BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 73(3), 2025, Article number: e153231
DOI: 10.24425/bpasts.2025.153231

ARTIFICIAL AND COMPUTATIONAL INTELLIGENCE

DODGE: Congestion control in MANET via dragonfly optimized deep learning model

Gladson S. [1]* and **Pandiarajan K.** [2]

[1] Department of Electronics and Communication Engineering, Erode Sengunthar Engineering College, Thudupathi, Erode, Tamil Nadu, India
[2] Department of Electronics and Communication Engineering, Mount Zion College of Engineering and Technology, Pudukkottai, Tamil Nadu, India

Abstract. A mobile ad hoc network (MANET) is a collection of mobile devices attached without infrastructure or central management. Network size increases rapidly, resulting in congestion, network delay, data packet loss, and a drop in throughput, resulting in poor energy efficiency. Data should be mitigated based on the prediction of congestion. To resolve the problem of congestion, a novel dragonfly optimized deep learning for congestion elimination (DODGE) technique was proposed, which predicts the congested node effectively. Initially, the Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) packets from the MANET environment were pre-processed and the features were selected using dragonfly optimization (DFO). The features that are selected from the DFO model were provided to the stacked convolutional neural network combined with bidirectional long short-term memory (SCNN-BiLSTM). The deep learning network will predict the congested node and if congestion is found, then the message will be displayed. The DODGE is simulated by using Network simulator2 (NS2) and a comparison is made between proposed DODGE and traditional approaches such as hybrid gravitational fuzzy neural network (HGFNN), quality of service-aware distributed congestion control (QoS-ADCC), and improved priority aware ad hoc on-demand distance vector (IPA-AODV) in terms of packet delivery ratio (PDR), delay (DE), throughput (TP), energy consumption (EC), latency (L), detection rate (DR), and network lifetime (NL). The proposed SCNN-BiLSTM improves the overall accuracy better than 10.05%, 6.59%, and 3.26% bidirectional long short-term memory (BiLSTM), deep neural network (DNN), convolutional neural network (CNN) for predicting the congested node in the shortest time.

Keywords: MANET; congestion node; deep learning; dragonfly optimization; bidirectional long short-term memory.

1. INTRODUCTION

Mobile ad hoc networks (MANETs) are networks of mobile nodes that communicate wirelessly and collaborate without the need for a centralized infrastructure. MANET nodes can function as transmitters, receivers, or routers. They are self-configuring, multifunctional, and extremely dynamic [1, 2]. If two hosts want to exchange data in such an environment, the intermediary nodes should be able to communicate between them to send and receive the data, with the ability to utilize it anytime and anywhere, as illustrated in Fig. 1 [3]. Nodes can travel at random in any direction and at varied speeds. Because of node mobility, topological changes occur often in MANETs [4, 5]. MANET has proven to be an excellent substitute for a wide range of applications, such as the Internet of Things (IoT), railroad, military, environmental monitoring, and unmanned aerial vehicles (UAVs) [6, 7]. As indicated in Fig. 1, they should be able to communicate with each other and transmit and receive data at any time and from any location [8].

Congestion control is one of the fundamental jobs that an ad hoc network must complete, with the main purpose of reducing the delay and buffer overflow caused by network congestion and offering enhanced network performance [9]. Network congestion arises when a deployment of network resources is not able to handle the level of traffic [10, 11]. Congestion, such as traffic and data loss, can be minimized by compressing the processing and capacity of intermediary nodes, reducing the number of steps required to get resources.

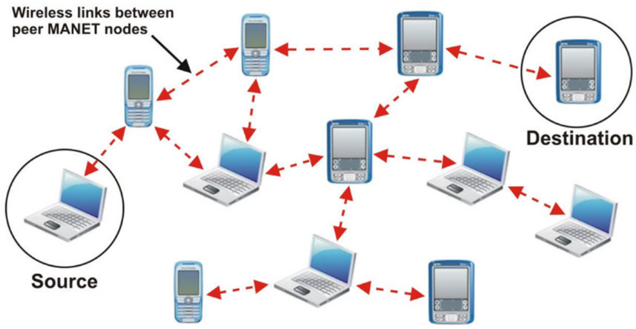

Fig. 1. Mobile ad hoc networks

Various issues in data transmission employing congestion are packet loss estimation, estimating bandwidth availability, and mobility management [12, 13].

MANETs suffer from much higher network congestion than infrastructure networks, and MANETs have limited resources, especially network bandwidth and energy supplies, at each mobile node [14]. By retransmitting network packets that could not be transmitted due to network congestion not only MANET traffic throughput is affected but also energy is wasted [15]. Energy

*e-mail: gladson86@gmail.com

Manuscript submitted 2024-06-15, revised 2024-10-16, initially accepted for publication 2024-10-27, published in May 2025.

is wasted as a result of the high frequency of lost packet retransmissions caused by the high degree of congestion. MANETs face considerable congestion and power control challenges at all levels [16, 17]. To address these issues, the suggested cross-layered method involves many tiers of protocol stack communicating with one another. Congestion has downsides such as extended delays, costly overhead, and a greater rate of packet loss. The main contribution is as follows:

- Initially, pre-processed TCP and UDP packets from the MANET environment and the features are selected using dragonfly optimization (DFO).
- The features that are selected from the DFO model are provided to the stacked CNN combined with Bi-LSTM (SCNN-BiLSTM). It predicts the congested node for reliable transmission and packet delivery with high stability, energy level, network lifetime, and low energy consumption, and then the message is displayed.
- The effectiveness of the DODGE approach is assessed using evaluation criteria such as PDR, DE, TP, EC, L, DR, and NL.

