International Journal of Intelligent Robotics and Applications (2022) 6:323–349
https://doi.org/10.1007/s41315-022-00224-4

REGULAR PAPER

Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM

V. Raghava Swamy Dora[1] · **V. Naga Lakshmi**[1]

Received: 30 September 2021 / Accepted: 8 January 2022 / Published online: 27 January 2022

Abstract
Nowadays, the digital era is reshaped by new technologies, and the cyber-attacks are more sophisticated and becoming as a commonplace. The distributed denial of service (DDoS) attacks are the exponentially-growing and major prevalent attack that targets the emerging and changing computational network infrastructures around the globe. It is complex to distinguish the DDoS attack traffic from the legitimate network traffic when the transit happens from the zombies or attacker to the victim. The DDoS attack is considered as a stubborn network security conflict. Yet, these algorithms need a priori knowledge regarding the classes, and it is not possible to adapt to the subsequent varying network traffic trends in an automatic manner. This creates the requirement for the enhancement of the novel DDoS detection mechanisms that in turn sophisticated and targets the DDoS attacks. The main intent of this paper is to implement the DDoS detection model through deep learning by the integration of convolutional neural network (CNN), and optimized long short-term memory (LSTM), so called CNN-O-LSTM. On the standard five benchmark datasets, the optimal feature selection is performed by the closest position-based grey wolf optimization (CP-GWO) with the consideration of minimizing the correlation among the features. With the optimally selected features, CNN is adopted for the feature learning process, from which the features of the second pooling layer are extracted, which is used for performing the detection. The adoption of optimally selected features with the CNN features enhances the detection performance with the most significant features. Finally, the optimized LSTM is used in the detection phase, which aims to maximize the detection accuracy by optimizing the hidden neurons of LSTM. The proposed DDoS detection scheme is experimented on a set of benchmark datasets, and the outcomes are compared over the traditional models.

Keywords DDoS attack detection · Optimal feature selection · CNN-feature learning · Optimized long short term memory · Closest position-based grey wolf optimization

1 Introduction

The network security present in the legacy network architectures depends on manual perimeter-oriented solutions (Yu et al. 2020). A high-level network security policy is implemented by configuring the network operators of every device with the help of the vendor-specific low-level commands. Yet, manual network security technology configuration likes IPSec technologies, intrusion detection/prevention systems (IDS/IPS), and firewalls are prone to inter- and intra- domain policy conflicts as well as configuration errors that lead to serious security threats and breaches (Hamed and Al-Shaer 2006). A study on the firewall configuration errors (Wool 2004) reveals that the rule sets are enforced by the corporate firewalls, and it breaches the familiar security guidelines. This occurs in every device because of the manual low-level configurations. Nowadays, the network security management seems to be much complicated. The primary network security principle is targeted by the DoS attack. The genuine users are prevented from the internet resource or usage of a particular web service (Ahmad et al. 2015). The vast attack packet traffic disrupts the connection among two systems. The services among system and particular system are interrupted. A legitimate user is prevented from the access of a particular service, website, or server. When the attack traffic is generated by multiple sources, then it is known as the DDoS attack (Kasim 2020). DDoS attacks compromised

✉ V. Raghava Swamy Dora
swamyraghava@gmail.com

V. Naga Lakshmi
nvadlama@gitam.edu

[1] Computer Science Department, GITAM Institute of Science, Visakhapatnam, India

several intermediate machines, which are present freely on the internet. These intermediate machines compose the zombies army referred as botnets (Shin et al. 2015). Majority of the attackers employed the botnet-oriented flooding attacks for launching the DDoS attacks. Nowadays, the technique of DDoS attacks has radically varied, and the protection against these attacks is more complicated. The network services are interrupted by modeling the DoS attacks. It degrades or blocks the resources employed by them (Zhang et al. 2019). One of the main problems for the DDoS detection techniques is the complexity in differentiating the legitimate and the DDoS attack packets. The attack traffic is criticized by the attackers within the legitimate traffic for hiding their attack. Hence, DDoS attacks are a major threat for the computers users (Zhang et al. 2019).

The IDS is an efficient technique for detecting the DDoS attack. The functional cloud services are also ensured (Gao et al. 2019). The computer attacks are detected by the IDS by analyzing the records that are gathered from the internet (Chen et al. 2007b). The IDS is classified into two forms such as the misuse or signature-oriented ID and anomaly-oriented ID. The attacker signatures are utilized by the signature-oriented detection technique, and it is available in the knowledge database for recognizing the attacks (Tan et al. 2015). It represents an efficient approach for detecting the known attacks (Bhuyan et al. 2014). The behavioral pattern is used by the anomaly-oriented technique using a period for measuring the appropriate patterns from the accepted characteristics (Bhuyan et al. 2014). A zero-day attack is detected by the anomaly detection. The deviations from the regular patterns are specified by the anomaly detection, in which the patterns are utilized by the signature detection and it is associated to the attacks for detecting the attacks (Zargar et al. 2013).

A major challenging task that exists in the early and effective DDoS attacks detection. An organization can be influenced by the DDoS attack at several levels that ranges from customer loss, prestige, and financial loss to the data exfiltration (Ravi and Shalinie 2020). Hence, an efficient DDoS protection system is needed to preserve the user loyalty, reputation, productivity, and revenue (Kasim 2020). Since DDoS attacks are becoming sophisticated and scaled, DDoS prevention seems to be a more challenging one. Machine learning (ML) algorithms are much applicable for learning the normal characteristics of traffic flows in an automatic format (Kushwah and Ranga 2020). Several analytic approaches were used like support vector machines (SVMs), decision trees (DTs), game theory, entropy analysis, artificial neural networks (ANNs), and hidden Markov modeling (HMM) (Bojović et al. 2019; Arun Raj Kumar and Selvakumar 2011).

The major contribution of this paper is as below.

- To implement the DDoS attack detection model with the help of CNN-O-LSTM by gathering the datasets from five standard publically available database such as, "DARPA1998 data set, DARPA LLS DDoS-1.0 dataset, CICIDS2017 dataset, NSL-KDD, KDD cup database".
- To perform the optimal feature selection on the five standard datasets for reducing the computational complexity with the intention of minimizing the correlation among the features.
- To accomplish the feature learning by the CNN, in which the features from the second pooling layer are extracted that is subjected to the detection process for detecting the DDoS attack.
- To perform the detection using the optimized LSTM, where the hidden neurons of LSTM are optimized for designing the univariate time series prediction issues with the consideration of maximizing the accuracy of DDoS attack detection.
- To develop a novel optimization algorithm called CP-GWO that attains a better convergence rate and less time complexity for enhancing the optimal feature selection process as well as the detection phase by optimizing the hidden neurons of OLSTM.

The organization of the paper is defined as below. Section 1 provides the introduction regarding the DDoS attack detection methods. The literature works of the DDoS attack detection methods are shown in Sect. 2. Section 3 discusses the intelligent model for DDoS detection using deep learning in computer networks. Section 4 portrays the optimal feature selection using CP-GWO. Section 5 explains the CNN feature learning with O-LSTM-based DDoS detection. Section 6 returns the results and discussions. Section 7 concludes the paper.

2 Literature survey

2.1 Related works

In 2020, Wang et al. (2020) have selected the multi layer perceptrons (MLP) for solving and describing the developed problem. The MLP was joined with sequential feature selection for choosing the optimal features in the training process, and a feedback mechanism was modeled for reconstructing the detector when the detection errors were perceived in a dynamic manner. In the final step, the efficiency of this technique was tested and differentiated with few existing works. The outcomes demonstrated that this technique could correct the detector and produce comparable detection behavior when it was accomplished in a poor manner. The practical deployment generated by the changeable traffic as well as the disabled features was not considered by the MLP.

It produced comparable detection behavior with the NSL-KDD dataset than the remaining works.

In 2016, Tabatabaie Nezhad et al. (2016a) have introduced a new DDoS and DoS detection algorithm, where the packets time series variance count was described with the help of the box-cox transformation. This choice created a better prediction on the basis of an ARIMA method. It also explored the error chaotic characteristics that were related to the time series. The non-chaotic, as well as the chaotic errors, were categorized by the local Lyapunov exponent. In the final step, the attack, as well as the normal traffics, was categorized on the basis of the defined rules. It joined the chaos-oriented analysis, ARIMA modeling, and box-cox-oriented pre processing, and defined rules were applied for enhancing the effectiveness of the DDoS/DoS detection.

In 2020, Singh et al. (2020) have addressed a distributed attack detection mechanism system known as a threshold-oriented collaborative attack detection (T-CAD) that mitigated and detected the DDoS attacks on the edge routers. The normalized router entropy was calculated by the T-CAD, and it was compared to several thresholds to effectively distinguish among the flash events, DDoS attack, and legitimate traffic. It was validated by accomplishing the experiments with INET and OMNeT++. It was better than the entropy-oriented DDoS attack detection mechanisms and traditional thresholds on distinct performance measures.

In 2020, Velliangiri and Pandey (2020) have developed an efficient fuzzy and Taylor elephant herd optimization (FT-EHO) that was motivated by the deep belief network (DBN) classifier for the DDoS attack detection. The rules learning was done by the fuzzy classifier and Taylor series. It was evaluated via rigorous computer simulations. Three standard benchmark databases known as the database 1, database 2, and KDD cup were employed during the simulations. The performance metrics considered were recall, precision, detection accuracy, and accuracy. It provided better recall, precision, detection rate, and accuracy than the remaining techniques. In 2020, Haider et al. (2020) have labeled a deep CNN ensemble framework for the effective detection of DDoS attack in the SDNs. It was evaluated on a traditional flow-oriented dataset under the established benchmarks. It has attained enhanced accuracy than state-of-the-art detection techniques.

In 2009, Chonka et al. (2009) have employed the network self-similarity for differentiating the legitimate self-similar as well as the DDoS flooding attack traffic present in the network. A novel algorithm known as the anomaly prediction algorithm was proposed, which forecasted the network traffic nature present in a dynamic system. A strange attractor returned to the steady state. It was considered as the bursty legitimate traffic. It diverged from the steady state, and this traffic was produced by the DDoS flooding attack. The DDoS traffic addressed the network traffic pattern. It also trained a neural network detector. The DDoS attack traffic was efficiently and accurately detected. Apart from detecting the attack traffic, it also filtered the attack during the transit.

In 2007, Chen et al. (2007a) have proposed a novel distributed technique for detecting the DDoS flooding attacks present at the traffic-flow level. It was applicable for the effective implementation that was functioned using the internet service providers (ISPs). Few traffic fluctuations could be detected at the gateways or the internet routers of the edge networks. Distributed change-point detection (DCD) architecture was proposed with the help of the change aggregation trees (CAT). The abrupt traffic variations were detected in the multiple network domains. It reduced the early flooding damages produced by the DDoS attack. It was constructed over attack-transit routers that functioned in a cooperative manner. Every ISP domain was composed of a CAT server for aggregating the flooding alerts. The final decision was made by the CAT domain servers. The policy conflicts were resolved at distinct ISP domains. The consensus or mutual trust was established by proposing a novel secure infrastructure protocol (SIP). The DCD system was simulated at the University of Southern California (USC) institute. Better detection accuracy was produced by the four network domains.

In 2020, Çakmakçı et al. (2020) have addressed a sequential, online, DDoS detection strategy that was applicable for the multivariate data. It employed a kernel-oriented learning algorithm, a Chi-square test, and the Mahalanobis distance. In the initial step, four statistical and four entropy-oriented features were extracted from the network flows. Next, the entropy features have used the kernel-oriented learning algorithm for detecting the input vectors, which were susceptible to be the DDoS. It did not assume any model for the DDos or network traffic. It adapted and built a feature dictionary that spanned the subspace of the normal characteristics. The Mahalanobis distance among the distribution of dictionary members and the suspicious vectors was computed for each T minutes. Consequently, the Mahalanobis distance was evaluated by the Chi-square test. It was subjected to the CICIDS2017 dataset, and the outcomes were compared with the traditional algorithms. It was better than the DDoS classification algorithms having an offline learning process.

In 2021, Tang et al. (2021) have developed a "novel AutoPedestrian scheme" for augmenting the pedestrian data and for determining the appropriate loss functions to achieve improved performance while detecting the pedestrian in crowded environment. The experimental analysis was carried out for proving the effectiveness of the suggested algorithm. In 2021, Zhou et al. (2021) have developed an approach for solving the security and authentication problems regarding the vehicles in VANET. This proposed method has included the "concept of identify-based encryption" for ensuring the access control for the vehicles and

the malicious packets were filtered using the deep learning approaches. The experimental results have shown that the enhanced performance of the proposed model was observed in validating under VANET in 6G communication systems.

In 2021, Jiang et al. (2021) have developed a segmentation approach using the ray-shooting model along with the LSTM-based network for enhancing the weak-signal neuronal structures and also for eliminating the background noises in the images. The comparative analysis with the segmented images has shown that the proposed model has secured high distance scores over the traditional methods. In 2019, Zeng et al. (2019) have developed a simple and efficient approach for obtaining the local metric map to detect the defocus blur areas based on the feature learning belongs to the ConvNets. The analysis has demonstrated the consistency of the suggested model when compared with the conventional techniques.

