space. The objective function considered for OOA based feature selection is accuracy that helps to discover highly relevant features which contributes to an effective intrusion detection by filtering redundant features. Therefore, the consideration of accuracy as selection criteria in OOA helps to improve the generalization and enhance the adaptability to the different datasets. The OOA terminates after reaching the maximum number of iterations.

3.3.1. Exploration phase (Identification of position and hunting of fish)

Generally, the osprey has strong eyesight, so it easily finds the fish position. It attacks and hunts the fish after finding the fish position. The identification of optimal search area is improved by increasing the exploration power of OOA and it helps this algorithm escape from the local optima problem. For every osprey, the other ospreys with better value of objective function are considered as the under-water fishes. The set of under-water fish for every osprey is mathematically determined in equation (3).

$$VP_i = \{X_k | k\epsilon\{1, 2, 3, \ldots., N\}^{V_k} < V_i\} \cup \{X_{best}\} \tag{3}$$

where, X_{best} represents the best osprey (best candidate solution), VP_i denotes the positions of the set of under-water fish for i^{th} osprey, and X indicates the matrix. The new position of the osprey is computed using equations (4) and (5) based on the ospreys movement towards the fish. Further, the osprey position is replaced based on equation (6), when the obtained new position enhances the value of objective function.

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \times \left(SV_{i,j} - Z_{i,j} \times x_{i,j}\right) \tag{4}$$

$$x_{i,j}^{P1} = \begin{Bmatrix} x_{i,j}^{P1}, & lb_j \le x_{i,j}^{P1} \le ub_j \\ lb_j, & x_{i,j}^{P1} < lb_j \\ ub_j, & x_{i,j}^{P1} > ub_j \end{Bmatrix} \tag{5}$$

$$X_i = \begin{Bmatrix} X_i^{P1}, & V_i^{P1} < V_i \\ X_i, & else \end{Bmatrix} \tag{6}$$

where, $Z_{i,j}$ represents the random number, which ranges the interval between one to two, $SV_{i,j}$ indicates the selected under-water fish for i^{th} osprey and its j^{th} dimension, V_i^{P1} states the value of objective function, $x_{i,j}^{P1}$ denotes the new position of the i^{th} osprey and its j^{th} dimension, and X_i^{P1} states the new position of the i^{th} osprey in the exploration phase.

3.3.2. Exploitation phase (Carrying the fish to the suited location)

The osprey carries the fish to a suited location after hunting the fish. The exploitation power of the OOA is increased by modelling of carrying the fish to the suited location that results in faster convergence rate. Furthermore, a new position is computed utilizing equations (7) and (8), and the prior position is replaced with the new position if it achieves better objective function value, which is mathematically stated in equation (9).

$$x_{i,j}^{P2} = x_{i,j} + \frac{lb_j + r_{ij} \times \left(ub_j - lb_j\right)}{iter}, i = 1, 2, \ldots.., N, j = 1, 2, \ldots.m, iter = 1, 2, \ldots..100 \tag{7}$$

$$x_{i,j}^{P2} = \begin{Bmatrix} x_{i,j}^{P2}, & lb_j \le x_{i,j}^{P2} \le ub_j \\ lb_j, & x_{i,j}^{P2} < lb_j \\ ub_j, & x_{i,j}^{P2} > ub_j \end{Bmatrix} \tag{8}$$

$$X_i = \begin{Bmatrix} X_i^{P2}, & V_i^{P2} < V_i \\ X_i, & else \end{Bmatrix} \tag{9}$$

where, x_{ij}^{P2} states the new position of the i^{th} osprey and its j^{th} dimension in the exploitation phase, and *iter* denotes the number of iterations. In this context, 80 % of the informative features are selected from the N-BaIoT, CICIDS-2017, and ToN-IoT datasets using OAA, which are passed into the modified Bi-LSTM network for intrusion classification. The main parameters assumed in the OOA are denoted as follows: upper bound value is 0.8, lower bound value is 0.2, maximum iteration is 100, and the size of initial population (osprey) is 100. The OAA selects 80 real attributes of Bashlite and Mirai classes and 672,347 instances in the N-BaIoT dataset. Furthermore, in the ToN-IoT dataset, the OAA selects 193,480 normal records and 120,428 attack records for classifying nine attacks (MITM, XSS, back door, data injection, DDoS, DoS, scanning, password attack, and ransomware). The records selected by OAA on the CICIDS-2017 dataset are represented in Table 3.

3.4. Intrusion attack classification

The features selected by the OOA are input into the modified Bi-LSTM network for the categorization of intrusion attacks. The conventional sigmoid and tangent activation functions are prone to vanishing gradient issues. In order to overcome this, the activation function is changed with ELU to avoid the vanishing gradient issue by enhancing the linear characteristics and fastening the training process. The LSTM network is the updated version of Recurrent Neural Network (RNN), which prevents vanishing gradient problems by applying input gate i_t, forget gate f_t, output gate o_t and memory cells c [49,50]. These three gates assist LSTM network in capturing

Table 3
Records selected by OAA on the CICIDS-2017 dataset.

Classes	Records
DDoS and benign	94,128 and 68,216
Portscan and benign	108,755 and 97,005
Bot and benign	1203 and 137,098
Infiltration and benign	36 and 208,520
Web attack-XSS, web attack-SQL injection, web attack-brute force, and benign	508, 20, 1,200, and 98,100
Heartbleed, DoS slow-loris, DoS slow-http-test, DoS-hulk, golden-eye, and benign	10, 5,000, 4,823, 100,070, 7,200, and 200,000
FTP-patator, benign, and SSH-patator	4,900, 230,000, and 3900
Benign states normal human activities	320,900

both short-term and long-term dependencies, and control the flow of the feature information. Additionally, the memory cells c controls the flow of the new feature information.