The remaining portion is organized as follows. The literature survey is covered in Section 2, and the DODGE approach is defined in Section 3. The results and conclusion are presented in Sections 4 and 5, respectively. The list of acronyms for the proposed DODGE approach is included in Table 1.

Table 1
List of acronyms

No	Acronyms	Meaning
1.	MANET	Mobile ad hoc network
2.	DODGE	Dragonfly optimized deep learning for congestion elimination
3.	TCP	Transmission control protocol
4.	UDP	User datagram protocol
5.	DFO	Dragonfly optimization
6.	SCNN	Stacked convolutional neural network
7.	BiLSTM	Bidirectional long short-term memory
8.	DL	Deep learning
9.	NS2	Network simulator 2
10.	HGFNN	Hybrid gravitational fuzzy neural network
11.	QoS-ADCC	Quality of service-aware distributed congestion control
12.	IPA-AODV	Improved priority-aware ad hoc on-demand distance vector
13.	PDR	Packet delivery ratio
14.	DE	Delay
15.	TP	Throughput
16.	EC	Energy consumption
17.	L	Latency
18.	DR	Detection rate
19.	NL	Network lifetime
20.	CNN	Convolutional neural network

Table 1 [cont.]

No	Acronyms	Meaning
21.	DNN	Deep neural network
22.	ACO	Ant colony optimization
23.	CH	Cluster head
24.	AIACOAR	Artificial intelligence ant colony optimization aware routing
25.	PA-AODV	Priority-aware ad hoc on-demand distance vector
26.	ACOLBR	Ant colony optimization load balancing routing
27.	OMNET++	Objective modular network testbed in C++
28.	ACEAMR	Ant colony efficient adaptive multipath routing
29.	C-EWA	Clustering-energy weighted algorithm

2. LITERATURE SURVEY

Several studies utilized techniques to predict the congested node in MANET. The following section covers a few of the current evaluation approaches along with their disadvantages.

In 2020, Krishnamoorthy *et al.* [18] suggested a link matrix method for MANETs that reduces crowded lines and boosts system capacity by maximizing the use of each transmission node range before distortion. The efficiency of the traffic matrix approach is assessed using the coefficient of congestion optimization (COCO). Findings for the COCO technique in terms of EC is 60%, TP is 41%, L is 28%, PDR is 5% and overhead is 48%.

In 2022, Saraswathi *et al.* [19] introduced a hybrid gravitational fuzzy neural network (HGFNN) used to identify cross-layer congestion and execute an energy-efficient routing mechanism. It minimizes energy consumption, and transmission delays, and improves packet delivery ratios, hence improving throughput.

In 2021, Kanthimathi and JhansiRani [20] designed an ideal routing-centered CC scheme using the modified ad hoc on-demand distances vector (MAODV) in MANET. The stochastic gradient descent deep learning neural network (SGD-DLNN) determines the congestion status (CS) of each node along the selected paths. The Levy flight-based butterfly optimization (LF-BWO) method selects the most efficient yet least crowded routing circuits.

In 2024, Muthulakshmi *et al.* [21] developed a quality of service-aware distributed congestion control (QoS-ADCC) method that combines passive and preventative aspects to construct and sustain data packet routing. As discussed above, the QoS-ADCC approach achieves outstanding results by delivering packets at a rate of 1.2% and reducing routing overhead by one with a throughput of 31 000 Mbps.

In 2020, Rajendran and Naganathan [22] developed an efficient hybrid clustering algorithm using the ant colony optimization (ACO) technique. The comparative findings demonstrate that the suggested method has excellent network stability, NT, cluster formation, EC, PDR, and TP. This strategy leads to wasting valuable resources and produces overheads in the nodes.

In 2023, Mohan and Vimala [23] developed a rate-aware neuro-fuzzy-based congestion control approach for detecting congestion using baseline parameters. After congestion is controlled, the ideal routes for packets are proposed using an artificial intelligence ant colony optimization aware routing (AIA-COAR) algorithm. The suggested approach has a PDR value of around 99%, creates a very short delay, and requires greater energy consumption.

In 2022, Nallayam Perumal and Selvi [24] developed the improved priority aware ad hoc on-demand distance vector, or IPA-AODV, protocol to improve the quality of service (QoS) of the MANET. The IPA-AODV performs better in QoS metrics than the current protocols, AODV and PA-AODV, suggesting a high mobility environment.

In 2022, Dholey and Sinha [25] developed an ant colony optimization load balancing routing (ACOLBR) approach for managing congestion and balancing load over many channels. ACOLBR efficiently performs load balancing along the path for data transfer from source to destination, according to the results obtained with the objective modular network testbed in C++ (OMNET++).

In 2023, Arun and Jayanthi [26] suggested an ant colony efficient adaptive multipath routing (ACEAMR) strategy for managing congestion and balancing load over several channels. Several network parameters like bandwidth, energy consumption (EC), mobility, etc., are taken into consideration when picking red/blue ants for packet transfers. OMNET++ results reveal that ACOLBR effectively balances the load for data transfer from source to destination when using the route for data transfer.

In 2020, Devika and Sudha [27] designed the clustering-energy weighted algorithm (C-EWA), which efficiently clusters data and modifies power and energy parameters via topology management. Once the optimal cluster head (CH) is found, the Gabriel graph is designed to reduce the transmission power of the nodes. With their respective values of 21.960 J, 0.729, 0.713, 0.295, and 5.256, the proposed approach is highly efficient for battery power, mobility, TP, L, and connection.

The literature review shows that the existing method does not concentrate on congestion control. Since the channel is shared by several nodes, unfortunately, its status or condition is not taken into account during transmission. Lack of availability causes higher latency and lower network performance. Thus, the DODGE method was introduced to reduce congestion, high network performance, and data security.