2.2 Review

DDoS attacks are continuously occurring since the computer and networking approaches of attackers also changing rapidly. Generally, diverse supervised DDoS detection technologies have been implemented. On the other hand, such methods need a priori knowledge of the classes and cannot adapt automatically for regularly altering the traffic trends of network. DDoS attacks are the most ubiquitous and expansion of rising attack, targeting the different and promising computational network environments around world. DDoS attack has raised as a security hazard for the services offered through ISP. This highlights the requirement to develop a new DDoS detection model. Numerous DDoS detection models are proposed that have diverse features and challenges, as given in Table 1. SBS-MLP (Wang et al. 2020) has proposed for improving the efficient, easily feasible, and interactive attack detection model. However, this method attains some detection errors. It cannot attain the global optimal features. ARIMA model (Tabatabaie Nezhad et al. 2016a) has developed for improving the detection accuracy and efficiently classifies non-chaotic and chaotic errors. It is limited to classification of attack and bursty traffic states. T-CAD (Singh et al. 2020) has reduced the cost and hardware devices. It attains low false alarm rate. This model affects the security of the network. FT-EHO (Velliangiri and Pandey 2020) achieves better recall, detection rate and better classification rate. This method is challenging due to the computational cost. CNN (Haider et al. 2020) obtains less computational complexity and high detection accuracy. It

Table 1 Features and challenges of traditional DDoS detection models

Author [citation]	Methodology	Features	Challenges
Wang et al. (2020)	Sequential backward selection (SBS)-MLP	It is efficient, easily feasible and interactive It gets better detection results	However, this method attains some detection errors It cannot attain the global optimal features
Tabatabaie Nezhad et al. (2016a)	ARIMA model	It improves the detection accuracy It efficiently classifies non-chaotic and chaotic errors	This model is limited for classification of attack and bursty traffic states
Singh et al. (2020)	T-CAD	It reduces the cost and hardware devices It attains low false alarm rate	This model affects the security of the network
Velliangiri and Mohan (2020)	FT-EHO	It also achieves better recall and detection rate It achieves better classification rate	This method is challenging due to the computational cost
Haider et al. (2020)	CNN	It obtains less computational complexity and high detection accuracy It improves the cost-effectiveness and the scalability	This model cannot be used in the large-scale distributed networks
Chonka et al. (2009)	Chaos theory	It efficiently detects and filters the attacks It improves the performance	More effort has to be given for choosing input parameters Highly complicated procedure, and the results are not much accurate
Chen et al. (2007a)	CAT	It detects the attacks faster It enhances the scalable performance	This approach is restricted for promoting the attack detection model as a real-time detection
Çakmakçı et al. (2020)	Enhanced kernel-based online anomaly detection (E-KOAD) algorithm	This method is appropriate for real-time applications It efficiently classifies the attacks	This model is challenging for the detection of DDoS attacks

improves cost-effectiveness and scalability. This model cannot be used in large-scale distributed networks. Chaos theory (Chonka et al. 2009) efficiently detects and filters the attacks and also improves the performance. Though, the method has to choose input parameters, and so it is complex process, and the results are not much accurate. CAT (Chen et al. 2007a) detects the attacks faster and enhances the scalable performance. On the other hand, this approach is restricted for promoting the attack detection model as a real-time detection. E-KOAD (Çakmakçı et al. 2020) is appropriate for real-time applications and efficiently classifies the attacks. However, it is challenging approach for the detection of DDoS attacks. These challenges are considered for developing a new DDoS attack detection model with network security.

3 Intelligent model for DDoS detection using deep learning in computer networks

3.1 DDoS detection model

The DDoS attacks drain the computing as well as the communication power of the network targets through the injection of a vast quantity of malicious traffic into them. Nowadays, the DDoS attacks boost up the attack traffic that attains tens or hundreds of GB bandwidth for every second. These are considered as the multiple attack sources that are included in an implicit manner. An attack is made by the attacker with the help of multiple sources that involve computers, IoT devices, and routers in a distributed environment. The malware infects these devices. An attacker tries to find the presence of any compromised network. With the help of these compromised networks, the target system is attacked by an attacker via the generation of requests or packet floods in a continuous manner for conquering the target system. When a DoS attack enters into legitimate systems that are preset in a distributed environment, it is known a DDoS attack. In the case of DDoS attack, a single system is targeted by multiple systems. When the messages are flooded to the target system, the services available in the system are stopped, and it is known as the zombies. Few forms of DDoS attacks is the Smurf attacks, UDP flood, PING flood, TCP SYN flood, IP spoofing, and flooding. The flooding in the DDoS attacks is constructed using multiple machines. The machine learning-oriented solutions have helped the researchers in detecting the DDoS attacks using the dynamic and complex patterns. The familiar machine learning techniques employed in the DDoS attack detection are the recurrent neural networks (RNNs), LSTM-NN, CNNs, etc. In these years, the well-performed techniques for detecting the intrusion detection with the DDoS and DoS attacks are dependent on the machine learning algorithms (Malipatil et al. 2020). The architectural view of the proposed intelligent model for DDoS detection is shown in Fig. 1.

The proposed intelligent model for the DDoS attack detection is composed of four phases namely, "data collection, optimal feature selection, feature learning, and detection". In the initial step, the dataset is gathered from five standards publically available database called the, "DARPA1998 data set, DARPA LLS DDoS-1.0 dataset, CICIDS2017 dataset, NSL-KDD, KDD cup database". For the collected datasets, the optimal features are selected in the second phase called the optimal feature selection. Here, the optimal features are chosen using the proposed CP-GWO with the consideration of minimizing the correlation among the features. These optimally chosen features are given to the detection using noven CNN-O-LSTM. The feature learning is accomplished with the help of CNN. Here, the features from the pooling layer 2 of CNN are extracted and subjected to the final classification layer using the O-LSTM instead of fully convolutional layer. Here the hidden neurons of LSTM are optimized using the same proposed CP-GWO with the intention of maximizing the accuracy. This O-LSTM detects the DDoS attack with high accuracy.

3.2 Description of datasets

Here, the dataset for the intelligent model of DDoS detection is gathered from five standard publically available database such as the, "DARPA1998 data set, DARPA LLS DDoS-1.0 dataset, CICIDS2017 dataset, NSL-KDD, KDD cup database". The description of each of these datasets is given below.

DARPA1998 data set: This dataset is gathered from the link, "https://www.kaggle.com/yashwanthkumbam/apaddos-dataset". It shows the traditional gap among the familiar attack patterns that involved PUSH-ACK and ACK flooding DDoS attacks. It can be employed by the IDS developers for enhancing the detection ratio using the detection modules.

DARPA LLS DDoS-1.0 dataset: This dataset is gathered from the link, "https://data.mendeley.com/datasets/jxpfjc64kr/1". This represents a SDN specific dataset that is produced with the help of mininet emulator. The network simulation is performed for ICMP traffic, UDP, and benign TCP as well as the malicious traffic that represents the collection of ICMP attack, UDP flood attack, and TCP Syn attack. The total number of features present here equals to 23, where few represent the switch extraction and the remaining are computed. The extracted features involve rx_bytes, tx_bytes, port number, destination IP, source IP, total duration, duration_nsec, duration_sec, byte_count, Packet_count, and Switch-id. The time and date is described by the dt field and it is transformed into number and a monitoring of flow occurs at an interval of 30 s. The computed features involve port bandwidth, data receiving rate, data transfer rate, total

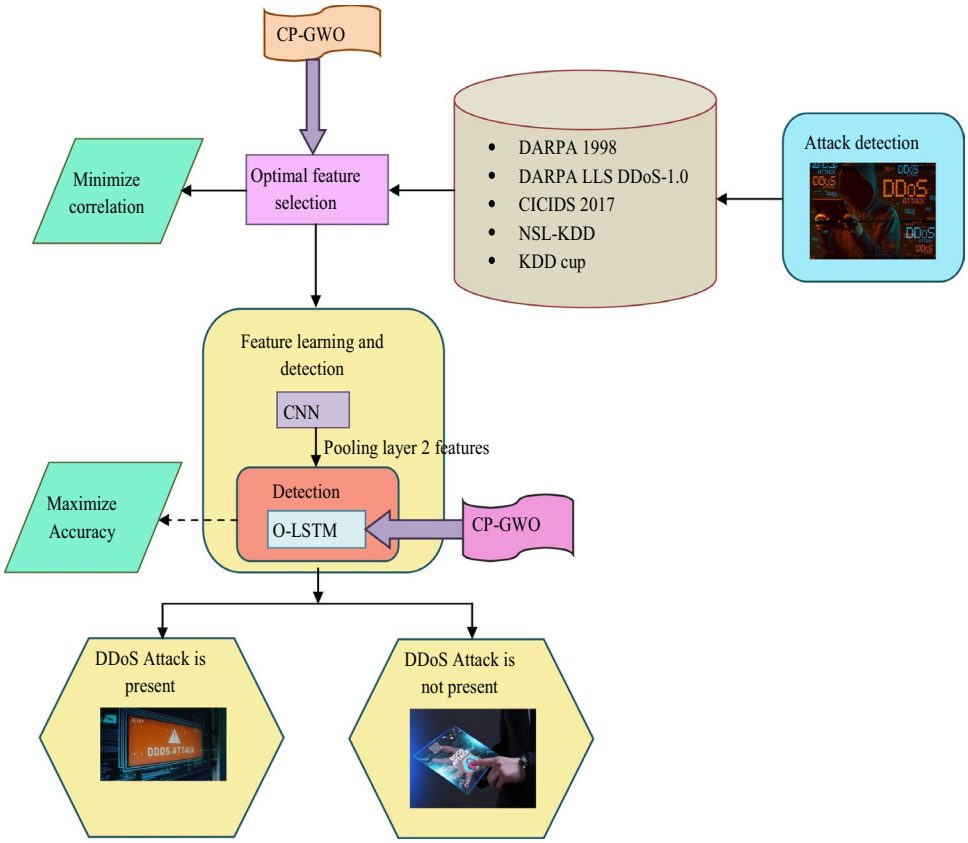

Fig. 1 Proposed intelligent model for DDos attack detection

flow entries, Packet_ins messages count, packet rate, byte per flow, and packet per flow. The class label defines the traffic type as malicious or benign. The simulations are performed for 250 min and 1,04,345 data rows are gathered. It is performed for the specified interval and multiple data can be gathered.

CICIDS2017 dataset: This dataset is gathered from the link, "https://www.kaggle.com/cicdataset/cicids2017". It is composed of the recent familiar and benign attacks that are similar to the real-world data like the PCAPs. It involves outcomes of the network traffic analysis by means of the CICFlowMeter having labelled flows on the basis of the attack, protocols, destination ports, source ports, destination IPs, source IPs, and time stamp. The definition of the extracted features is also present.

NSL-KDD dataset: This dataset is gathered from the link, "https://www.kaggle.com/hassan06/nslkdd". It is composed of the information regarding the network security, information security, and cyber security.

KDD cup dataset: This dataset is gathered from the link, "http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html". It is employed for the, "Third International Knowledge Discovery and Data Mining Tools Competition" that was performed together with the "KDD-99 the Fifth International Conference on Knowledge Discovery and Data Mining". It is also composed of a group of data for the audition process that involves broad method of intrusions being performed in a military network environment.

4 Optimal feature selection using closest position-based grey wolf optimization

4.1 Optimal feature selection

The optimal feature selection is used to choose the significant features for reducing the complexity by the proposed CP-GWO with the consideration of minimizing the correlation among the features. It selects the most relevant variables such that the final model is accurate, interpretable, and simpler. It returns less false alarm rate. It reduces the irrelevant features present in the data that minimizes the accuracy of the methods and converts the model to learn on the basis of irrelevant features. Here five datasets are considered. Assume the features from first dataset as Fs_{az}^{dt1} that is composed of 23 features, second dataset as Fs_{az}^{dt2} that consist of 23 features, third dataset as Fs_{az}^{dt3} that consist of 79 features, fourth dataset as Fs_{az}^{dt4} that consist of 43 features, and fifth dataset as Fs_{az}^{dt5} that consist of 42 features. Here, $az = 1, 2, \ldots, AZ$, in which AZ denotes the total number of features present in the five datasets. It is necessary to select the important features called optimal features for reducing

the computational complexity. This optimal feature selection is accomplished using the suggested CP-GWO. Hence, the optimally selected features are represented by $Fs_{az*}^{Optimal}$, where $az* = 1, 2, \ldots, AZ*$, in which $AZ*$ represents the optimally selected features. Here, five features from each dataset are optimally chosen by the proposed CP-GWO.

The major objective of the optimal feature selection is to minimize the correlation among the features by optimizing the features attained from the five datasets using the developed CP-GWO. Hence, the objective can be modelled as in Eq. (1).

$$OF1 = \underset{\{Fs_{az}^{dt1}, Fs_{az}^{dt2}, Fs_{az}^{dt3}, Fs_{az}^{dt4}, Fs_{az}^{dt5}\}}{\arg\max} \left(\frac{1}{Corrn}\right) \tag{1}$$

Here, the objective function of the optimal feature selection is given by $OF1$, the features from five datasets are defined by Fs_{az}^{dt1}, Fs_{az}^{dt2}, Fs_{az}^{dt3}, Fs_{az}^{dt4}, and Fs_{az}^{dt5}, and the correlation among the features is defined by $Corrn$. The correlation among two key points at and bt is given as in Eq. (2).

$$Corrn = \frac{nt\sum atbt - \sum at \sum bt}{\sqrt{\left(nt\sum at^2 - \left(\sum at\right)^2\right) - \left(nt\sum bt^2 - \left(\sum bt\right)^2\right)}} \tag{2}$$

In the above equation, the feature pair count is defined by nt respectively. The diagrammatic representation for the optimal feature selection is given in Fig. 2.

4.2 CP-GWO for DDoS detection

The proposed CP-GWO is used for enhancing the feature selection as well as the detection phases by optimizing the features of the collected datasets with the consideration of minimizing the correlation among the features as well as the hidden neurons of the LSTM with the intention of maximizing the accuracy. The GWO (Mirjalili 2014) mimics the hunting and the leadership nature of the grey wolves. The leadership hierarchy is motivated by four grey wolves such as, "alpha, beta, delta, and omega". The three steps in the hunting are the "searching for prey, encircling prey, and attacking prey". The apex predators are assumed as the grey wolves and it is in the first step of the food chain. It lives in the form of pack. A female is considered as the leader and

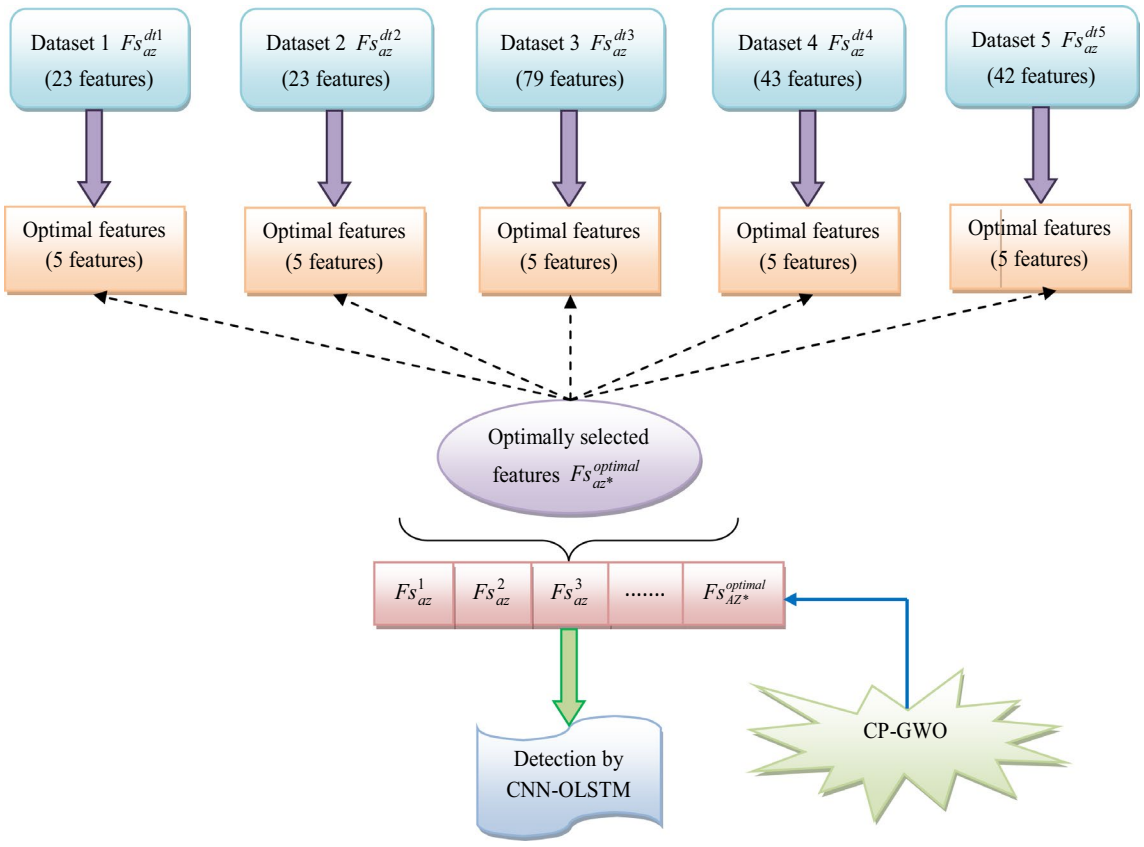

Fig. 2 Optimal feature selection process

the alphas are defined as the male. The decisions made by the alpha are the, "time to wake, sleeping place, hunting, etc.". Beta is the second level. This subordinate wolf helps in the pack or the decision activities of the alpha. The last rank for the grey wolf is given to the omega. It behaves similar to the scapegoat. It is subjected to the complete dominant wolves. The other wolves are taken as the delta or the subordinate wolf. It is responsible only to the alpha and beta. This category involves "caretakers, hunters, elders, sentinels, and scouts". The grey wolf encircles the prey in the hunting process. The encircling characteristics are designed as in Eqs. (3) and (4), where $o = 1, 2, \ldots, OE$, OE is the total population size.