Particularly, the forget gate f_t deletes the cell state information, and the output gate o_t efficiently regulates the flow of the cell state's internal memory. Furthermore, the Bi-LSTM network accesses both future and past feature information at every time step t, which is more beneficial for tasks like intrusion detection. In comparison to the LSTM network, the Bi-LSTM network effectively recognizes sequences and patterns of IoT system traffic. The Bi-LSTM network is capable of learning short-term and long-term dependencies in both directions (forward and backward). It is vital to find sophisticated and subtle attacks in IoT systems [51]. The formula used for computing the forget gate is defined in equation (10).

$$f_t = \sigma\left(W_f \times [h_{t-1}, x_t] + b_f\right) \tag{10}$$

where, W_f denotes the weight matrix of the forget gate, b_f denotes the bias value, σ states the ELU activation function, x_t denotes the input units (selected features by the OOA), and h_{t-1} specifies the hidden state. In this context, ELU activation function has fast convergence during data training related to conventional activation functions namely, hyperbolic tangent and sigmoid. The ELU activation function decreases the computational resources and running time of the Bi-LSTM network. On the test and validation datasets, ELU activation function enhances the generalization capacity of the Bi-LSTM network by reducing the risks of overfitting and providing a more robust and smoother gradient during training. Furthermore, the mathematical expression of the weight matrix of the forget gate is presented in equation (11) [52]. Furthermore, the input gate i_t is computed utilizing equation (12).

$$\left[W_f\right]\begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} = \left[W_{fh}\ W_{fx}\right]\begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} = W_{fh}h_{t-1} + W_{fh}x_t \tag{11}$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{12}$$

where, the weight matrix and bias value of the input gate i_t are represented as W_i and b_i. The present cell memory status $c_t^{'}$ is determined based on the last output and input gates, as depicted in equation (13). Correspondingly, the cell state c_t is multiplied with forget and input gates in the element states c_{t-1} and $c_t^{'}$ as specified in equation (14).

$$c_t^{'} = \sigma\left(W_c \times [h_{t-1}, x_t] + b_c\right) \tag{13}$$

$$c_t = f_t \times c_{t-1} + i_t \times c_t^{'} \tag{14}$$

where, the weight matrix and bias value of the cell state are indicated as W_c and b_c. The new cell state c_t is generated by integrating current cell memory $c_t^{'}$ and long-term cell memory c_{t-1}. The formula utilized to compute the output gate o_t is mathematically stated in equation (15). By inspecting equation (15), the weight matrix and bias value of the output gate o_t are represented as W_o and b_o.

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o],\ h_t = o_t \times \sigma(c_t) \tag{15}$$

The parameters considered in the modified LSTM network are presented as follows: number of hidden layers is 50, number of hidden units in each layer is 150, number of epochs is 100, learning rate is 0.0001, optimizer is Adam, L2 regularization is 0.5, minimum batch size is 64, gradient threshold and gradient decay factor is one. The architecture of Bi-LSTM network is presented in Fig. 2. The empirical results of the OOA based modified Bi-LSTM network are detailed in section 4.

4. Results

The proposed framework, OOA based modified Bi-LSTM network is simulated utilizing the Python 3.11.2 software environment.

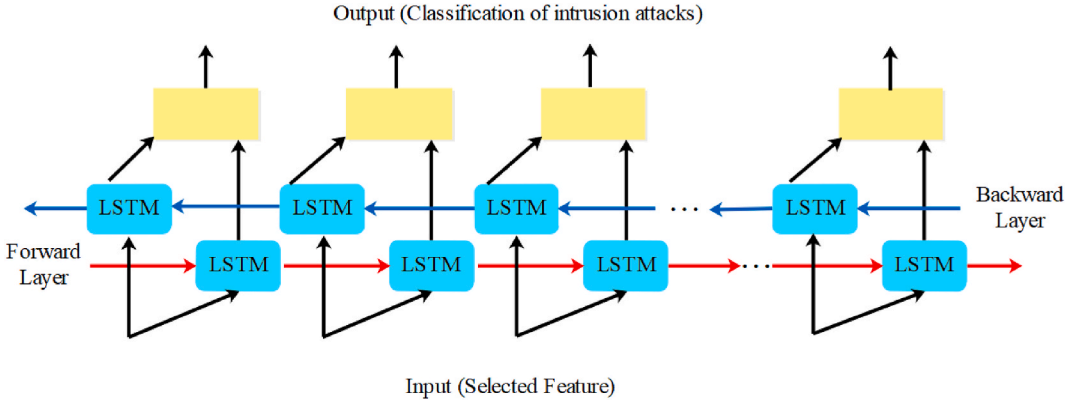

Fig. 2. Architectural diagram of the Bi-LSTM network.

Numerous software packages of the python libraries are used for analyzing intrusion data, namely Numpy, Pandas, Matplotlib, SciKit Learn, TensorFlow, and Keras. The OOA based modified Bi-LSTM network's efficacy is validated on a computer with a memory of 128 GB RAM and windows 10 pro (64-bit) operating system. This framework's effectiveness is tested on three online benchmark datasets related to IoT systems (N-BaIoT, CICIDS-2017, and ToN-IoT), and analysed using five various performance measures where the training and testing ratio used for the evaluation is 80:20. The expressions utilized to compute f1-score, accuracy, specificity, recall, and fall-out are presented in equation 16–20.

$$F1-score=\frac{2TP}{2TP+FP+FN}\times 100 \tag{16}$$

$$Accuracy=\frac{TN+TP}{TP+TN+FP+FN}\times 100 \tag{17}$$

$$Specificity=\frac{TN}{TN+FP}\times 100 \tag{18}$$

$$Recall=\frac{TP}{TP+FN}\times 100 \tag{19}$$

$$Fall-out=\frac{FP}{FP+TN}\times 100 \tag{20}$$

where, the terms FP, FN, TP, and TN are represented as False Positive, False Negative, True Positive, and True Negative.