3. DODGE METHODOLOGY

In this section, a novel dragonfly optimized deep learning for congestion elimination (DODGE) technique is proposed, which effectively predicts the congested node. Initially, the TCP and UDP packets from the MANET environment are pre-processed and the features are selected using dragonfly optimization (DFO) [28, 29]. The features that are selected from the DFO model are provided to the stacked convolutional neural network (SCNN) combined with Bi-LSTM named SCNN-BiLSTM. The deep learning network will predict the congested node and if congestion is found, then the message will be displayed. The system architecture is shown in Fig. 2.

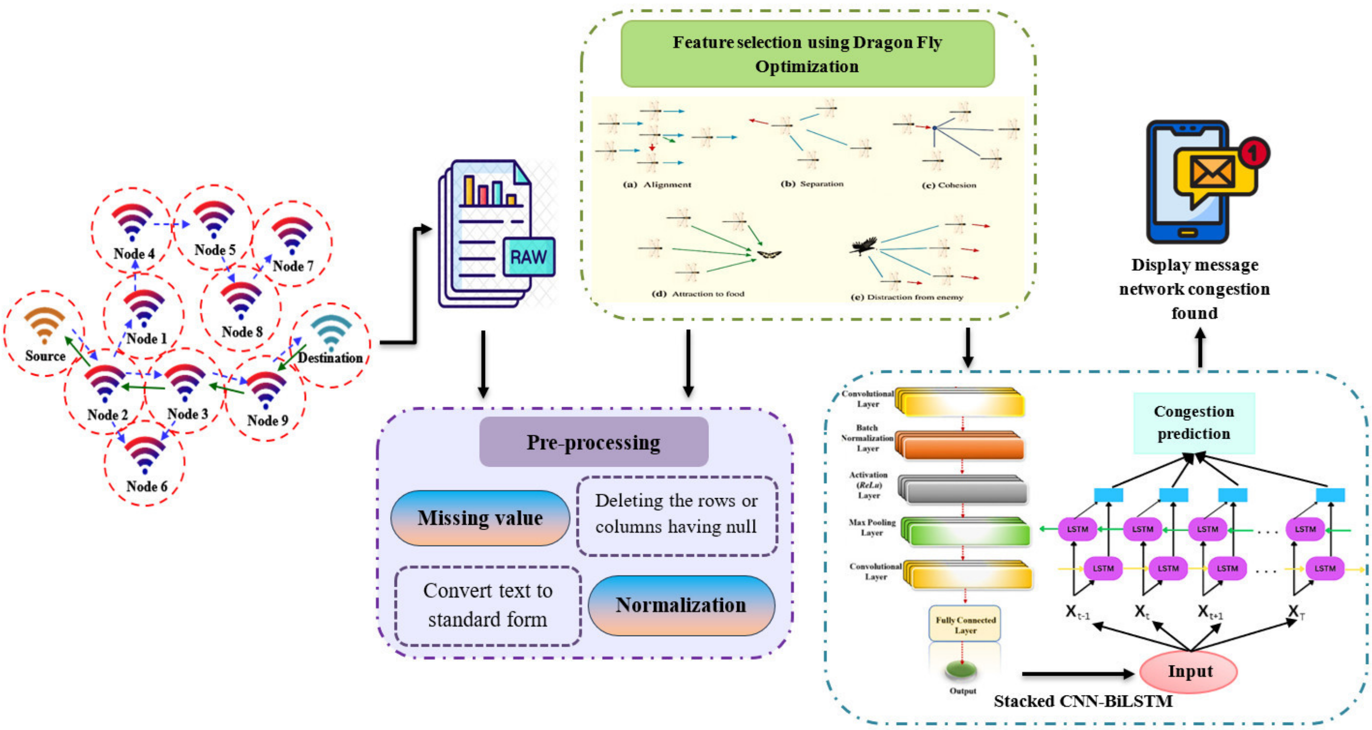

Fig. 2. The overall workflow of DODGE methodology

3.1. Pre-processing

To improve the performance and accuracy, pre-processing is a crucial step that must be taken to use the gathered raw TCP and UDP packets for the analysis process. Pre-processing techniques such as normalization, and handling missing values are used to convert the raw node data to standard form and delete the rows or columns with null from the data.

3.1.1. Handling missing value

Handling missing values is critical to ensuring that machine learning models can use all available data to provide accurate predictions. To handle missing values, remove the rows or columns that contain null values. If a column contains more than 50% null records, it can be eliminated.

3.1.2. Normalization

Normalization requires the simultaneous completion of multiple tasks. All text must be converted to uppercase or lowercase, punctuation must be removed, and numerals must be changed to words. As a result, every text will undergo more consistent pre-processing.

3.2. Feature selection via dragonfly optimization (DFO)

In dragonfly optimization (DFO) based intrusion detection (ID), the optimization process aims to identify the most relevant subset of network information. The ID system is to be optimized by selecting the subset of features that are most discriminative and informative for distinguishing between normal and pathological network data. Figure 3 demonstrates dragonfly behaviour in both static and dynamic swarms. Five weights determine the moving direction of an artificial dragonfly:

A. Separation

The method separates the search agents in the neighborhood. Equation (1) mathematically models the separation behaviour:

$$S_i = -\sum_{j=1}^{N} (X_i - X_j), \tag{1}$$

where N is the number of neighborhoods, X_i is the individual's location, S_i is the individual's distance from themselves, and X_j is the neighboring agent's position.