$$D\vec{W}_o = \left| C\vec{W} * X\vec{W}_{pw}(tw) - X\vec{W}_o(tw) \right| \tag{3}$$

$$X\vec{W}_o(tw+1) = X\vec{W}_{pw}(tw) - A\vec{W} * D\vec{W}_o \tag{4}$$

In the above equation, the coefficient vectors are shown by $A\vec{W}$ and $C\vec{W}$, the position vector of a grey wolf is shown by $X\vec{W}$, the present iteration is shown by tw, and the position vector of the prey is shown by $X\vec{W}_{pw}$ respectively. The operation "$*$" indicates the Hadamard product. The coefficient vectors are measured as in Eqs. (5) and Eq. (6).

$$A\vec{W} = 2a\vec{w} * r\vec{w}_1 - a\vec{w} \tag{5}$$

$$C\vec{W} = 2 * r\vec{w}_2 \tag{6}$$

Here, the term $a\vec{w}$ is reduced from 2 to 0 and the random vectors in [0,1] is shown by rw_1 and rw_2 respectively. The updating procedure of existing GWO is based on the alpha wolf (best solution), beta wolf (second best solution), and delta wolf (third best solution). As a modification to the existing GWO, the proposed model updates the solution by finding the distance between the best solution, over all other solutions. Further, the average of the distance function based on Eq. (7) is used in gathering a condition for solution update.

$$Dis_o = \frac{1}{OE} \sum_{o=1}^{OE} \left(X\vec{W}_{pw} - X\vec{W}_o \right) \tag{7}$$

The solution update is done by counting the total number of distance that is less than the mean distance. Those solutions are further updated using Eq. (4), and the averaging over all solution will provide the final updated solution based on Eq. (8), where OE^* is the number of solutions, whose distance over the best solution is less than the mean distance.

$$X\vec{W}_o(tw+1) = \frac{1}{OE^*} \sum_{o=1}^{OE^*} XW_o \tag{8}$$

The traditional GWO returns various benefits like better performance on the unconstrained and constrained problems, handles the real problem, better performance in unknown, challenging search spaces, etc. Still, it limits in attaining faster convergence rate. Therefore, the algorithm is modified by concentrating only the closest position, and the remaining positions are skipped, thus called as CP-GWO. This proposed CP-GWO offers several benefits such as reaching better convergence rate, less time consumption, etc. The pseudo code of the proposed CP-GWO is given in Algorithm 1, and the flowchart of the proposed CP-GWO is given in Fig. 3.

Algorithm 1: Proposed CP-GWO

Start
Grey wolf population initialization
Parameter initialization
Fitness calculation
$X\vec{W}_{pw}$ as best search agent
Compute the distance between best solution and other all solutions
Compute mean of distance Dis_o by Eq. (7)
The total number of solutions that are less than the Dis_o is considered as the OE^*
For $kw \rightarrow 1$ *to* OE^*
  Find the parameters such as rw_1, rw_2, AW_2, CW_2, $D\vec{W}_o$, $X\vec{W}_o(tw+1)$
End for
Update the solution using Eq. (8)
Return the best solution
Stop

5 CNN feature learning with optimized LSTM-based DDoS detection

5.1 Proposed CNN-O-LSTM

The proposed CNN-OLSTM is used for detection of DDoS attack from computer networks. The proposed model includes the CNN as it reduces the computational complexity and ensures the scalability and cost-effectiveness in the DDoS attack detection, whereas in other methods like VGG and ResNet methods are not utilized for feature learning. The VGG network performs very slowly while training the data and the assigned weights in the network are quite larger. Similarly, the computational speed relies highly on the implementation part in Resnet. Here, the CNN is used for the feature learning and OLSTM is connected at last layer for detection. The feature learning represents a group of approaches that permits the system in discovering the

representations required for the detection from the raw data in an automatic manner. Here, the features are learned with unlabeled input data. CNN (Namatēvs 2017) describes the feedforward network. The parameters used for the learning process are reduced. The three concepts utilized are the "local receptive fields, shared weights, and temporal or spatial sampling". It is composed of several layers such as convolutional layers. These convolutional layers are constructed of small kernels for extracting the high level features. If minimum parameter count is employed for the learning, then the connections are low and the process of training is simpler. The filters are understood in a data-driven fashion for the feature extraction, where the inputs are shown by the features. It consists of an, "input layer, alternating convolutional layers, pooling or sub-sampling layers, and non-linear layers". A convolutional net, otherwise called as single convolutional layer involves, "convolutional stage, detector stage, and pooling stage". Thus, each convolutional layer includes multiple stages. The layers in the CNN are defined as follows.

Input layer: In this section, the input to the CNN describes the optimal features from the proposed CP-GWO.

Convolutional layers or convolutional stage: It is the major building block. The filter size is given by the kernel

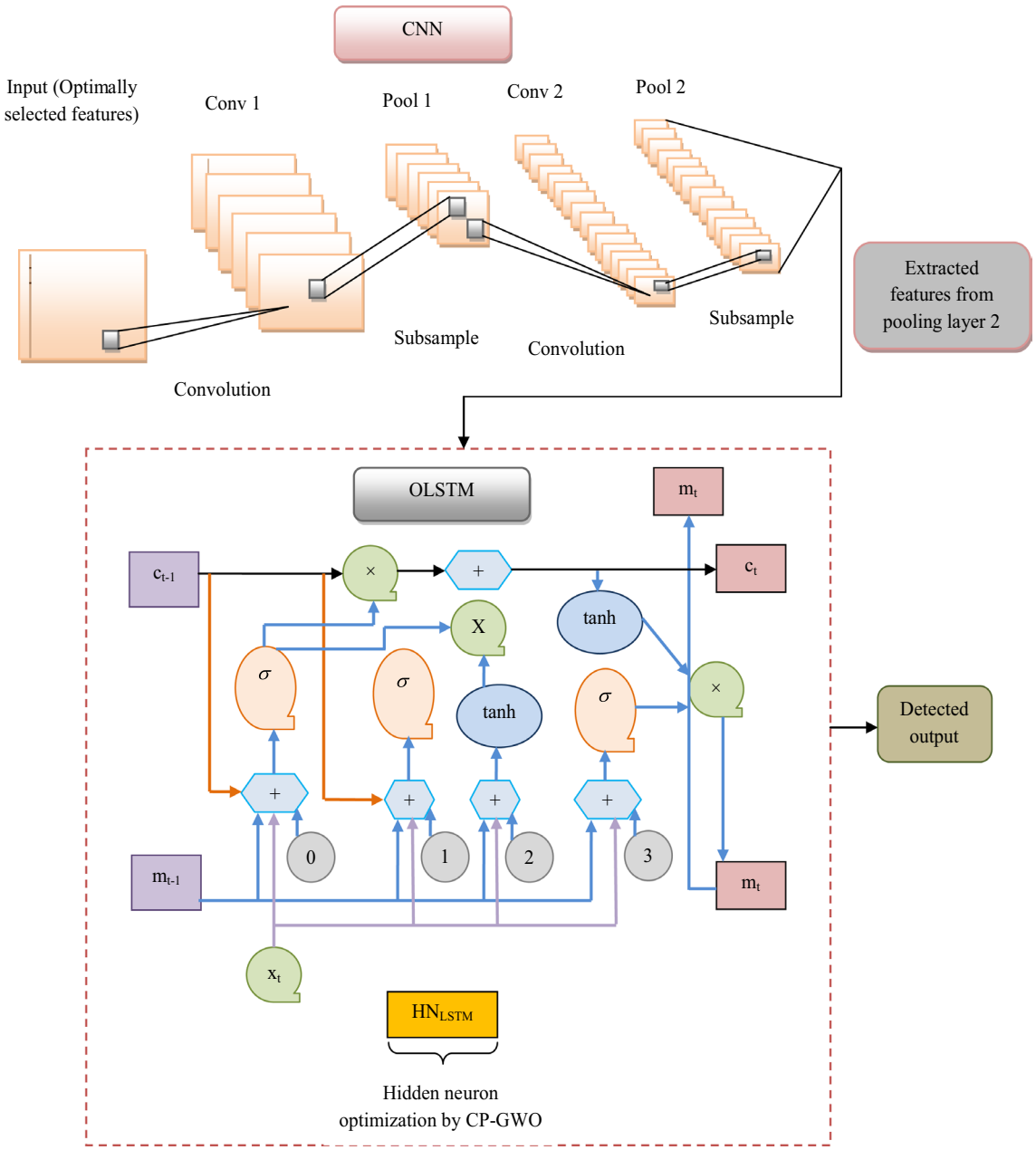

Fig. 3 Proposed CNN-O-LSTM for the DDoS attack detection

size and it is present outside the feature map. The padding is also an important feature here.

Non-linear layers or detector stage: Here, the nonlinear activation function detects each linear activation. Several complex models are learnt. The optimal way to design a neuron output fr as a function of its input xr having Rectified Linear Unit (ReLU), $sigmoid(xr)$, or $fr(xr) = \tanh(xr)$. The function $yr = \max(xr, 0)$ is utilized by the ReLU. It enhances the nonlinear properties associated with the decision function.

Pooling or subsampling or downsampling layers: The computational complexity, as well as the compressing features, is reduced by the previous feature map resolution. The features representing robustness to the disorder and noise can be adjusted. It is robust to the variations that are produced by the existing learned features. Here, the important patterns are focused with the help of the network. It also creates the downsampled versions with the input map. The inputs are divided into regions with size $RR \times RR$. It, in turn returns single output from each region. If an output with size $WR \times WR$ is subjected to the pooling layer, the output size PR is attained as in Eq. (9).

$$PR = \left\lfloor \frac{WR}{RR} \right\rfloor \tag{9}$$

The pooling may be, "max pooling, average of a rectangular neighbourhood, and pooling by downsampling". It also reduces the feature map size. The max pooling labels the invariance. Using each output map, the convolution is combined with multiple input maps as in Eq. (10).

$$xr_{jr}^{LR} = fr\left(\sum_{ir \in Fs_{az*}^{Optimal}} xr_{jr}^{LR-1} * kr_{irjr}^{LR} + br_{jr}^{LR}\right) \tag{10}$$

Here, the input map solution is shown by $Fs_{az*}^{Optimal}$, the convolutional layer is shown by LR, the downsampling layer is shown by $LR-1$, the input features of $LR-1$ convolutional layer is shown by xr^{LR-1}, the input is shown by ir, the output is shown by jr, kernel maps of LR convolutional layer is shown by kr_{irjr}, and the additive bias of LR convolutional layer is shown by br^{LR} respectively. Usually, the feature extraction includes distinct identical steps with three cascading layers such as, "convolution layer, activation layer, and pooling function".

Here, the features from the second pooling layer are extracted and subjected to the final phase called detection that is accomplished using the O-LSTM instead of fully connected layer. The LSTM (Abbasi et al. 2019) is composed of components such as, candidate layer CP NN having tanh, forget gate FP that is a NN having a sigmoid function, output gate OP NN having sigmoid, input gate IP NN having a sigmoid, memory state MP and hidden state HP hat is a vector. Assume, the memory from input blocks as $cp_{(tp-1)}$, the input vector as xp_{tp}, and the previous output from the blocks is shown by $mp_{(tp-1)}$. The output of the current block is shown by mp_{tp}, and the memory of the current block is shown by cp_{tp} respectively. Here, the structure of the neural network is described as follows. The input shape of the network iSn dataset 1 is considered to be 5650×22 and similarly, in dataset 2, dataset 3, dataset 4 and dataset 5 is known to be $104{,}345 \times 22$, $225{,}745 \times 78$, $148{,}517 \times 42$ and $494{,}020 \times 41$, respectively. Then, the output shape of the network in all five datasets is determined to be 0 or 1, which means whether the attacks happened or not is detected. The weight parameters of the LSTM network like number of epochs are determined to be 6, mini-batch size is estimated to be 20, learning rate is computed as 0.01, gradient threshold is 1 and the hidden neurons count is counted to be 10. The network involves three inputs. The decision is taken by the single unit that considers the previous output, present input, and the earlier memory that creates a new output and hence the memory is alerted.