4.1. Quantitative analysis

The quantitative results of the proposed framework (OOA based modified Bi-LSTM network) is analysed on three different benchmark datasets. These results are compared with other classifiers (LSTM, Gated Recurrent Unit (GRU), and Bi-LSTM) and optimizers (AFSA, BRO, and MBO). By inspecting Tables 4–6, it is evident that the proposed framework obtains significant classification results in comparison to other combinations. Particularly, in Table 4, the proposed framework achieves 99.89 % of f1-score, 99.98 % of accuracy, 99.90 % of specificity, 99.94 % of recall, and 99.95 % of fall-out on the N-BaIoT dataset. These obtained results are superior to other combinations of classifiers and optimizers. The OOA based modified Bi-LSTM network effectively classifies one class of 'benign' and 10 classes of 'attacks'. The accuracy comparison for the N-BaIoT dataset is mentioned in Fig. 3. The highly informative features are selected by avoiding the local optima risk and obtaining the global optimum solution during the feature subset selection. This helps to improve the detection. The parameters considered in the comparative optimizers are depicted as follows.

AFSA: Congestion rate is 20, step size is 0.5, visual rate is 1.5, try number is 50, population size is 100, error rate is 0.01, and iteration number is 100.

BRO: Lower and upper bound are 0.3 and 0.8 respectively, size of population is 100 and iteration number is 100.

MBO: Attractiveness is 0.2, light absorption coefficient is one, exhaustiveness is four, adjusting rate of butterfly is 5/12, migration period is 1.2, migration ratio is 5/12, population size is 100, and iteration number is 100.

Correspondingly in Table 5, the proposed framework obtains significant classification results with f1-score of 99.96 %, accuracy of 99.97 %, specificity of 99.92 %, recall of 99.91 %, and fall-out of 99.94 %. These results are better than other classifiers (LSTM, GRU, and Bi-LSTM) and optimizers (AFSA, BRO, and MBO) on the CICIDS-2017 dataset. The OOA based modified Bi-LSTM network

Table 4
Results achieved on the N-BaIoT dataset.

N-BaIoT dataset						
Classifier	Optimizer	F1-score (%)	Accuracy (%)	Specificity (%)	Recall (%)	Fall-out (%)
LSTM	AFSA	98.12	98.22	98.04	98.32	98.18
GRU		98.29	98.35	98.13	98.54	98.67
Bi-LSTM		98.74	98.78	98.65	98.80	97.98
Modified Bi-LSTM		99.16	98.90	98.92	98.98	98.88
LSTM	BRO	98.60	97.86	97.90	98.28	97.34
GRU		98.78	98.33	98.22	98.50	98.77
Bi-LSTM		98.92	98.56	98.78	98.88	97.86
Modified Bi-LSTM		99.04	99.08	99.14	99.44	99.22
LSTM	MBO	98.32	97.45	98.75	98.33	98.44
GRU		98.56	98.37	98.90	98.92	98.68
Bi-LSTM		99.02	98.80	99	99.04	99.20
Modified Bi-LSTM		99.44	99.53	99.12	99.48	99.35
LSTM	OOA	98.78	98.87	98.92	98.56	98.67
GRU		99.12	99.34	99.30	99.22	99.10
Bi-LSTM		99.35	99.65	99.72	99.43	99.50
Modified Bi-LSTM		99.89	99.98	99.90	99.94	99.95

Table 5
Results achieved on the CICIDS-2017 dataset.

CICIDS-2017 dataset						
Classifier	Optimizer	F1-score (%)	Accuracy (%)	Specificity (%)	Recall (%)	Fall-out (%)
LSTM	AFSA	97.12	97.56	97.55	96.90	95.44
GRU		97.46	97.90	97.80	97.77	97.39
Bi-LSTM		98.84	98.44	98.33	97.94	97.97
Modified Bi-LSTM		99.32	98.94	98.94	98.88	98.62
LSTM	BRO	97.84	96.43	96.95	97.54	97.38
GRU		96.59	97.92	97.76	97.86	96.60
Bi-LSTM		98.94	98.98	98.93	97.98	97.59
Modified Bi-LSTM		99.40	99.25	99.40	98.55	98.36
LSTM	MBO	96.57	97.62	97.44	96.62	94.79
GRU		97.82	97.89	97.89	97.87	96.28
Bi-LSTM		98.09	98.66	98.86	97.90	98.50
Modified Bi-LSTM		99.10	98.80	99.09	98.44	98.56
LSTM	OOA	98.34	98.32	98.43	97.55	97.30
GRU		98.90	98.87	98.98	97.60	97.58
Bi-LSTM		99.52	99.26	99.34	98.79	98.98
Modified Bi-LSTM		99.96	99.97	99.92	99.91	99.94

Table 6
Results achieved on the ToN-IoT dataset.

ToN-IoT dataset						
Classifier	Optimizer	F1-score (%)	Accuracy (%)	Specificity (%)	Recall (%)	Fall-out (%)
LSTM	AFSA	96.22	96.90	96.44	96.65	95.32
GRU		97.39	97.40	95.70	96.90	96.93
Bi-LSTM		98.88	97.88	96.38	97.44	97.74
Modified Bi-LSTM		99.20	98.65	97.89	98.85	98.88
LSTM	BRO	97.54	96.77	96.65	96.54	96.55
GRU		96.69	96.90	97.94	97.90	96.90
Bi-LSTM		97.03	98.54	98.57	96.50	97.86
Modified Bi-LSTM		98.04	98.59	98.79	98.94	98.84
LSTM	MBO	96.74	96.70	97.50	96.59	95.90
GRU		97.80	97.68	97.69	97.28	97.94
Bi-LSTM		98.57	97.74	98.24	98.94	97.97
Modified Bi-LSTM		98.80	98.36	98.68	98.95	98.54
LSTM	OOA	98.40	98.80	98.87	97.37	97.75
GRU		98.45	98.98	99.30	97.84	97.90
Bi-LSTM		99.22	99.37	99.63	98.90	98.54
Modified Bi-LSTM		**99.90**	**99.88**	**99.92**	**99.89**	**99.91**