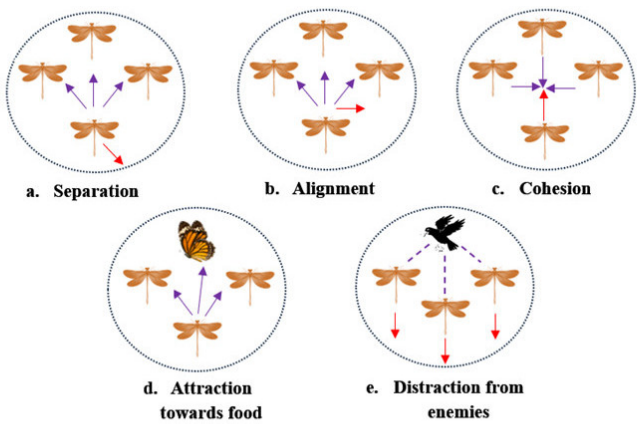

Fig. 3. Dragonfly characteristics: desire for food, diversion from challenger

B. Alignment

It is the matching of the person's velocity with that of the neighboring individual. It is the agent's velocity setting concerning the velocity vectors of the neighboring dragonflies. It is computed as in (2):

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N}, \tag{2}$$

where V_j represents the surrounding individual's velocity, and A_i represents the individual's alignment.

C. Cohesion

It is the distance an individual travels to the center of their neighborhood. It is denoted as (3):

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X. \tag{3}$$

The i-th individual's position is denoted by X, the j-th neighbor by X_j, and the total number of nearby individuals in the swarm by N. Furthermore, C_i demonstrates the consistency of the ith individual.

D. Attraction towards food

Equation (4) represents the dragonfly's journey towards the lure of food:

$$F_i = X^+ - X. \tag{4}$$

The current position of the i-th individual is represented by X, and its attraction to the food is indicated by F_i, and the location of the food supply is indicated by X^+.

E. Distraction from enemies

The dragonflies are staying away from enemies, as shown in (5):

$$E_i = X^- - X, \tag{5}$$

where the opponent's position is shown by X^- and its separate adversary distraction motion is indicated by E_i. Inside a search zone, the step vector (ΔX) and position vector (X), which imitate the movements of dragonflies, are updated. The step vector, which has the following definition, represents the direction in which dragonflies move:

$$\Delta X_i^{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i)\,\omega \Delta X_i^t. \tag{6}$$

In this instance, c stands for weight cohesion, A for alignment weight, and S is for separation weight. The food factor, the opponent factor, the weight of inertia, and the number of repetitions are represented by the variables f, F_i, e, ω, and t. The DFO algorithm selects the features including PDR, DE, TP, EC, L, DR, and NL that are crucial for network performance and congestion prediction. The best feature is obtained by selecting the proper inertia weight with the lowest number of repetitions.

3.3. Congestion node prediction using stacked CNN-BiLSTM

In this phase, a subset of characteristics is learned and categorized using stacked CNN-BiLSTM to forecast the congested node. The congestion node is forecasted based on features such as PDR, DE, TP, EC, L, DR, and NL which are selected by the DFO algorithm. The stacked CNN-BiLSTM classifies the given input features into time-series data and raw network data based on their characteristics.

3.3.1. Stacked CNN

Convolutional layers in this standard CNN design are continuously layered between ReLus, going through the pooling layer before passing between one or more fully linked ReLus. The five main layers of the CNN are input, pooling, convolutional, fully connected, and output. The ReLU function is often used after it, which causes the network to become non-linear. The application of 3×3 filters is controlled by the four convolutional layers. The filters are modified automatically to activate the most pertinent features. Rectified linear units (ReLU) layers use an activation function. The function of ReLU is illustrated in equation (7):

$$f(y) = \max(0, y). \tag{7}$$

After the convolution layers, there are pooling layers. Use the $\max(0, y)$ function in the pooling layers with a 2×2 window to obtain the highest values for each region.

3.3.2. Bi-directional LSTM

The forward and backward LSTM are combined into BiLSTM. To accomplish it, three structures are used: input gate, forget gate, and output gate. The operational process can be expressed as follows:

$$f_t = \text{sigmoid}\left(w_t h_{t-1} + u_t x_t + b_t\right), \tag{8}$$

$$i_t = \text{sigmoid}\left(w_i h_{t-1} + u_i x_t + b_i\right), \tag{9}$$

$$g_t = \tanh\left(w_g h_{t-1} + u_g x_t + b_g\right), \tag{10}$$

$$c_t = f_t c_{t-1} + i_t g_t, \tag{11}$$

$$p_t = \text{sigmoid}\left(w_o h_{t-1} + u_o x_t + b_o\right), \tag{12}$$

$$h_t = p_t \times \tanh\left(c_t\right). \tag{13}$$

The input and output of the current feature vector are denoted by w_f, w_i, w_g, and w_o, respectively, and u_f, u_i, u_g, and u_o, depending on the weight of each control gate.

Bias terms including b_t, b_i, b_g, and b_o are transmitted via the control gate. The amount of data lost after the forget gate is calculated using equation (8). Next, utilizing the input gate state update rate i_t, forgot gate f_t, and state update vector g_t in steps LSTM unit through (11), the update value c_t of c_{t-1} is computed. It is decided by equations (12) and (13), whose portion of the unit state is sent via the output gate. The structural diagram of SCNN-BiLSTM is depicted in Fig. 4 and the hyperparameter settings for the architecture are illustrated in Table 2.