It consists of special blocks known as memory blocks in the hidden layers. Every memory block involves the input and the output gate. The control functions are done at the input and the output activation. The forget gate is also included in the final stage. The mapping is finding from the input sequence $xp = (xp_1, xp_2, \ldots, xp_{TP})$ to the output sequence $yp = (yp_1, yp_2, \ldots, yp_{TP})$. The network unit activations are figured as given below.

$$ip_{tp} = \sigma\left(WP_{ipxp}xp_{tp} + WP_{ipmp}mp_{tp-1} + WP_{ipcp}cp_{tp-1} + bp_{ip}\right) \tag{11}$$

$$fp_{tp} = \sigma\left(WP_{fpxp}xp_{tp} + WP_{fpmp}mp_{tp-1} + WP_{fpcp}cp_{tp-1} + bp_{fp}\right) \tag{12}$$

$$cp_{tp} = fp_{tp} \otimes cp_{tp-1} + ip_{tp} \otimes gp\left(WP_{cpxp}xp_{tp} + WP_{cpmp}mp_{tp-1} + bp_{cp}\right) \tag{13}$$

$$op_{tp} = \sigma\left(WP_{opxp}xp_{tp} + WP_{opmp}mp_{tp-1} + WP_{opcp}cp_{tp-1} + bp_{op}\right) \tag{14}$$

$$mp_{tp} = op_{tp} \otimes hpcp_{tp} \tag{15}$$

$$yp_{tp} = \varphi\left(WP_{ypmp}mp_{tp} + bp_{yp}\right) \tag{16}$$

Here, the maximum weight from the input gate to the input is shown by $WP_{ip}xp$, and the weight matrices are shown by WP. The diagonal weights associated with the peep-holes connections are shown by $WP_{ip}cp$, $WP_{fp}cp$, and $WP_{op}cp$. The sigmoid function is shown by σ, the input gate bias vector is shown by bp_{ip}, the forget gate is shown by fp, the input gate is shown by ip, the cell activation vector is shown by cp, and the output gate is given by op respectively. The cell output function is shown by hp, the cell input function is shown by gp, and the network output activation function is shown by φ respectively. The complex structure is learnt by the activation layer. The multilayer LSTM uses

the hyperbolic tangent tanh and sigmoid σ functions. The input value is transformed by these functions among 0–1 and − 1–1. The hyperbolic tanh is employed as the block output activation function and block input activation function, and the sigmoid function is utilized as the gate input activation function. The complex mapping functions are learnt by the non linear activation function.

The traditional LSTM offers several advantages such as applicable for predicting, processing, classifying time series with unknown duration time lags, insensitivity to the gap length, has the ability to learn order dependence in the sequence prediction problems, etc. But, it limits from several shortcomings like overfitting, difficulty in applying the dropout algorithm, needs a large quantity of memory bandwidth for the computation process, etc. Hence, to overcome these shortcomings, the hidden neurons of LSTM are optimized by the proposed CP-GWO with the intention of maximizing the accuracy, and the proposed model is termed as CNN-O-LSTM. This suggested CNN-O-LSTM offers several advantages such as it can be utilized for modelling the univariate time series prediction problems, controls the mixing and flow of inputs in the form of pre trained weights, contains controlling capability, etc. The diagrammatic illustration of the proposed CNN-O-LSTM for the detection of DDoS attacks is given in Fig. 3.

5.2 Objective model for optimal detection

The major objective of the CNN-O-LSTM-based detection is to optimize the hidden neurons of the LSTM with the consideration of maximizing the accuracy. The bounding limit of the hidden neurons lies in between 5 to 255. Hence, the objective is designed as in Eq. (17).

$$OF2 = \underset{\{HN_{LSTM}\}}{\arg\min} \left(\frac{1}{Acuy} \right) \tag{17}$$

Here, the objective function of the detection phase is given by $OF2$, the hidden neurons of LSTM are given by HN_{LSTM}, and the accuracy is given by $Acuy$ respectively. Accuracy is described as, "the discrepancy in the recognized outcome to the ground value" as shown in Eq. (18).

$$Acry = \frac{TE^{PE} + TE^{NE}}{TE^{PE} + TE^{NE} + FE^{PE} + FE^{NE}} \tag{18}$$

In the above equation, the terms TE^{PE}, TE^{NE}, FE^{PE}, and FE^{NE} represent the "true positive, true negative, false positive, and false negative" respectively.

6 Results and discussions

6.1 Experimental setup

The proposed CP-GWO-based DDoS attack detection was implemented in MATLAB 2020a and the results were analyzed. Here, the maximum iteration count, as well as the population size, was taken as 10. The proposed CP-GWO-O-LSTM was compared with several optimization algorithms such as PSO-O-LSTM (Liu et al. 2019), WOA-O-LSTM (Penmatsa et al. 2021), MFO-O-LSTM (Chaithanya et al. 2020), and GWO-O-LSTM (Mirjalili et al. 2014) as well as the classification algorithms like, NN (Li et al. 2010), DNN (Makuvaza et al. 2021), RNN (Chen et al. 2020), DBN (Jing et al. 2020), ARIMA (Tabatabaie Nezhad et al. 2016b), and LSTM (Abbasi et al. 2019) in terms of five datasets with respect to several performance measures such as, "accuracy, sensitivity, specificity, precision, FPR, FNR, FDR, NPV, F1 score, and MCC" to define the superiority of the considered method.

6.2 Performance measures

The several performance measures (Priyadarshini et al. 2020) used here are described below.

(a) Specificity: "It expresses the rate of the wrong data correctly neglected during the data retrieval".

$$Speciy = \frac{TE^{NE}}{FE^{PE} + TE^{NE}} \tag{19}$$

(b) FDR: "It refers to the total false positives within the positive values"

$$FDR = \frac{FE^{PE}}{FE^{PE} + TE^{PE}} \tag{20}$$

(c) MCC: "It is the correlation coefficient between the response parameters".

$$MCC = \frac{TE^{PE} * TE^{NE} - FE^{PE} * FE^{NE}}{\sqrt{\begin{matrix}\left(TE^{PE} + FE^{PE}\right)\left(TE^{PE} + FE^{NE}\right) \\ \left(TE^{NE} + FE^{PE}\right)\left(TE^{NE} + FE^{NE}\right)\end{matrix}}} \tag{21}$$

(d) NPV: "It indicates the direct false values from the retrieval system".

$$NPV = \frac{TE^{NE}}{FE^{NE} + TE^{NE}} \tag{22}$$

(e) Accuracy: It is clearly described in Eq. (18).

(f) FNR: "It refers to the rate of negative response within the positive values".

$$FNR = \frac{FE^{NE}}{FE^{NE} + TE^{PE}} \tag{23}$$

(g) F1 score: "It is the mean value between sensitivity and specificity".

$$F1\ score = \frac{2TE^{PE}}{2TE^{PE} + FE^{PE} + FE^{NE}} \tag{24}$$

(h) Precision: "It is the measure of the deviation in the data retrieval from the original data".

$$\Pr sn = \frac{TE^{PE}}{TE^{PE} + FE^{PE}} \tag{25}$$

(i) FPR: "It refers to the total positive results within the negative output".

$$FPR = \frac{FE^{PE}}{FE^{PE} + TE^{NE}} \tag{26}$$

(j) Sensitivity: "It refers to the correctly identified retrieved data among the various images".

$$Sensy = \frac{TE^{PE}}{TE^{PE} + FE^{NE}} \tag{27}$$

6.3 Convergence analysis

The convergence analysis of the proposed and conventional heuristic-based DDoS attack detection in terms of minimizing the correlation among the features is depicted in Fig. 4. On considering the 6th iteration, the cost function of the CP-GWO is 0.25%, 0.05%, 1.48%, and 0.02% higher than GWO, MFO, WOA, and PSO respectively. Thus, it is clear that the convergence analysis outcomes are better with the proposed CP-GWO than the remaining heuristic-based techniques.

6.4 Algorithmic analysis

The algorithmic analysis of the proposed and the existing heuristic-based techniques in terms of five datasets for different performance measures are shown clearly in Figs. 5, 6, 7, 8, and 9 respectively. From Fig. 5a, for dataset 1, the accuracy of the CP-GWO-O-LSTM with CNN at 65% learning percentage is 0.31%, 0.95%, 1.26%, and 0.63% advanced than GWO-O-LSTM, MFO-O-LSTM, WOA-O-LSTM, and PSO-O-LSTM respectively. Through the sensitivity comparison between the proposed and conventional techniques, the proposed secures very high values at all the learning percentage of 30–90. Hence, it is clear that the proposed CP-GWO-O-LSTM with CNN attains better algorithmic analysis for the DDoS attack detection than the remaining heuristic-based approaches.

6.5 Classification analysis

The classification analysis of the proposed and conventional machine learning-based methods for the DDoS detection in

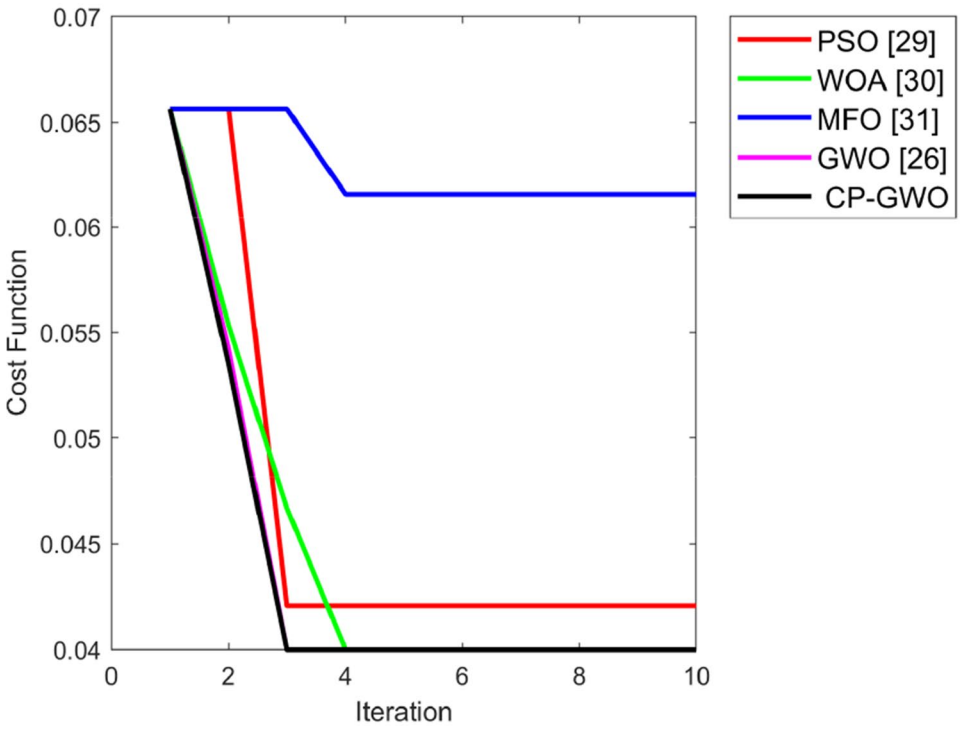

Fig. 4 Convergence analysis related to the correlation minimization of the proposed and conventional heuristic-based DDoS attack detection methods

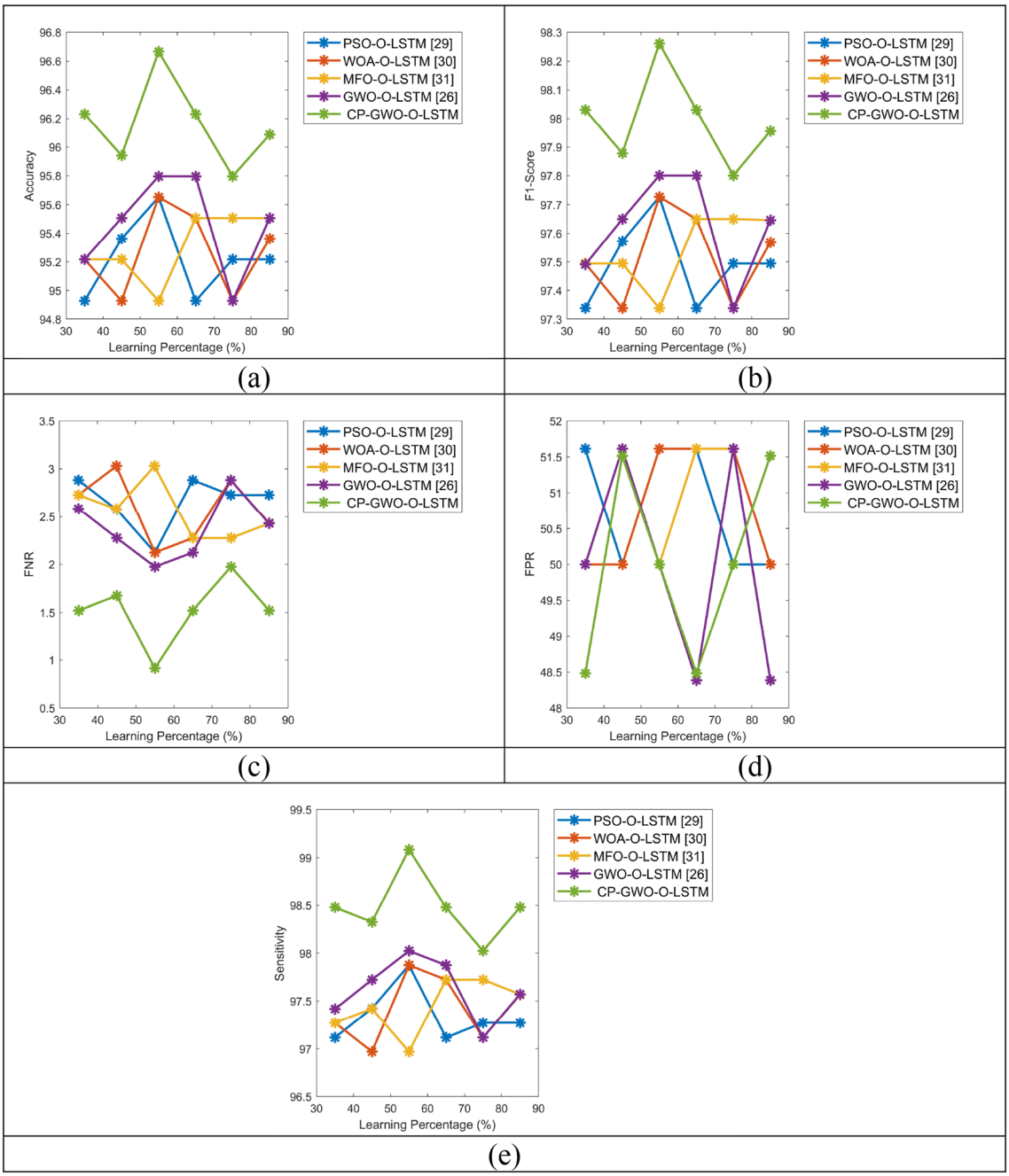

Fig. 5 Algorithmic analysis of the proposed and conventional heuristic-based DDoS attack detection methods for dataset 1 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

terms of five datasets is described in Figs. 10, 11, 12, 13, and 14 respectively. On considering Fig. 11a, for dataset 2, the accuracy of CP-GWO-O-LSTM at 75% learning percentage is 4.30%, 5.43%, 2.11%, 3.19%, 14.12%, and 21.25% higher than LSTM, ARIMA, DBN, RNN, DNN, and NN respectively. While observing the FNR of the dataset 3, the proposed model ensures very less values like 1.5, 1.6, 1.5, 1.7, 1.8 and 1.6 at the learning percentage of 35, 45, 55, 65, 75 and 85. Thus, the classification analysis of the DDoS attack detection reveals better results with the proposed CP-GWO-O-LSTM (CNN-OLSTM) than the other methods.