significantly classifies the classes mentioned in Table 2. The common parameters considered in the existing classifiers (LSTM, GRU, and Bi-LSTM) are depicted as follows: number of epochs is 100, learning rate is 0.001, optimizer is Adam, gradient threshold and gradient decay factor is one, and minimum batch size is 64. The accuracy comparison for the CICIDS-2017 dataset is specified in Fig. 4. The detection performance is enhanced based on the OOA based informative feature selection, as well as an integration of ELU in Bi-LSTM supports in obtaining the complex and nonlinear relationships of data that additionally enhances the detection.

On the ToN-IoT dataset, in comparison to other combinations, the proposed framework obtains improved results with f1-score of 99.90 %, accuracy of 99.88 %, specificity of 99.92 %, recall of 99.89 %, and fall-out of 99.91 %, as depicted in Table 6. The accuracy comparison for the ToN-IoT dataset is mentioned in Fig. 5. The OOA based modified Bi-LSTM network efficiently classifies 10 classes (one normal class and nine attack classes (ransomware, password attack, scanning, DoS, DDoS, data injection, back door, XSS, and MITM)).

In the context of intrusion detection, the modified Bi-LSTM network (incorporation of ELU with the traditional Bi-LSTM network) efficiently handles noisy IoT data and learns a broader range of data representations by capturing negative and positive values. This process is useful to deal with the high-dimensional and complex network traffic data for precise detection of intrusions in IoT systems. In comparison to the traditional activation functions, ELU generates extremely large gradients that help in generalizing the Bi-LSTM network on the unseen IoT data. On the other hand, the inclusion of OOA within the modified Bi-LSTM network reduces its complexity and processing time by selecting the optimal set of features from the datasets. In IDS, the network latency is a communication pattern which is considers features to know about the traffic, protocol type, and frequency of communication. Relatively, the network latency is lower in IoT networks compared to the network with attack traffic. So, abnormal value of network latency represents the network has the attack traffic. Therefore, it is considered a crucial parameter of feature across all the datasets such as N-BaIoT, CICIDS-2017, and ToN-IoT for a precise identification of intrusion.

The developed OOA based feature selection is used to discover the relevant features from the overall feature set. The ability of

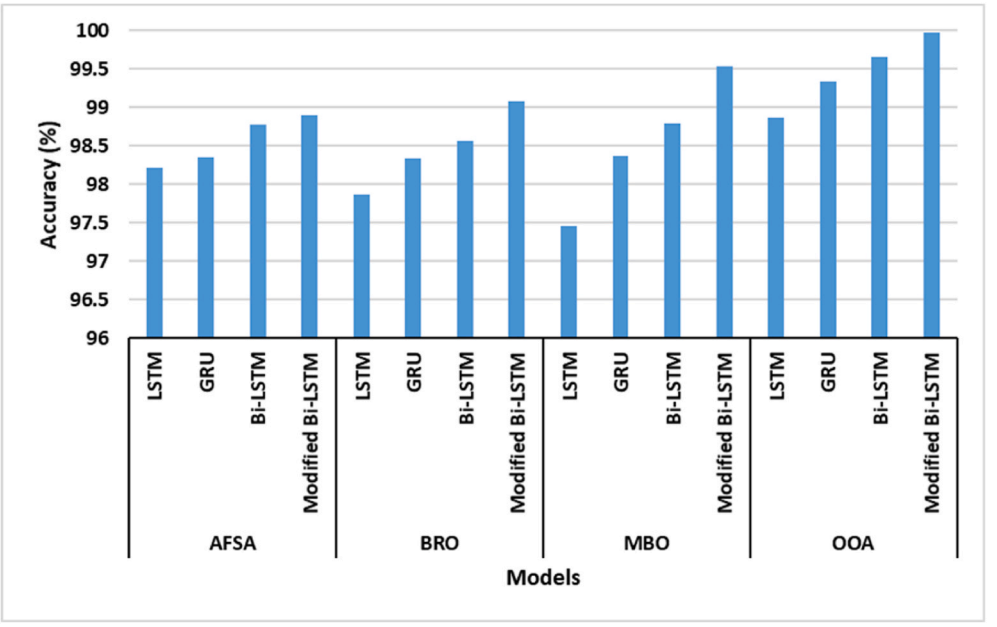

Fig. 3. Accuracy comparison for the N-BaIoT dataset.