Table 2
Hyperparameter setting in SCNN-BiLSTM

Parameter	Value
Batch size	64
Learning rate	0.001
Optimizer	Adam
Filter size	3x3
Number of filters (Conv layer 1)	64
Number of filters (Conv layer 2)	128
Dropout rate	0.4
LSTM units	128
Activation function	ReLU
Dense layer neurons	256
Epochs	50
Optimizer	Adam

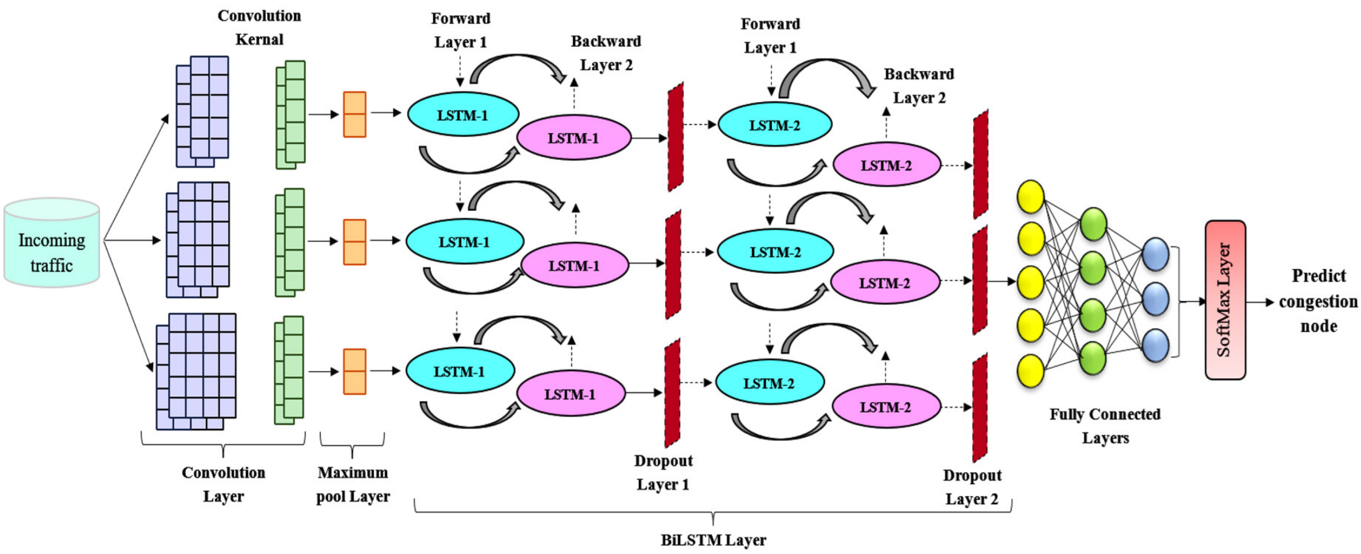

Fig. 4. Structural diagram of stacked CNN-BiLSTM

x_t represents deep features retrieved from the occluded images to create a feature vector; w_i $(i = 1, \ldots, 6)$ represents the weight of a stack of units stacked on top of one another through the VGG layer; h' and h are the LSTM units of the input feature sequences; o_t indicates the output, which follows the feature vector. The operational procedure is written as follows:

$$h_t = \text{sigmoid}\left(w_1 x_t + w_2 h_{t-1} + b_t^{(1)}\right), \tag{14}$$

$$h'_t = \text{sigmoid}\left(w_3 x_t + w_5 h'_{t+1} + b_t^{(2)}\right), \tag{15}$$

$$o'_t = \tanh\left(w_4 h_t + b_t^{(3)}\right), \tag{16}$$

$$o''_t = \tanh\left(w_6 h_t + b_t^{(4)}\right), \tag{17}$$

$$o_t = \frac{(o'_t + o''_t)}{2}. \tag{18}$$

The values of $b_t^{(1)}$, $b_t^{(2)}$, $b_t^{(3)}$, and $b_t^{(4)}$ reflect the biases in the Bi-LSTM at time t, while o'_t and o''_t are the outputs of the two LSTM units handling the feature vectors. In equation (18), the output feature vector is equal to the average of the two vectors at the relevant instant. Finally, the stacked CNN-BiLSTM provides the congestion status based on the processed features and if congestion is detected, an alert or message is generated. This output helps in identifying congested nodes in real-time and improves the network efficiency which secures the data from the attacks in the network.

4. RESULT AND DISCUSSION

This section presents an in-depth analysis and description of the DODGE method outcomes. The efficacy and efficiency of this effort are compared to those of other currently employed techniques. The DODGE data collection method was created in Network Simulator2. After simulation using the produced trace files, the network performance under various assault situations is assessed and compared. To facilitate experimentation and effective performance research, the number of nodes during simulations is increased, and the experimental findings are arranged in the next section. The simulation result for the congestion detection is shown in Fig. 5.

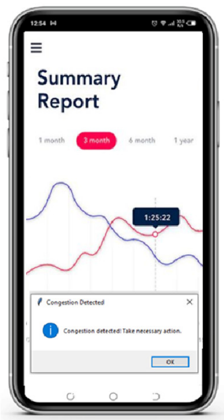

Fig. 5. Congestion detection

4.1. Dataset description

The NS-3 dataset is generated through simulations using the NS-3 network simulator, often used in MANET and wireless network research. It provides a rich set of data capturing various network parameters, such as packet loss, delay, throughput, and congestion metrics. The total number of data samples in the NS-3 dataset depends on the simulation setup, which includes node count, traffic scenarios, and simulation time. Typically, the total dataset has around 100 000 records, the training set would contain approximately 80 000 records. For machine learning tasks, this dataset is often split into training and testing sets, with common ratios like 80% for training and 20% for testing. The dataset contains features useful for predicting network congestion, including timestamps, node ID, and traffic metrics like queue lengths and packet drops.

The accuracy curve for both vectors and epochs is displayed in Fig. 6. As epochs improve, the model loss reduces, as shown by the epoch versus loss curve in Fig. 7.