6.6 Overall performance analysis

The overall performance analysis of the proposed and conventional heuristic-based techniques with CNN in terms of five datasets for the DDoS attack detection is listed in Tables 2, 3, 4, 5, and 6 respectively. In Table 4, for dataset

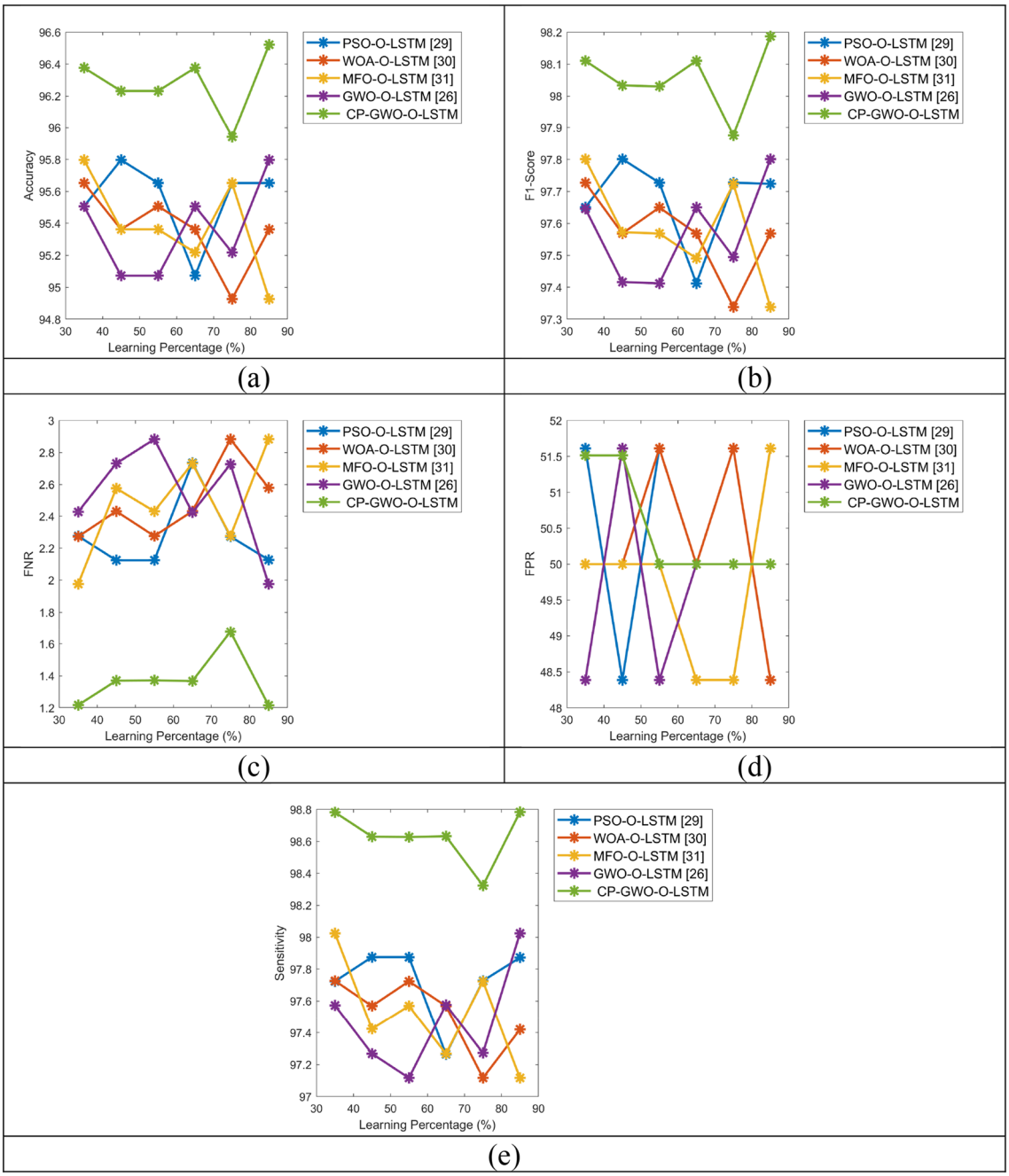

Fig. 6 Algorithmic analysis of the proposed and conventional heuristic-based DDoS attack detection methods for dataset 2 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

3, the accuracy of CP-GWO-O-LSTM is 1.37%, 1.53%, 0.91%, and 1.37% surpassed than GWO-O-LSTM, MFO-O-LSTM, WOA-O-LSTM, and PSO-O-LSTM respectively. Similarly, on considering the other datasets, the proposed model enhances its performance than other heuristic-based models. Therefore, the overall performance analysis is better with the CP-GWO-O-LSTM with CNN than the other methods for the DDoS attack detection.

6.7 Overall classification analysis

The overall classification analysis of the proposed and conventional machine learning-based methods for the DDoS attack detection is portrayed in Tables 7, 8, 9, 10, and 11 respectively. From Table 7, for dataset 1, the accuracy of CP-GWO-O-LSTM is 2.46%, 2.62%, 2.78%, 1.83%, 10.63%, and 17.05% advanced than ARIMA, DBN, RNN, DNN, and

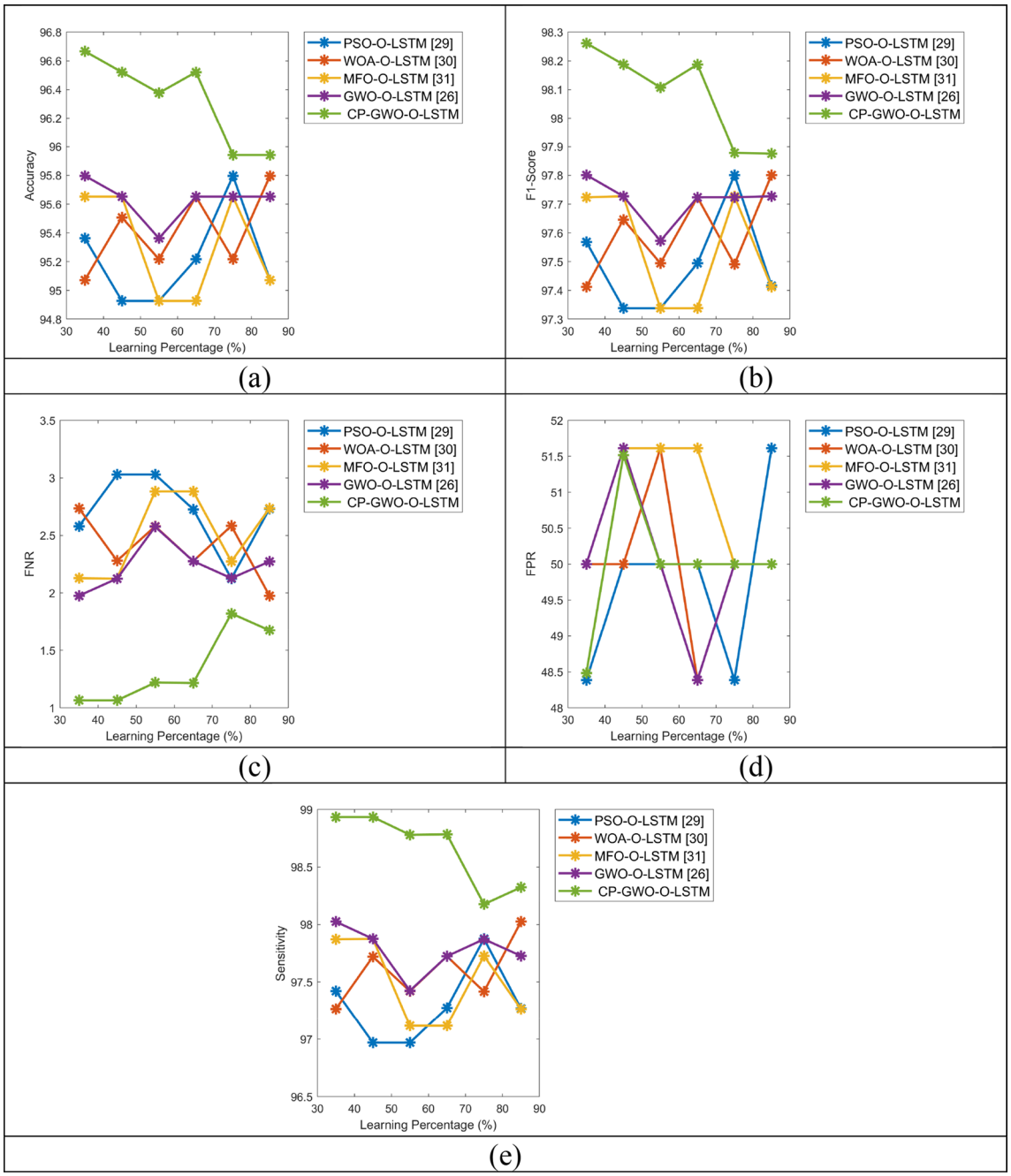

Fig. 7 Algorithmic analysis of the proposed and conventional heuristic-based DDoS attack detection methods for dataset 3 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

NN respectively. From Table 11, for dataset 5, the accuracy of CP-GWO-O-LSTM is 2.31%, 2.31%, 1.84%, 2.15%, 15.45%, and 19.40% progressed than ARIMA, DBN, RNN, DNN, and NN respectively. Thus, it is clear that the overall classification analysis outcomes are better with the CP-GWO-O-LSTM with CNN than the remaining methods for the DDoS attack detection.

7 Conclusion

This paper has implemented the DDoS detection model by involving the CNN and LSTM. Initially, the dataset was gathered from five standard publically available dataset. Next, the optimal features were selected using proposed CP-GWO by considering the objective function of minimizing

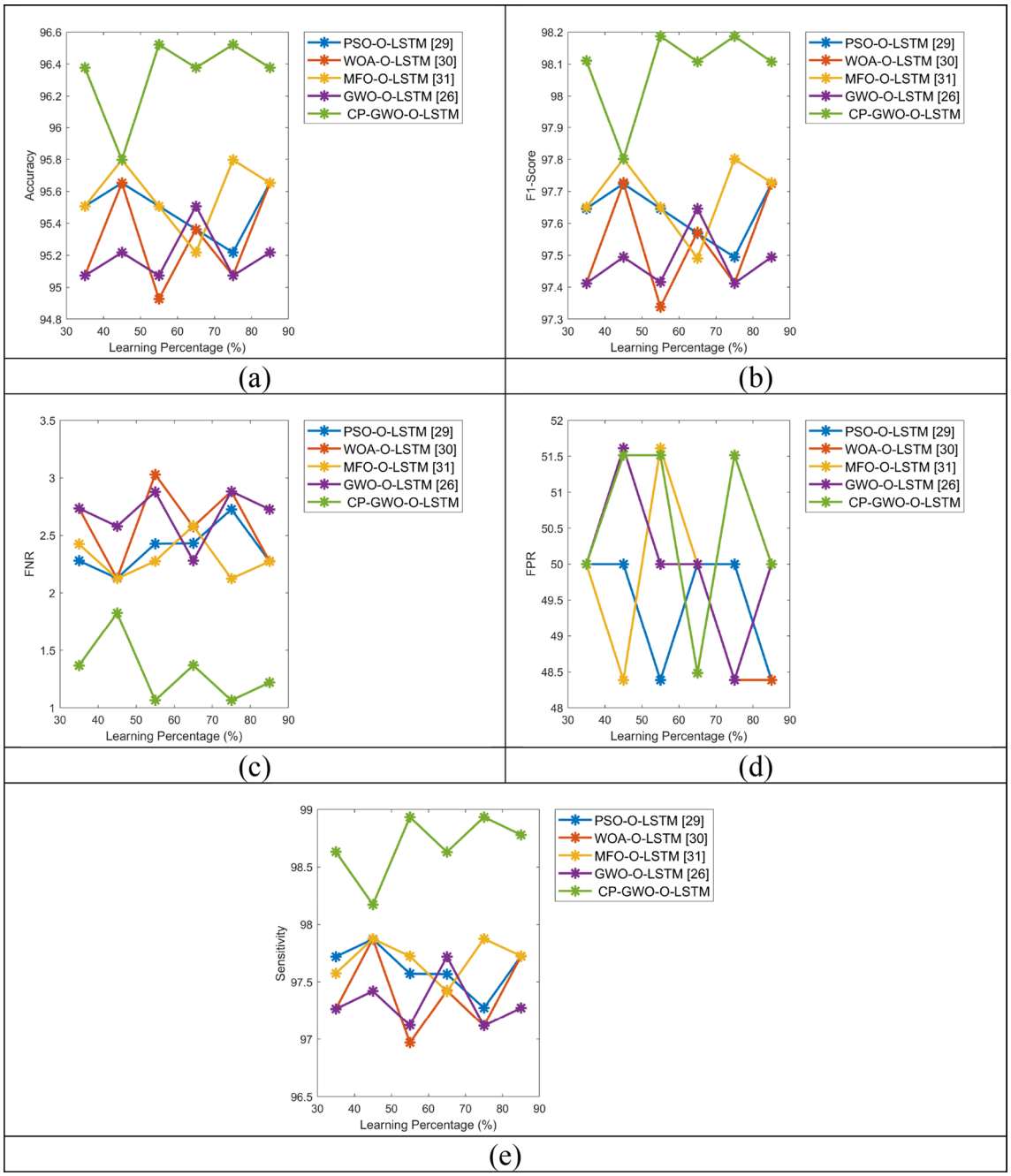

Fig. 8 Algorithmic analysis of the proposed and conventional heuristic-based DDoS attack detection methods for dataset 4 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

the correlation among the features. The CNN was then used for the feature learning process, where the features from the second pooling layer were extracted and given to the final detection phase. In the detection phase, the proposed O-LSTM was employed for achieving accurate detection by optimizing the hidden neurons of the LSTM using the same CP-GWO. This O-LSTM detected whether the DDoS attack was occurred or not. From the analysis, the accuracy of the proposed CP-GWO-based CNN-OLSTM was 2.46%, 2.62%, 2.78%, 1.83%, 10.63%, and 17.05% advanced than LSTM,