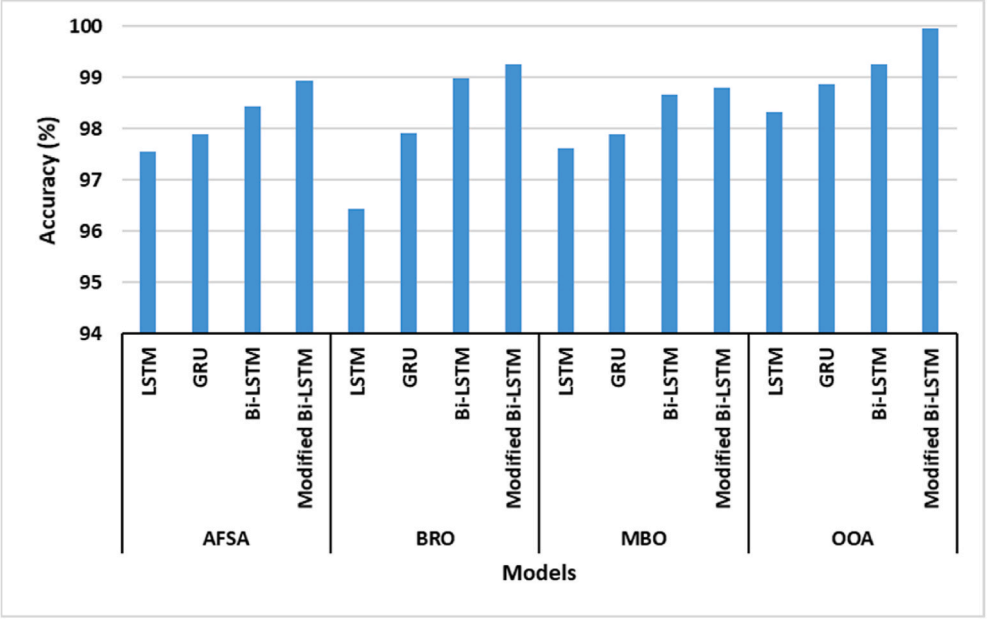

Fig. 4. Accuracy comparison for the CICIDS-2017 dataset.

avoiding the local optima risk and obtaining the global optimum solution of the OOA is used to discover the highly informative features. Therefore, the identification of highly informative features is used to differentiate the normal and attack traffic that helps to minimize the risk of FP and FN. The Bi-LSTM is efficient in acquiring the long-term dependencies in the IoT data i.e., network traffic patterns. The bidirectional processing of input sequences makes the Bi-LSTM efficiently obtain the contextual data and temporal dependencies which span over multiple time steps. This helps to discover hidden, time varying behavior of both normal and attack that leads to reduce the FP and FN. Moreover, the integration of ELU in Bi-LSTM supports for the learning of complex and nonlinear relationships among the input and target. The incorporated ELU preserves both the positive and negative values that are used to prevent saturation and ensure the network for learning highly informative depictions of input. This reliability in learning the complex patterns is used to enhance the differentiation among the normal and attack network traffic which supports in reducing FP and FN.

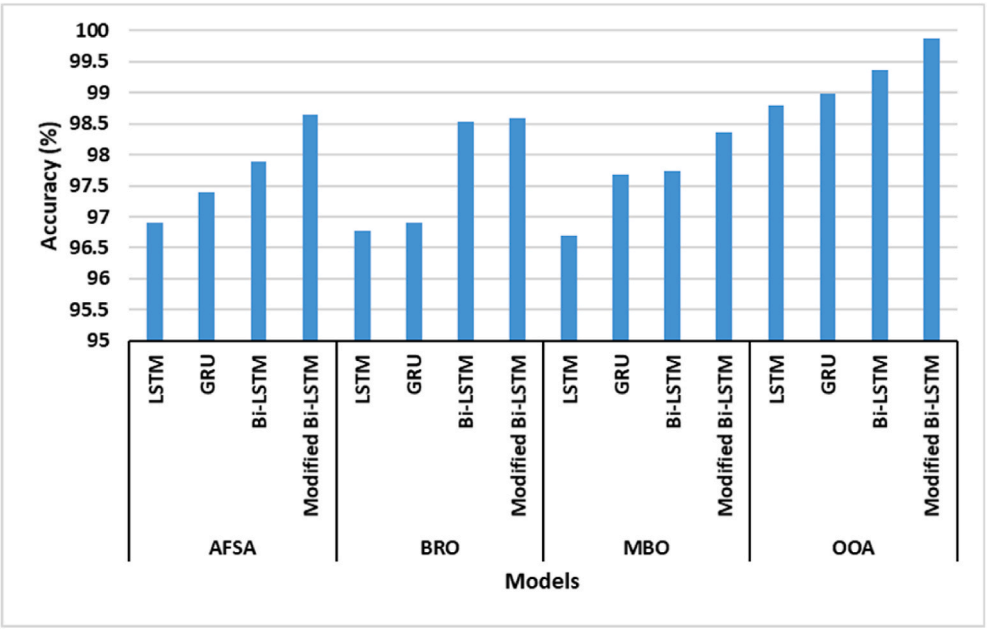

Fig. 5. Accuracy comparison for the ToN-IoT dataset.

As shown in Table 7, the scientific contribution of OOA with modified Bi-LSTM network is evaluated by analyzing its results with different K-fold validations (K = 2, 3, 5, and 8). By viewing Table 7 and it is evident that the OOA based modified Bi-LSTM network obtains improved classification during five-fold cross validation (i.e., training data is 80 % and testing data is 20 %). Intrusion detection is one of the vital tasks in cybersecurity application, and K-fold cross validation is a standard technique used for ensuring that the OOA and modified Bi-LSTM network is capable, reliable, and robust in generalizing the unseen IoT data. Additionally, K-fold cross validation helps in detecting overfitting problems by analysing the OOA based modified Bi-LSTM network's performance on various validation sets. The proposed framework has limited overfitting on the training data when it consistently performs well on dissimilar validation sets. Furthermore, the processing time of the modified Bi-LSTM network is reduced by choosing these discriminative and relevant features, as depicted in Table 8. The proposed framework consumes a short processing time of 8.02, 5.33, and 6.24 s on the N-BaIoT, CICIDS-2017, and ToN-IoT datasets, respectively. Therefore, it is superior in comparison to other classification models like LSTM, GRU, and Bi-LSTM.

4.2. Comparative analysis

The scientific contribution of the proposed framework, OOA based modified Bi-LSTM network is validated by evaluating its outcomes with existing frameworks designed by Keserwani et al. [25], Hassan et al. [26], Om Kumar et al. [31], and Elsayed et al. [34]. Keserwani et al. [25] combined PSO and GWO algorithms for eliminating irrelevant, unnecessary, and inappropriate features from the CICIDS-2017 dataset. The appropriate/relevant features were then passed into the random forest classifier for differentiating between normal and attack traffic. The developed framework, PSO-GWO-random forest achieves 99.88 % of detection accuracy on the CICIDS-2017 dataset. Correspondingly, Hassan et al. [26] used the random forest classifier with an improved MRFO algorithm for accurate identification of intrusion attacks. This framework achieves an impressive detection accuracy of 99.30 % on the CICIDS-2017 dataset.