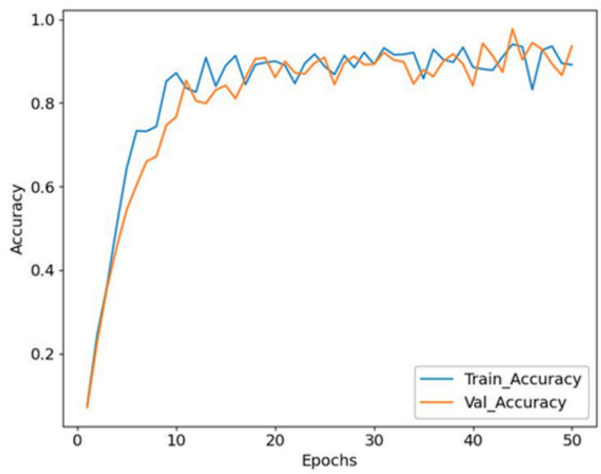

Fig. 6. Accuracy curve of the DODGE model

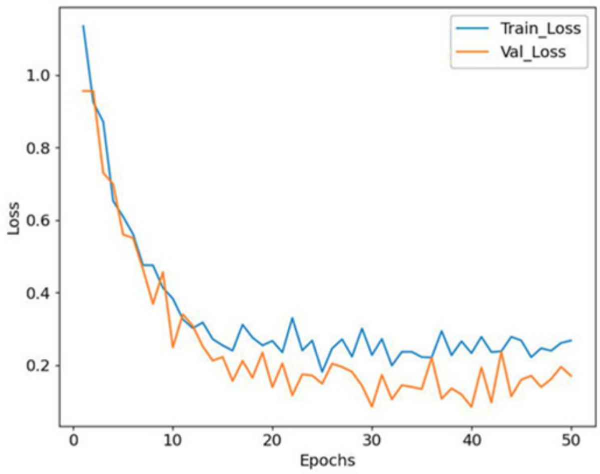

Fig. 7. Loss curve of the DODGE model

The effectiveness of the DODGE approach is measured by the following performance criteria.

4.2. Performance analysis

The proposed DODGE technology was compared with state-of-art technologies such as HGFNN [19], QoS-ADCC [21], and IPA-AODV [24] in terms of PDR, DE, TP, EC, L, DR, and NL.

$$\text{Packet delivery ratio} = \frac{\text{total delivered packets}}{\text{total sent packets}}, \tag{19}$$

$$\text{Delay} = \sum_{i=1}^{n} \frac{(\text{Dest time}\,(i) - \text{Src time}\,(i))}{n}, \tag{20}$$

$$\text{Throughput} = \frac{\text{No. of successfully received packets}}{\text{Stop time} - \text{Start time}}, \tag{21}$$

$$E_c = \sum_{x=1}^{N} E_{x,P}, \tag{22}$$

where $E_{x,P}$ represents Network X total energy usage after P rounds of data collection, and N denotes the number of networks. The PDR represents the number of packets successfully delivered to the target computer. In Fig. 8, the PDR overall performance ratings are displayed.

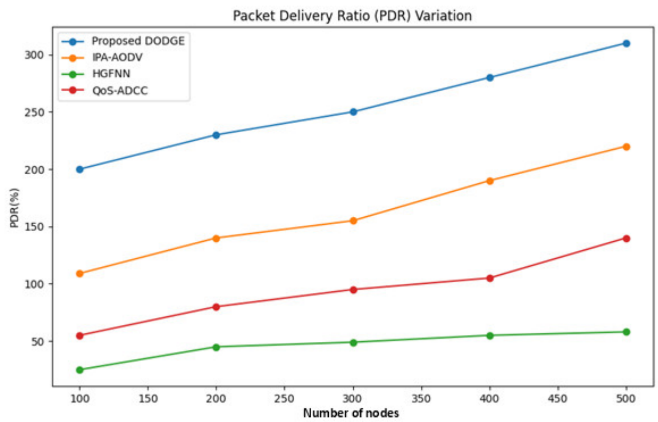

Fig. 8. Comparison of PDR

Figure 9 categorically shows how the proposed strategy finds the intermediate node with the smallest latency and keeps trying to transmit the data packet there to achieve the greatest performance. The proposed technique has a lower end-to-end latency than the present method.

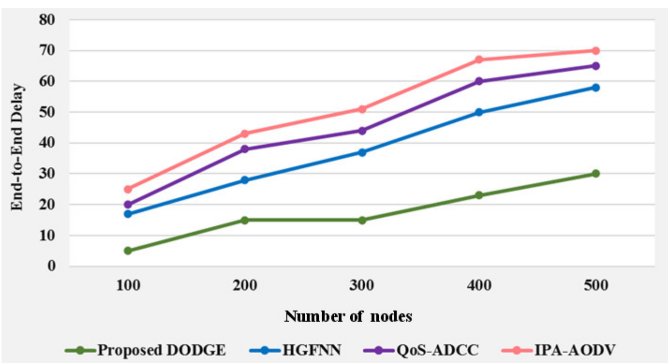

Fig. 9. Comparison of delay rate

Figure10 shows the comparison of throughput with existing techniques. Network TP is the average rate at which a message is successfully delivered via a communication link.

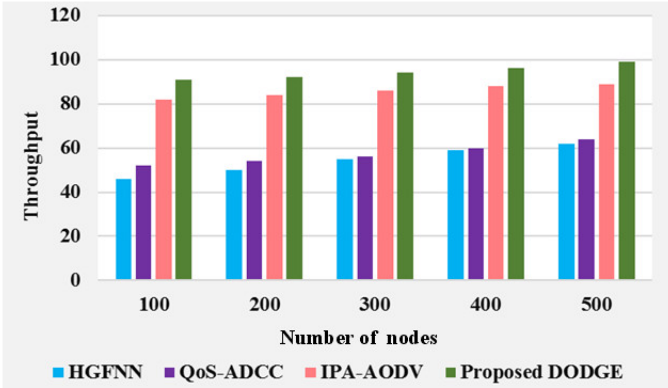

Fig. 10. Comparison of throughput

According to the observations of results in regular MANET during different assaults, throughput reduces dramatically; nevertheless, network performance in the proposed network is unaffected by the attack.