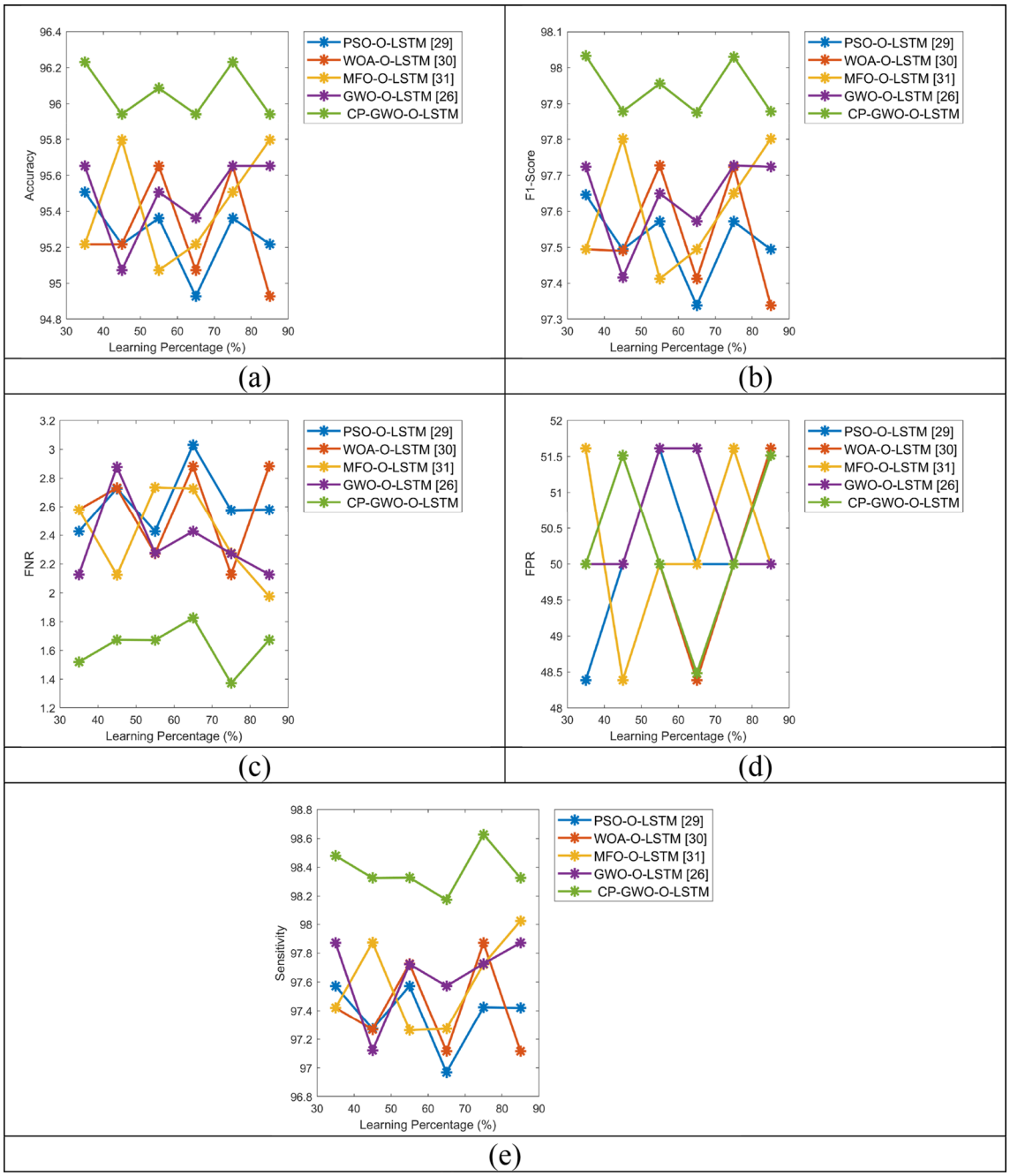

Fig. 9 Algorithmic analysis of the proposed and conventional heuristic-based DDoS attack detection methods for dataset 5 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

ARIMA, DBN, RNN, DNN, and NN respectively. Hence, the proposed CP-GWO-O-LSTM holds better results with the DDoS attack detection than the remaining conventional methods. The proposed model includes the detection of volume-based attacks and application layer-based attacks, whereas in the future work, the proposed model will be extended with detecting the protocol-based attacks and also will analyze the validity of the proposed model for all subclasses of DDoS.

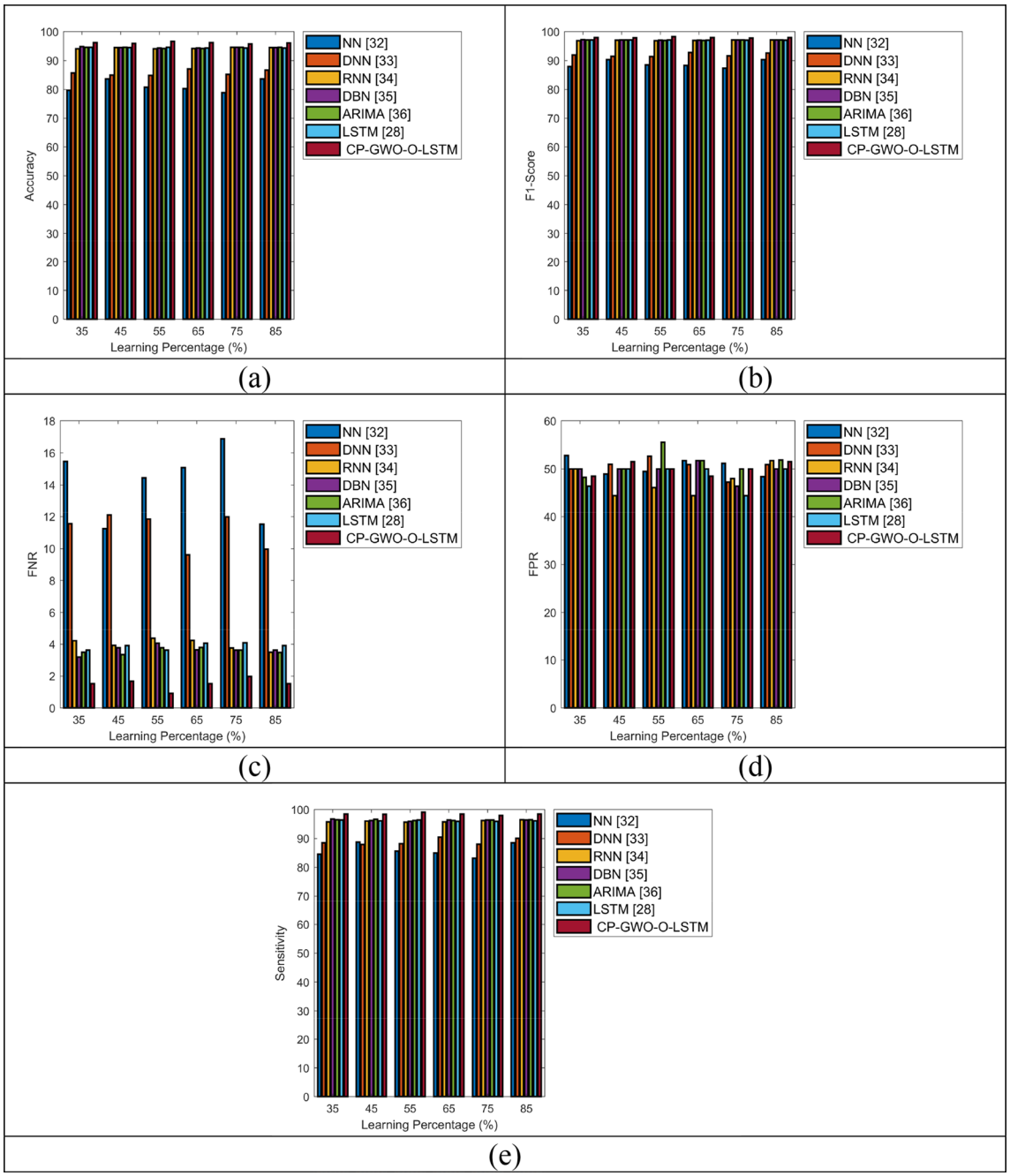

Fig. 10 Classification analysis of the proposed and conventional machine learning-based DDoS attack detection methods for dataset 1 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

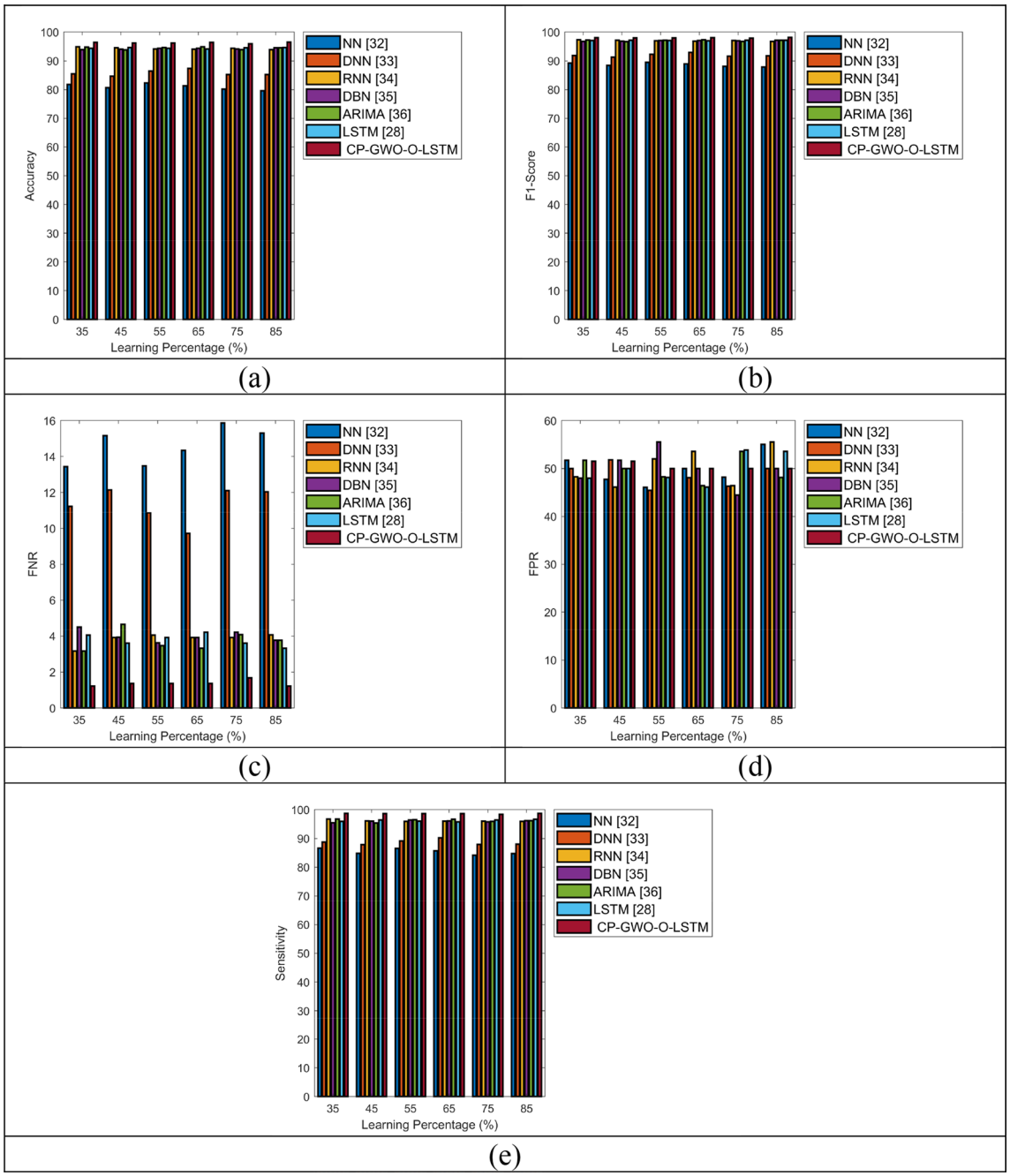

Fig. 11 Classification analysis of the proposed and conventional machine learning-based DDoS attack detection methods for dataset 2 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

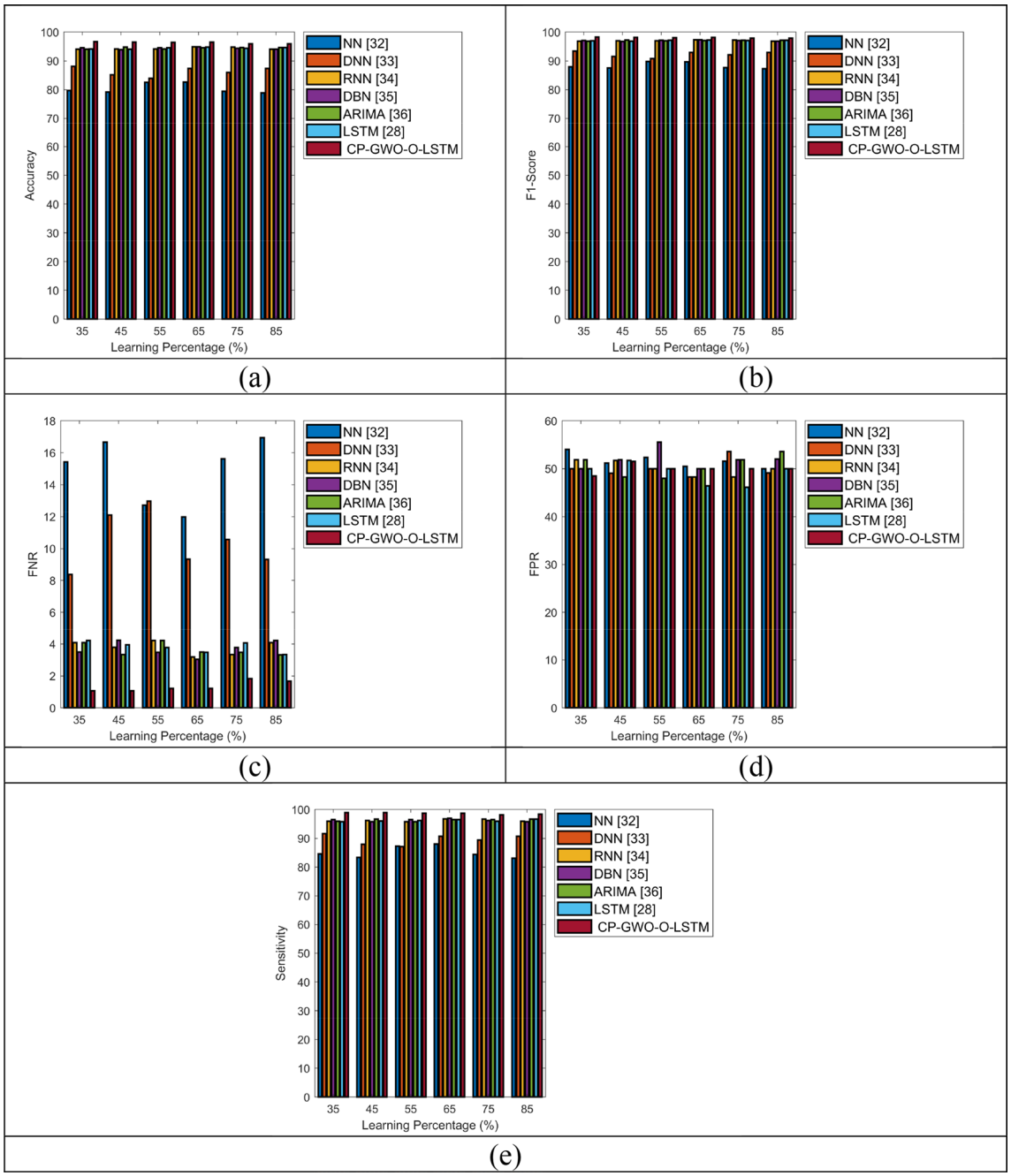

Fig. 12 Classification analysis of the proposed and conventional machine learning-based DDoS attack detection methods for dataset 3 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