Kumar et al. [31] employed the min-max normalization technique, IBRO algorithm, and a recurrent kernel CNN model with the MMBO algorithm for precise intrusion detection. The presented framework obtains 99.95 % and 99.96 % of detection accuracy on the CICIDS-2017 and N-BaIoT datasets. Furthermore, Elsayed et al. [34] developed an improved LSTM network for intrusion detection in software defined networks and IoT systems. The improved LSTM network obtains a detection accuracy of 96.35 % on the ToN-IoT dataset. In comparison to these four existing frameworks, the proposed framework, OOA based modified Bi-LSTM network achieves higher detection accuracy on all four datasets, as depicted in Table 9.

4.3. Discussion

Feature selection and classification of intrusion attacks are crucial steps in this paper. Initially, IoT data acquired from N-BaIoT, CICIDS-2017, and ToN-IoT datasets is rescaled using the min-max normalization technique. The discriminative and relevant features are then selected from the pre-processed IoT data utilizing OOA, where it has a better convergence rate and searching ability in comparison to other algorithms such as AFSA, BRO, and MBO. The selected discriminative and relevant features are passed into the

Table 7
Achieved results of the OOA-based modified Bi-LSTM network under various K-fold validations.

Datasets	Measures (%)	K = 2	K = 3	K = 5	K = 8
N-BaIoT	F1-score	94.50	96.46	**99.89**	92.58
	Accuracy	94.12	94.40	**99.98**	94.06
	Specificity	96.28	95.96	**99.90**	94.96
	Recall	94.34	94.78	**99.94**	94.78
	Fall-out	95.70	94.58	**99.95**	95.92
CICIDS-2017	F1-score	94.58	95.62	**99.96**	93.94
	Accuracy	95.68	94.26	**99.97**	94.64
	Specificity	94.03	94.16	**99.92**	95.35
	Recall	95.88	94.48	**99.91**	96.66
	Fall-out	95.79	94.32	**99.94**	94.58
ToN-IoT	F1-score	92.28	92.58	**99.90**	95.16
	Accuracy	93.68	93.32	**99.88**	94.44
	Specificity	92.58	92.18	**99.92**	95.66
	Recall	94.05	94.18	**99.89**	95.28
	Fall-out	93.36	95.72	**99.91**	96.86

Table 8
Processing time of different classifiers on all three datasets.

Processing time (seconds)			
Classifiers	N-BaIoT	CICIDS-2017	ToN-IoT
LSTM	11.20	10.34	11.24
GRU	9.48	9.50	10.27
Bi-LSTM	9.22	8.32	8.46
Modified Bi-LSTM	8.02	5.33	6.24

Table 9
Achieved results of proposed and existing frameworks.

Frameworks	Dataset	Detection accuracy (%)
PSO-GWO-random forest [25]	CICIDS-2017	99.88
Improved MRFO-random forest [26]	CICIDS-2017	99.30
MMBO-recurrent kernel CNN [31]	CICIDS-2017	99.95
	N-BaIoT	99.96
Improved LSTM [34]	ToN-IoT	96.35
OOA-modified Bi-LSTM	N-BaIoT	99.98
	CICIDS-2017	99.97
	ToN-IoT	99.88

modified Bi-LSTM network for intrusion classification. In the modified Bi-LSTM network, the typical activation functions (sigmoid and tangent) are changed with the ELU activation function. This process assists the conventional Bi-LSTM network in learning a wider range of data representations by capturing both negative and positive values in the datasets. The ELU activation function mitigates overfitting and vanishing gradient problems in the Bi-LSTM network with better detection accuracy. The efficiency of the OOA based modified Bi-LSTM network is stated in Tables 4–9. In the real time environment, the proposed OOA based modified Bi-LSTM network is suitable to be integrated with the existing security systems, network infrastructures, and hardware for efficient processing. Additionally, monitoring and logging functionalities is suitable to be implemented for tracking the outcome of the proposed IDS in real time environments. It is vital to identify issues and ensure the reliability of the proposed IDS.

5. Conclusion and future work

In this paper, an efficient framework is proposed for intrusion detection in IoT systems. The proposed framework named OOA based modified Bi-LSTM network is implemented using the python programming language. It comprises three phases, namely preprocessing of IoT data, feature selection, and the classification of benign and attack traffics. After rescaling the acquired IoT data using min-max normalization technique, important/relevant features are selected utilizing OOA. The discriminative feature selection by OOA reduces the framework's complexity and processing time. These selected discriminative features are then fed into the modified Bi-LSTM network for precise classification of benign and attack traffics. An integration of ELU in the Bi-LSTM enhances the generalization ability by minimizing the overfitting risk, avoiding vanishing gradient issue and fastening the training process. In comparison to existing frameworks, the proposed framework obtains an impressive detection accuracy of 99.98 %, 99.97 % and 99.88 % on the N-BaIoT, CICIDS-2017, and ToN-IoT datasets, respectively. Furthermore, the proposed framework consumes minimal processing time of

8.02, 5.33, and 6.24 s on the N-BaIoT, CICIDS-2017, and ToN-IoT datasets, respectively.

As a future extension, the proposed framework is suitable to be implemented in a real time learning environment for precise intrusion detection to protect IoT systems. An effective distillation method is also useable to develop network IDS for deployment in IoT devices.

Dataset

N-BaIoT dataset: https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset.
CICIDS2017 dataset: https://www.kaggle.com/datasets/cicdataset/cicids2017.
ToN-IoT dataset: https://www.kaggle.com/datasets/amaniabourida/ton-iot.