The proposed DODGE has a substantially lower power consumption than the HGFNN, QoS-ADCC, and IPA-AODV that are currently in use, as shown in Fig. 11. As a result, the proposed DODGE uses less network power than the existing method. DR is used to identify the attacker's IP address, detect vulnerable activity, and contact the coordinator to check further features.

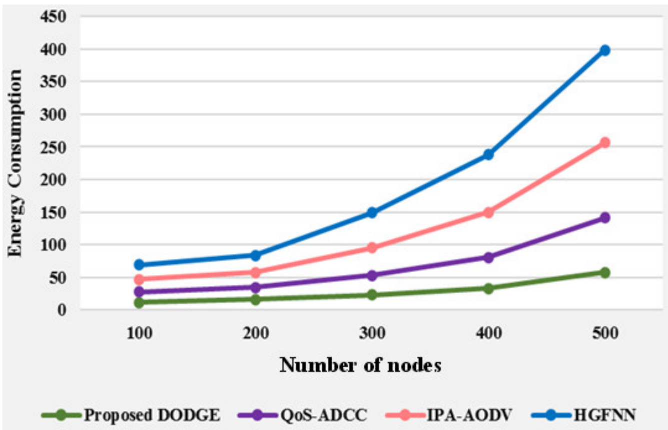

Fig. 11. Comparison of energy consumption

Figure 12 depicts the average time needed to uncover a single assault. When compared to existing approaches, DODGE has the fastest attack detection time and is 87% more effective than HGFNN, QoS-ADCC, and IPA-AODV.

Figure 13 compares network lifetime with existing approaches. It is defined as the amount of time that sensor networks in a MANET dedicate to sensing.

Figure 14 displays a box plot that displays the maximum latency, first quartile, median, third quartile, and lowest latency for each of the servers 1, 3, 7, and 15. The test will run for 600

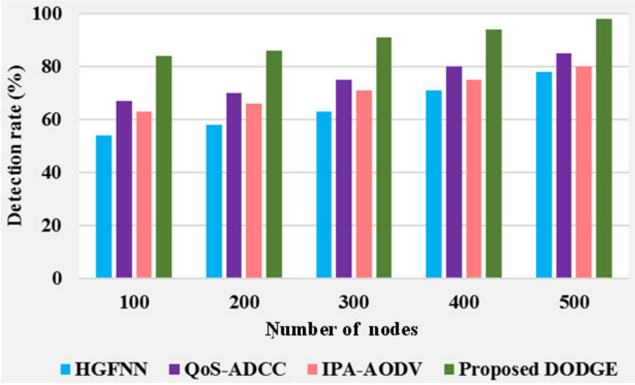

Fig. 12. Comparison of detection rate

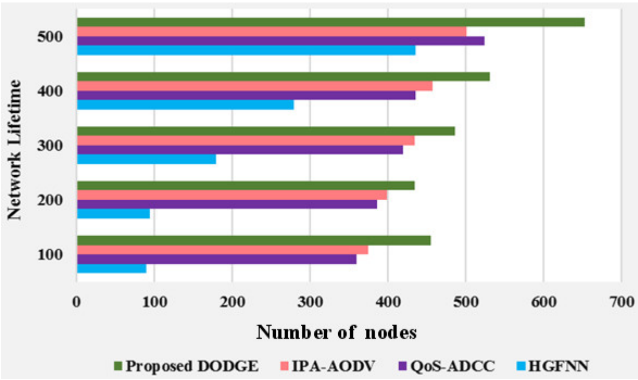

Fig. 13. Comparison of network lifetime

seconds for each node that is, 20, 40, 60, 80, and 100 during each update. For a given user count, all nodes incur roughly the same delay.

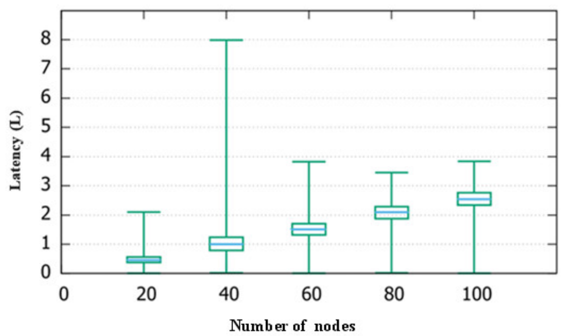

Fig. 14. Latency vs no. of nodes

4.3. Comparison analysis

The proposed SCNN-BiLSTM is compared with current DL models such as BiLSTM, DNN, and CNN for congested node prediction. Table 3 compares the various algorithms such as BiLSTM, DNN, and CNN and it predicts the congested node in the shortest time.

The proposed SCNN-BiLSTM improves the overall accuracy of BiLSTM, DNN, and CNN by 10.05%, 6.59%, and 3.26%, respectively. And it indicates that the SCNN-BiLSTM achieved better results than existing techniques.