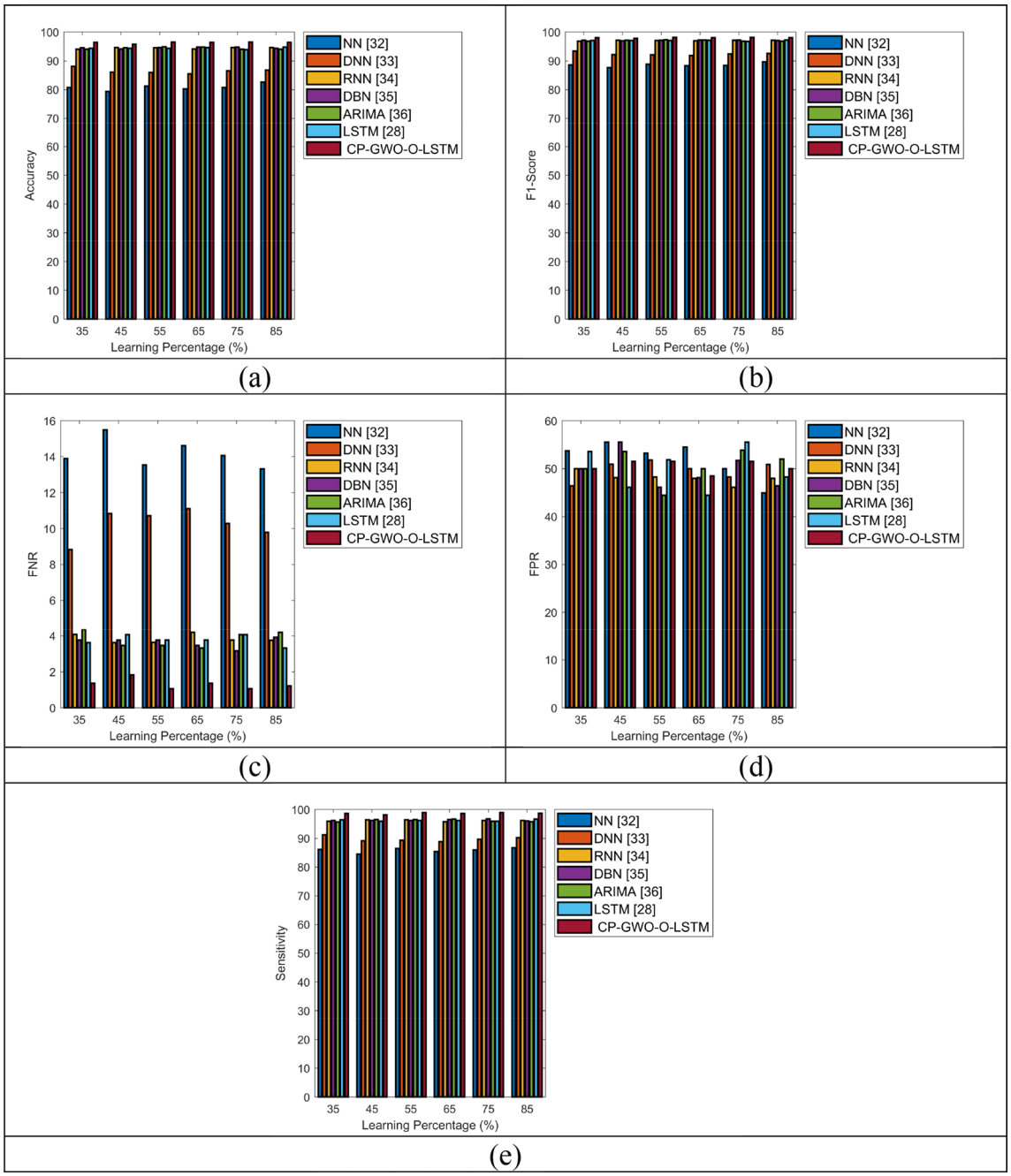

Fig. 13 Classification analysis of the proposed and conventional machine learning-based DDoS attack detection methods for dataset 4 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

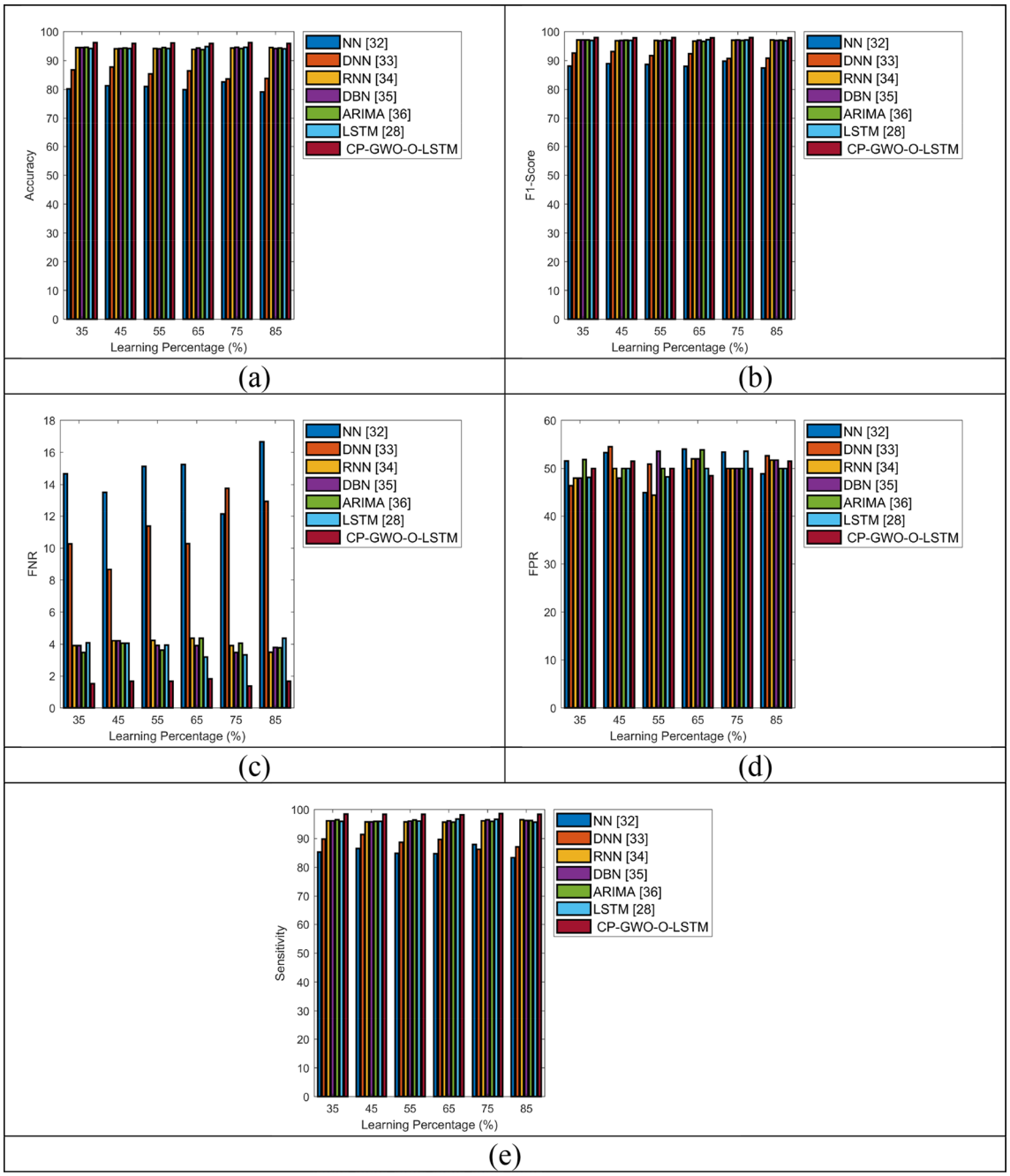

Fig. 14 Classification analysis of the proposed and conventional machine learning-based DDoS attack detection methods for dataset 5 in terms of "**a** accuracy, **b** F1 score, **c** FNR, **d** FPR, and **e** sensitivity"

Table 2 Overall performance analysis of the proposed and conventional heuristic-based DDoS detection methods for dataset 1

Performance measures	PSO-O-LSTM (Liu et al. 2019)	WOA-O-LSTM (Penmatsa et al. 2021)	MFO-O-LSTM (Chaithanya et al. 2020)	GWO-O-LSTM (Mirjalili et al. 2014)	CP-GWO-O-LSTM
Accuracy	0.95507	0.95072	0.95217	0.95072	**0.96522**
Sensitivity	0.9772	0.97121	0.97416	0.97117	**0.98782**
Specificity	0.5	0.5	0.5	0.51613	**0.51515**
Precision	0.97572	0.97713	0.97565	0.9771	**0.97594**
FPR	0.5	0.5	0.5	0.48387	**0.48485**
FNR	0.022796	0.028788	0.025836	0.028832	**0.012177**
NPV	0.5	0.5	0.5	0.51613	**0.51515**
FDR	0.024279	0.022866	0.024353	0.022901	**0.02406**
F1 score	0.97646	0.97416	0.9749	0.97412	**0.98185**
MCC	0.48447	0.44397	0.46728	0.46001	**0.57439**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 3 Overall performance analysis of the proposed and conventional heuristic-based DDoS detection methods for dataset 2

Performance measures	PSO-O-LSTM (Liu et al. 2019)	WOA-O-LSTM (Penmatsa et al. 2021)	MFO-O-LSTM (Chaithanya et al. 2020)	GWO-O-LSTM (Mirjalili et al. 2014)	CP-GWO-O-LSTM
Accuracy	0.95652	0.95652	0.94928	0.95652	**0.95942**
Sensitivity	0.97872	0.97727	0.97117	0.97872	**0.98323**
Specificity	0.5	0.5	0.48387	0.5	**0.5**
Precision	0.97576	0.97727	0.97561	0.97576	**0.97432**
FPR	0.5	0.5	0.51613	0.5	**0.5**
FNR	0.021277	0.022727	0.028832	0.021277	**0.016768**
NPV	0.5	0.5	0.48387	0.5	**0.5**
FDR	0.024242	0.022727	0.02439	0.024242	**0.02568**
F1 score	0.97724	0.97727	0.97338	0.97724	**0.97876**
MCC	0.49367	0.47727	0.43549	0.49367	**0.53008**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 4 Overall performance analysis of the proposed and conventional heuristic-based DDoS detection methods for dataset 3

Performance measures	PSO-O-LSTM (Liu et al. 2019)	WOA-O-LSTM (Penmatsa et al. 2021)	MFO-O-LSTM (Chaithanya et al. 2020)	GWO-O-LSTM (Mirjalili et al. 2014)	CP-GWO-O-LSTM
Accuracy	0.95217	0.95652	0.95072	0.95217	**0.96522**
Sensitivity	0.97273	0.97876	0.97264	0.97269	**0.98782**
Specificity	0.5	0.48387	0.5	0.51613	**0.51515**
Precision	0.97717	0.97579	0.97561	0.97713	**0.97594**
FPR	0.5	0.51613	0.5	0.48387	**0.48485**
FNR	0.027273	0.021244	0.027356	0.027314	**0.012177**
NPV	0.5	0.48387	0.5	0.51613	**0.51515**
FDR	0.022831	0.024206	0.02439	0.022866	**0.02406**
F1 score	0.97494	0.97727	0.97412	0.9749	**0.98185**
MCC	0.45176	0.47759	0.45923	0.46782	**0.57439**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 5 Overall performance analysis of the proposed and conventional heuristic-based DDoS detection methods for dataset 4

Performance measures	PSO-O-LSTM (Liu et al. 2019)	WOA-O-LSTM (Penmatsa et al. 2021)	MFO-O-LSTM (Chaithanya et al. 2020)	GWO-O-LSTM (Mirjalili et al. 2014)	CP-GWO-O-LSTM
Accuracy	0.95362	0.95072	0.95217	0.95072	**0.96377**
Sensitivity	0.9742	0.97121	0.97273	0.97117	**0.9863**
Specificity	0.51613	0.5	0.5	0.51613	**0.51515**
Precision	0.97717	0.97713	0.97717	0.9771	**0.9759**
FPR	0.48387	0.5	0.5	0.48387	**0.48485**
FNR	0.025797	0.028788	0.027273	0.028832	**0.013699**
NPV	0.51613	0.5	0.5	0.51613	**0.51515**
FDR	0.022831	0.022866	0.022831	0.022901	**0.024096**
F1 score	0.97568	0.97416	0.97494	0.97412	**0.98107**
MCC	0.47596	0.44397	0.45176	0.46001	**0.56195**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 6 Overall performance analysis of the proposed and conventional heuristic-based DDoS detection methods for dataset 5

Performance measures	PSO-O-LSTM (Liu et al. 2019)	WOA-O-LSTM (Penmatsa et al. 2021)	MFO-O-LSTM (Chaithanya et al. 2020)	GWO-O-LSTM (Mirjalili et al. 2014)	CP-GWO-O-LSTM
Accuracy	0.95072	0.95072	0.94928	0.95217	**0.96377**
Sensitivity	0.97121	0.97264	0.9697	0.97273	**0.98782**
Specificity	0.5	0.5	0.5	0.5	**0.48485**
Precision	0.97713	0.97561	0.9771	0.97717	**0.97447**
FPR	0.5	0.5	0.5	0.5	**0.51515**
FNR	0.028788	0.027356	0.030303	0.027273	**0.012177**
NPV	0.5	0.5	0.5	0.5	**0.48485**
FDR	0.022866	0.02439	0.022901	0.022831	**0.025526**
F1 score	0.97416	0.97412	0.97338	0.97494	**0.9811**
MCC	0.44397	0.45923	0.43651	0.45176	**0.5505**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 7 Overall classification analysis of the proposed and conventional machine learning-based DDoS detection methods for dataset 1