CRediT authorship contribution statement

Siva Surya Narayana Chintapalli: Data curation, Conceptualization. **Satya Prakash Singh:** Funding acquisition, Formal analysis. **Jaroslav Frnda:** Methodology, Investigation. **Parameshachari Bidare Divakarachari:** Project administration, Funding acquisition. **Vijaya Lakshmi Sarraju:** Writing – review & editing, Visualization, Validation. **Przemysław Falkowski-Gilski:** Writing – original draft, Software, Resources.

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.

Acknowledgement

The research was co-funded by the European Union within the REFRESH project - Research Excellence For REgion Sustainability and High-tech Industries ID No. CZ.10.03.01/00/22_003/0000048 of the European Just Transition Fund and also by the Ministry of Education, Youth and Sports of the Czech Republic within the project SGS ID No. SP2024/061 conducted by VSB-Technical University of Ostrava.

References

[1] M. Vishwakarma, N. Kesswani, DIDS: a deep neural network based real-time intrusion detection system for IoT, Decision Analytics Journal 5 (2022) 100142.
[2] B. Madhu, M.V.G. Chari, R. Vankdothu, A.K. Silivery, V. Aerranagula, Intrusion detection models for IOT networks via deep learning approaches, Measurement: Sensors 25 (2023) 100641.
[3] A. Kumar, K. Abhishek, M.R. Ghalib, A. Shankar, X. Cheng, Intrusion detection and prevention system for an IoT environment, Digital Communications and Networks 8 (4) (2022) 540–551.
[4] P. Nimbalkar, D. Kshirsagar, Feature selection for intrusion detection system in Internet-of-Things (IoT), ICT Express 7 (2) (2021) 177–181.
[5] P. Sanju, Enhancing intrusion detection in IoT systems: a hybrid Metaheuristics-deep learning approach with ensemble of recurrent neural networks, Journal of Engineering Research (2023) 100122.
[6] D. Mishra, B. Naik, J. Nayak, A. Souri, P.B. Dash, S. Vimal, Light gradient boosting machine with optimized hyperparameters for identification of malicious access in IoT network, Digital Communications and Networks 9 (1) (2023) 125–137.
[7] J. Ahmad, S.A. Shah, S. Latif, F. Ahmed, Z. Zou, N. Pitropakis, DRaNN_PSO: a deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things, Journal of King Saud University-Computer and Information Sciences 34 (10) (2022) 8112–8121.
[8] A. Davahli, M. Shamsi, G. Abaei, Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks, J. Ambient Intell. Hum. Comput. 11 (2020) 5581–5609.
[9] C. Prajisha, A.R. Vasudevan, An efficient intrusion detection system for MQTT-IoT using enhanced chaotic salp swarm algorithm and LightGBM, Int. J. Inf. Secur. 21 (6) (2022) 1263–1282.
[10] N.A. Alawad, B.H. Abed-alguni, M.A. Al-Betar, A. Jaradat, Binary improved white shark algorithm for intrusion detection systems, Neural Comput. Appl. (2023) 1–25.
[11] V. Prabhakaran, A. Kulandasamy, mLBOA-DML: modified butterfly optimized deep metric learning for enhancing accuracy in intrusion detection system, Journal of Reliable Intelligent Environments (2023) 1–15.
[12] S. Tharewal, M.W. Ashfaque, S.S. Banu, P. Uma, S.M. Hassen, M. Shabaz, Intrusion detection system for industrial Internet of Things based on deep reinforcement learning, Wireless Commun. Mobile Comput. 2022 (2022) 1–8.
[13] M.A. Siddiqi, W. Pak, Tier-based optimization for synthesized network intrusion detection system, IEEE Access 10 (2022) 108530–108544.
[14] J. Liu, D. Yang, M. Lian, M. Li, Research on intrusion detection based on particle swarm optimization in IoT, IEEE Access 9 (2021) 38254–38268.
[15] Y. Badr, Enabling intrusion detection systems with dueling double deep Q-learning, Digital Transformation and Society 1 (1) (2022) 115–141.
[16] P.M. Vijayan, S. Sundar, Hybrid MQTTNet: an intrusion detection system using heuristic-based optimal feature integration and hybrid Fuzzy with 1DCNN, Cybern. Syst. (2022) 1–34.
[17] G. Rohini, C. Gnana Kousalya, J. Bino, Intrusion detection system with an Ensemble learning and feature selection framework for IoT networks, IETE J. Res. (2022) 1–17.
[18] F. Hosseini, F.S. Gharehchopogh, M. Masdari, A botnet detection in IoT using a hybrid multi-objective optimization algorithm, New Generat. Comput. 40 (3) (2022) 809–843.
[19] T.S. Naseri, F.S. Gharehchopogh, A feature selection based on the farmland fertility algorithm for improved intrusion detection systems, J. Netw. Syst. Manag. 30 (3) (2022) 40.
[20] M. Samadi Bonab, A. Ghaffari, F. Soleimanian Gharehchopogh, P. Alemi, A wrapper-based feature selection for improving performance of intrusion detection systems, Int. J. Commun. Syst. 33 (12) (2020) e4434.
[21] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, A.Y. Zomaya, An explainable deep learning-enabled intrusion detection framework in IoT networks, Inf. Sci. 639 (2023) 119000.

[22] H.C. Altunay, Z. Albayrak, A hybrid CNN+LSTM based intrusion detection system for industrial IoT networks, Engineering Science and Technology, an International Journal 38 (2023) 101322.
[23] F.G. Gebretsadik, S. Nayak, R. Patgiri, eBF: an enhanced Bloom Filter for intrusion detection in IoT, Journal of Big Data 10 (1) (2023) 102.
[24] A. Alsirhani, M.M. Alshahrani, A.M. Hassan, A.I. Taloba, R.M. Abd El-Aziz, A.H. Samak, Implementation of African vulture optimization algorithm based on deep learning for cybersecurity intrusion detection, Alex. Eng. J. 79 (2023) 105–115.
[25] P.K. Keserwani, M.C. Govil, E.S. Pilli, P. Govil, A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO-PSO-RF model, Journal of Reliable Intelligent Environments 7 (2021) 3–21.
[26] I.H. Hassan, M. Abdullahi, M.M. Aliyu, S.A. Yusuf, A. Abdulrahim, An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection, Intelligent Systems with Applications 16 (2022) 200114.
[27] R. Anushiya, V.S. Lavanya, A new deep-learning with swarm based feature selection for intelligent intrusion detection for the Internet of things. Measurement, Sensors 26 (2023) 100700.
[28] M. Alweshah, A. Hammouri, S. Alkhalaileh, O. Alzubi, Intrusion detection for the internet of things (IoT) based on the emperor penguin colony optimization algorithm, J. Ambient Intell. Hum. Comput. 14 (5) (2023) 6349–6366.
[29] M. Alweshah, S. Alkhalaileh, M. Beseiso, M. Almiani, S. Abdullah, Intrusion detection for IoT based on a hybrid shuffled shepherd optimization algorithm, J. Supercomput. 78 (10) (2022) 12278–12309.
[30] Y. Li, S.M. Ghoreishi, A. Issakhov, Improving the accuracy of network intrusion detection system in medical IoT systems through butterfly optimization algorithm, Wireless Pers. Commun. 126 (3) (2022) 1999–2017.
[31] C.U. Om Kumar, S. Marappan, B. Murugeshan, P.M.R. Beaulah, Intrusion detection model for IoT using recurrent kernel convolutional neural network, Wireless Pers. Commun. 129 (2) (2023) 783–812.
[32] A. Dahou, M. Abd Elaziz, S.A. Chelloug, M.A. Awadallah, M.A. Al-Betar, M.A. Al-Qaness, A. Forestiero, Intrusion detection system for IoT based on deep learning and modified reptile search algorithm, Comput. Intell. Neurosci. 2022 (2022).
[33] A. Li, S. Yi, Intelligent intrusion detection method of industrial Internet of things based on CNN-BiLSTM, Secur. Commun. Network. 2022 (2022).
[34] R.A. Elsayed, R.A. Hamada, M.I. Abdalla, S.A. Elsaid, Securing IoT and SDN systems using deep-learning based automatic intrusion detection, Ain Shams Eng. J. 14 (10) (2023) 102211.
[35] M. Abd Elaziz, M.A. Al-qaness, A. Dahou, R.A. Ibrahim, A.A. Abd El-Latif, Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm, Adv. Eng. Software 176 (2023) 103402.
[36] L. Yi, M. Yin, M. Darbandi, A deep and systematic review of the intrusion detection systems in the fog environment, Transactions on Emerging Telecommunications Technologies 34 (1) (2023) e4632.
[37] F.S. Gharehchopogh, B. Abdollahzadeh, S. Barshandeh, B. Arasteh, A multi-objective mutation-based dynamic Harris Hawks optimization for botnet detection in IoT, Internet of Things 24 (2023) 100952.
[38] H. Asgharzadeh, A. Ghaffari, M. Masdari, F.S. Gharehchopogh, Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm, J. Parallel Distr. Comput. 175 (2023) 1–21.
[39] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-baiot-network-based detection of iot botnet attacks using deep autoencoders, IEEE Pervasive Computing 17 (3) (2018) 12–22.
[40] M.G. Karthik, M.M. Krishnan, Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks, J. Ambient Intell. Hum. Comput. (2021) 1–11.
[41] D. Stiawan, M.Y.B. Idris, A.M. Bamhdi, R. Budiarto, CICIDS-2017 dataset feature analysis with information gain for anomaly detection, IEEE Access 8 (2020) 132911–132921.
[42] Z.K. Maseer, R. Yusof, N. Bahaman, S.A. Mostafa, C.F.M. Foozy, Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset, IEEE Access 9 (2021) 22351–22370.
[43] A.R. Gad, A.A. Nashat, T.M. Barkat, Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset, IEEE Access 9 (2021) 142206–142217.
[44] A.R. Gad, M. Haggag, A.A. Nashat, T.M. Barakat, A Distributed intrusion detection system using machine learning for IoT based on ToN-IoT dataset, Int. J. Adv. Comput. Sci. Appl. 13 (6) (2022).
[45] J.K. Samriya, R. Tiwari, X. Cheng, R.K. Singh, A. Shankar, M. Kumar, Network intrusion detection using ACO-DNN model with DVFS based energy optimization in cloud framework, Sustainable Computing: Informatics and Systems 35 (2022) 100746.
[46] M.A. Siddiqi, W. Pak, An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection, IEEE Access 9 (2021) 137494–137513.
[47] M. Dehghani, P. Trojovský, Osprey optimization algorithm: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems, Front. Mech. Eng. 8 (2023) 1126450.
[48] H. Dai, Y. Wang, C. Xu, Osprey: a heterogeneous search framework for spatial-temporal similarity, Computing 104 (9) (2022) 1949–1975.
[49] J. Gao, Network intrusion detection method combining CNN and BiLSTM in cloud computing environment, Comput. Intell. Neurosci. 2022 (2022).
[50] P.V. Sontakke, N.B. Chopade, Hybrid DNN-BiLSTM-aided intrusion detection and trust-clustering and routing-based intrusion prevention system in VANET, Journal of Control and Decision (2023) 1–18.
[51] Z. Ma, J. Li, Y. Song, X. Wu, C. Chen, Network intrusion detection method based on FCWGAN and BiLSTM, Comput. Intell. Neurosci. 2022 (2022).
[52] Z. Xiang, X. Li, Fusion of transformer and ML-CNN-BiLSTM for network intrusion detection, EURASIP J. Wirel. Commun. Netw. 2023 (1) (2023) 71.