Performance measures	NN (Li et al. 2010)	DNN (Makuvaza et al. 2021)	RNN (Chen et al. 2020)	DBN (Jing et al. 2020)	ARIMA (Tabatabaie Nezhad et al. 2016b)	LSTM (Abbasi et al. 2019)	CP-GWO-O-LSTM
Accuracy	0.82464	0.87246	0.94783	0.93913	0.94058	0.94203	**0.96522**
Sensitivity	0.87728	0.90823	0.96386	0.95495	0.96067	0.96224	**0.98782**
Specificity	0.45977	0.48276	0.53846	0.5	0.48276	0.46429	**0.51515**
Precision	0.9184	0.95033	0.9816	0.98148	0.97692	0.97699	**0.97594**
FPR	0.54023	0.51724	0.46154	0.5	0.51724	0.53571	**0.48485**
FNR	0.12272	0.091772	0.036145	0.045045	0.039334	0.037764	**0.012177**
NPV	0.45977	0.48276	0.53846	0.5	0.48276	0.46429	**0.51515**
FDR	0.081597	0.049669	0.018405	0.018519	0.023077	0.023006	**0.02406**
F1 score	0.89737	0.9288	0.97264	0.96804	0.96873	0.96956	**0.98185**
MCC	0.30127	0.32845	0.41931	0.34866	0.38074	0.36892	**0.57439**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 8 Overall classification analysis of the proposed and conventional machine learning-based DDoS detection methods for dataset 2

Performance measures	NN (Li et al. 2010)	DNN (Makuvaza et al. 2021)	RNN (Chen et al. 2020)	DBN (Jing et al. 2020)	ARIMA (Tabatabaie Nezhad et al. 2016b)	LSTM (Abbasi et al. 2019)	CP-GWO-O-LSTM
Accuracy	0.80145	0.87246	0.94348	0.94638	0.93913	0.94493	**0.95942**
Sensitivity	0.85378	0.90363	0.96078	0.96526	0.95639	0.96515	**0.98323**
Specificity	0.47368	0.52632	0.51852	0.5	0.48	0.5	**0.5**
Precision	0.91039	0.95492	0.98	0.97856	0.97997	0.97699	**0.97432**
FPR	0.52632	0.47368	0.48148	0.5	0.52	0.5	**0.5**
FNR	0.14622	0.096367	0.039216	0.034743	0.043609	0.034848	**0.016768**
NPV	0.47368	0.52632	0.51852	0.5	0.48	0.5	**0.5**
FDR	0.089606	0.045075	0.02	0.02144	0.020031	0.023006	**0.02568**
F1 score	0.88118	0.92857	0.9703	0.97186	0.96804	0.97104	**0.97876**
MCC	0.28687	0.3498	0.39771	0.40751	0.34494	0.41583	**0.53008**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 9 Overall classification analysis of the proposed and conventional machine learning-based DDoS detection methods for dataset 3

Performance measures	NN (Li et al. 2010)	DNN (Makuvaza et al. 2021)	RNN (Chen et al. 2020)	DBN (Jing et al. 2020)	ARIMA (Tabatabaie Nezhad et al. 2016b)	LSTM (Abbasi et al. 2019)	CP-GWO-O-LSTM
Accuracy	0.78551	0.85797	0.94638	0.94493	0.94058	0.94783	**0.96522**
Sensitivity	0.82266	0.88942	0.96229	0.96386	0.95633	0.96818	**0.98782**
Specificity	0.50617	0.50877	0.55556	0.46154	0.53846	0.5	**0.51515**
Precision	0.92606	0.95262	0.98154	0.97859	0.98145	0.97706	**0.97594**
FPR	0.49383	0.49123	0.44444	0.53846	0.46154	0.5	**0.48485**
FNR	0.17734	0.11058	0.037707	0.036145	0.043675	0.031818	**0.012177**
NPV	0.50617	0.50877	0.55556	0.46154	0.53846	0.5	**0.51515**
FDR	0.073937	0.047377	0.018462	0.021407	0.018547	0.022936	**0.02406**
F1 score	0.8713	0.91993	0.97182	0.97117	0.96873	0.9726	**0.98185**
MCC	0.25724	0.31269	0.42969	0.36427	0.38976	0.42935	**0.57439**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 10 Overall classification analysis of the proposed and conventional machine learning-based DDoS detection methods for dataset 4

Performance measures	NN (Li et al. 2010)	DNN (Makuvaza et al. 2021)	RNN (Chen et al. 2020)	DBN (Jing et al. 2020)	ARIMA (Tabatabaie Nezhad et al. 2016b)	LSTM (Abbasi et al. 2019)	CP-GWO-O-LSTM
Accuracy	0.81014	0.85942	0.94928	0.94348	0.94348	0.94203	**0.96377**
Sensitivity	0.84934	0.88836	0.9697	0.96229	0.96218	0.95796	**0.9863**
Specificity	0.53488	0.51852	0.5	0.48148	0.51724	0.5	**0.51515**
Precision	0.92767	0.95601	0.9771	0.97853	0.97846	0.98154	**0.9759**
FPR	0.46512	0.48148	0.5	0.51852	0.48276	0.5	**0.48485**
FNR	0.15066	0.11164	0.030303	0.037707	0.037821	0.042042	**0.013699**
NPV	0.53488	0.51852	0.5	0.48148	0.51724	0.5	**0.51515**
FDR	0.072333	0.043993	0.022901	0.021472	0.021538	0.018462	**0.024096**
F1 score	0.88678	0.92095	0.97338	0.97034	0.97025	0.9696	**0.98107**
MCC	0.31815	0.31173	0.43651	0.37721	0.41165	0.35907	**0.56195**

The proposed values are highlighted in bold to shows the better performence over the existing works

Table 11 Overall classification analysis of the proposed and conventional machine learning-based DDoS detection methods for dataset 5

Performance measures	NN (Li et al. 2010)	DNN (Makuvaza et al. 2021)	RNN (Chen et al. 2020)	DBN (Jing et al. 2020)	ARIMA (Tabatabaie Nezhad et al. 2016b)	LSTM (Abbasi et al. 2019)	CP-GWO-O-LSTM
Accuracy	0.80725	0.83478	0.94348	0.94638	0.94203	0.94203	**0.96377**
Sensitivity	0.84653	0.86499	0.96364	0.96241	0.96218	0.96078	**0.98782**
Specificity	0.52381	0.4717	0.5	0.52	0.48276	0.48148	**0.48485**
Precision	0.92767	0.95164	0.97696	0.9816	0.97696	0.97849	**0.97447**
FPR	0.47619	0.5283	0.5	0.48	0.51724	0.51852	**0.51515**
FNR	0.15347	0.13501	0.036364	0.037594	0.037821	0.039216	**0.012177**
NPV	0.52381	0.4717	0.5	0.52	0.48276	0.48148	**0.48485**
FDR	0.072333	0.048359	0.023041	0.018405	0.023041	0.021505	**0.025526**
F1 score	0.88525	0.90625	0.97025	0.97191	0.96951	0.96956	**0.9811**
MCC	0.30357	0.24403	0.40944	0.39516	0.38661	0.37136	**0.5505**

The proposed values are highlighted in bold to shows the better performence over the existing works

References

Abbasi, M.U., Rashad, A., Basalamah, A., Tariq, M.: Detection of epilepsy seizures in neo-natal EEG using LSTM architecture. IEEE Access **7**, 179074–179085 (2019)

Ahmad, I., Namal, S., Ylianttila, M., Gurtov, A.: Security in software defined networks: a survey. IEEE Commun. Surv. Tutor. **17**(4), 2317–2346 (2015)

Arun Raj Kumar, P., Selvakumar, S.: Distributed denial of service attack detection using an ensemble of neural classifier. Comput. Commun. **34**(11), 1328–1341 (2011)

Bhuyan, M.H., Kashyap, H.J., Bhattacharyya, D.K., Kalita, J.K.: Detecting distributed denial of service attacks: methods, tools and future directions. Comput. J. **57**(4), 537–556 (2014)

Bojović, P.D., Bašičević, I., Ocovaj, S., Popović, M.: A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method. Comput. Electr. Eng. **73**, 84–96 (2019)

Çakmakçı, S.D., Kemmerich, T., Ahmed, T., Baykal, N.: Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm. J. Netw. Comput. Appl. **168**, 102756 (2020)

Chaithanya, P.S., Gauthama Raman, M.R., Nivethitha, S., Seshan, K.S., Shankar Sriram, V.: An efficient intrusion detection approach using enhanced random forest and moth-flame optimization technique. In: Computational Intelligence in Pattern Recognition, vol. 999, pp. 877–884 (2020)

Chen, Y., Hwang, K., Ku, W.: Collaborative detection of DDoS attacks over multiple network domains. IEEE Trans. Parallel Distrib. Syst. **18**(12), 1649–1662 (2007a)

Chen, Z., Chen, Z., Delis, A.: An inline detection and prevention framework for distributed denial of service attacks. Comput. J. **50**(1), 7–40 (2007b)

Chen, C.-Y., Chen, L.-A., Cai, Y.-Z., Tsai, M.-H.: RNN-based DDoS detection in IoT scenario. In: 2020 International computer symposium (ICS), pp. 448–453 (2020)

Chonka, A., Singh, J., Zhou, W.: Chaos theory based detection against network mimicking DDoS attacks. IEEE Commun. Lett. **13**(9), 717–719 (2009)

Gao, Y., Wu, H., Song, B., Jin, Y., Luo, X., Zeng, X.: A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network. IEEE Access **7**, 154560–154571 (2019)

Haider, S., et al.: A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. IEEE Access **8**, 53972–53983 (2020)

Hamed, H., Al-Shaer, E.: Taxonomy of conflicts in network security policies. Commun. Mag. IEEE **44**(3), 134–141 (2006)

Jiang, Y., Chen, W., Liu, M., Wang, Y., Meijering, E.: 3D neuron microscopy image segmentation via the ray-shooting model and a DC-BLSTM network. IEEE Trans. Med. Imaging **40**(1), 26–37 (2021)

Jing, Z.H.U., Zhongdong, W.U., Longbin, D.I.N.G., Yang, W.A.N.G.: DDoS attack detection based on DBN in SDN environment. Comput. Eng. **46**(4), 157–161 (2020)

Kasim, Ö.: An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. Comput. Netw. **180**, 107390 (2020)

Kushwah, G.S., Ranga, V.: Voting extreme learning machine based distributed denial of service attack detection in cloud computing. J. Inf. Secur. Appl. **53**, 102532 (2020)

Li, J., Liu, Y., Gu, L.: DDoS attack detection based on neural network. In: 2010 2nd international symposium on aware computing, pp. 196–199 (2010)

Liu, Z., He, Y., Wang, W., Zhang, B.: DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN. China Commun. **16**(7), 144–155 (2019)

Makuvaza, A., Jat, D.S., Gamundani, A.M.: Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs). SN Comput. Sci. **2**, 1–10 (2021)

Malipatil, S., Maheshwari, V., Chandra, M.B.: Area optimization of CMOS full adder design using 3T XOR. In: 2020 International conference on wireless communications signal processing and networking (WiSPNET), pp. 192–194 (2020)

Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)

Namatēvs, I.: Deep convolutional neural networks: structure, feature extraction and training. Inf. Technol. Manag. Sci. **20**, 40–47 (2017)

Penmatsa, R.K.V., Subba Raju, K.V., Ruthala, S.: Application of whale optimization algorithm in DDOS attack detection and feature reduction. In: Inventive Computation and Information Technologies, vol. 173, pp. 93–102 (2021)

Ravi, N., Shalinie, S.M.: Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. IEEE Internet Things J. **7**(4), 3559–3570 (2020)
Shin, S., Wang, H., Gu, G.: A first step toward network security virtualization: from concept to prototype. IEEE Trans. Inf. Forensics Secur. **10**(10), 2236–2249 (2015)
Singh, K., Dhindsa, K.S., Nehra, D.: T-CAD: a threshold based collaborative DDoS attack detection in multiple autonomous systems. J. Inf. Secur. Appl. **51**, 102457 (2020)
Tabatabaie Nezhad, S.M., Nazari, M., Gharavol, E.A.: A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks. IEEE Commun. Lett. **20**(4), 700–703 (2016a)
Tabatabaie Nezhad, S.M., Nazari, M., Gharavol, E.A.: A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks. IEEE Commun. Lett. **20**(4), 700–703 (2016b)
Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R.P., Hu, J.: Detection of denial-of-service attacks based on computer vision techniques. IEEE Trans. Comput. **64**(9), 2519–2533 (2015)
Tang, Y., Li, B., Liu, M., Chen, B., Wang, Y., Ouyang, W.: AutoPedestrian: an automatic data augmentation and loss function search scheme for pedestrian detection. IEEE Trans. Image Process. **30**, 8483–8496 (2021)
Tuan, T.A., Long, H.V., Son, L.H., Kumar, R., Priyadarshini, I., Kim Son, N.T.: Performance evaluation of Botnet DDoS attack detection using machine learning. Evol. Intell. **13**, 283–294 (2020)
Velliangiri, S., Pandey, H.M.: Fuzzy-Taylor-elephant herd optimization inspired deep belief network for DDoS attack detection and comparison with state-of-the-arts algorithms. Futur. Gener. Comput. Syst. **110**, 80–90 (2020)
Wang, M., Lu, Y., Qin, J.: A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput. Secur. **88**, 101645 (2020)
Wool, A.: A quantitative study of firewall configuration errors. Computer **37**(6), 62–67 (2004)
Yu, J., Lee, E., Oh, S., Seo, Y., Kim, Y.: A survey on security requirements for WSNs: focusing on the characteristics related to security. IEEE Access **8**, 45304–45324 (2020)
Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutor. **15**(4), 2046–2069 (2013)
Zeng, K., Wang, Y., Mao, J., Liu, J., Peng, W., Chen, N.: A local metric for defocus blur detection based on CNN feature learning. IEEE Trans. Image Process. **28**(5), 2107–2115 (2019)
Zhang, C., Luo, F., Ranzi, G.: An advanced persistent distributed denial-of-service attack model with reverse-path forwarding-based defending strategy. IEEE Access **7**, 185590–185596 (2019)
Zhou, Z., Gaurav, A., Gupta, B.B., Lytras, M.D., Razzak, I.: A fine-grained access control and security approach for intelligent vehicular transport in 6G communication system. In: IEEE transactions on intelligent transportation systems, pp. 1–10 (2021)

V. Raghava Swamy Dora is lecturer in computer science, DR VSK GDC, Visakhapatnam. He has an experience of 17 years in teaching and presently research scholar at GITAM University and his research interest include network security.

V. Naga Lakshmi is professor in Department of Computer Science in GITAM (Deemed to be university). She published more than 30 technical papers in international journals and 21 technical papers in national and international conferences. She received various awards and recognitions for her research work that she carried throughout the career. Five scholars were awarded PhD degree under her guidance and guiding many research scholars.