Table 3
Comparative analysis of existing DL networks and SCNN-BiLSTM

Network	AC	PR	RE	SP
BiLSTM	89.26	87.11	78.69	82.38
DNN	92.72	90.35	86.29	89.94
CNN	96.05	94.93	90.56	87.42
SCNN-BiLSTM (Proposed)	99.31	98.37	96.05	93.52

5. CONCLUSION

This paper, a novel DODGE technique to predict the congested node effectively. The DODGE scheme is simulated by using NS2. In terms of PDR, DE, TP, EC, L, DR, and NL. The DODGE is compared to more established techniques like HGFNN, QoS-ADCC, and IPA-AODV. The DODGE model increases the TP, PDR, and reduces DE, and DR, with lower packet loss and considerably lower computational complexity. Moreover, the findings show that the SCNN-BiLSTM model is compared with existing techniques such as BiLSTM, DNN, and CNN. The proposed SCNN-BiLSTM improves the overall accuracy better than 10.05%, 6.59%, and 3.26% for BiLSTM, DNN, and CNN, respectively, in predicting the congested node in the shortest time. Soon, the system will be improved to identify more types of network assaults by adding new parameters.

REFERENCES

[1] V.K. Krishnamoorthy *et al.*, "Energy saving optimization technique-based routing protocol in Mobile ad-hoc network with IOT Environment," *Energies*, vol. 16, no. 3, p. 1385, 2023, doi: 10.3390/en16031385.

[2] A.R. Aarthy and K. Vinoth Kanth, "DNA Encryption-Based Secure Access Control and Data Sharing in an Enabled Cloud Environment," *Int. J. Syst. Des. Comput.*, vol. 2, no. 1, pp. 14–19, 2024.

[3] S. Ghosh, A. Banerjee, A. Sufian, S.K. Gupta, S.H. Alsamhi, and A. Saif, "Efficient selfish node detection using SVM in IoT-MANET environment," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 12, p. e4858, 2023, doi: 10.1002/ett.4858.

[4] A. Khalid, R.A. Rehman, and M. Burhan, "CBILEM: A novel energy aware mobility handling protocol for SDN based NDN-MANETs," *Ad Hoc Netw.*, vol. 140, p. 103049, 2023, doi: 10.1016/j.adhoc.2022.103049.

[5] H. Kabir, M. L. Tham and Y.C. Chang, "Internet of robotic things for mobile robots: concepts, technologies, challenges, applications, and future directions," *Digital Commun. Netw.*, vol. 9, no. 6, pp. 1265–1290, 2023, doi: 10.1016/j.dcan.2023.05.006.

[6] Z.E. Ahmed, A.A. Hashim, R.A. Saeed, and M.M. Saeed, "Mobility management enhancement in smart cities using software defined networks," *Sci. Afr.*, vol. 22, p. e01932, 2023, doi: 10.1016/j.sciaf.2023.e01932.

[7] R. Vatambeti, S.V. Mantena, K.V.D. Kiran, S. Chennupalli, and M.V. Gopalachari, "Black hole attack detection using Dolphin Echo-location-based machine learning model in MANET envi-

ronment," *Comput. Electr. Eng.*, vol. 114, p. 109094, 2024, doi: 10.1016/j.compeleceng.2024.109094.

[8] M. Amanullakhan, M. Usha, and S. Ramesh, "Intrusion Detection Architecture (IDA) In IOT Based Security System," *Int. J. Comput. Eng. Optimiz.*, vol. 1, no. 1, pp. 33–42, 2023.

[9] A. Abdelhamid, M.S. Elsayed, A.D. Jurcut, and M.A. Azer, "A lightweight anomaly detection system for black hole attack," *Electronics,* vol. 12, no. 6, p. 1294, 2023, doi: 10.3390/electronics12061294.

[10] A. Maheshwari, R.K. Yadav and P. Nath, "Congestion aware data transmission in mobile and constrained IoT network," *Wireless Pers. Commun.*, vol. 130, no. 3, pp. 2121–2136, 2023, doi: 10.1007/s11277-023-10374-8.

[11] X. Zheng, Y., Zhao, J. Lee, and W. Chen, "Multi-agent deep reinforcement learning for cross-layer scheduling in mobile adhoc networks," *China Commun.,* vol. 20, no. 8, pp. 78–88, 2023, doi: 10.23919/JCC.fa.2022-0496.202308.

[12] N. Mast, S. Khan, M.I. Uddin, Y.Y. Ghadi, H.K. Alkahtani, and S.M. Mostafa, "A cross-layer solution for contention control to enhance tcp performance in wireless ad-hoc networks," *IEEE Access*, vol. 11, pp. 24875–24893, 2023, doi: 10.1109/ACCESS.2023.3244888.

[13] S.A. Rana and B. Harsoor, "A novel traffic controlling model for enhancing routing in mobile Ad-Hoc network," *Mater. Today Proc*., vol. 81, pp. 1141–1147, 2023, doi: 10.1016/j.matpr.2021.04.417.

[14] A.K. Budati, S. Islam, M.K. Hasan, N. Safie, N. Bahar, and T.M. Ghazal, "Optimized visual internet of things for video streaming enhancement in 5G sensor network devices," *Sensors*, vol. 23, no. 11, p. 5072, 2023, doi: 10.3390/s23115072.

[15] Y. Jiang, H. Sun, and M. Yang, "AODV-EOCW: An Energy-Optimized Combined Weighting AODV Protocol for Mobile Ad Hoc Networks," *Sensors*, vol. 23, no. 15, p. 6759, 2023, doi: 10.3390/s23156759.

[16] H.K. Molia and A.D. Kothari, "TCP-RLLD: TCP with reinforcement learning based loss differentiation for mobile adhoc networks," *Wireless Netw.*, vol. 29, no. 5, pp. 1937–1948, 2023, doi: 10.1007/s11276-023-03254-3.

[17] I. Ambika, S. Bhatia, S. Basheer, and P. Dadheech, "Optimized Resource Allocation and Queue Management for Traffic Control in MANET," *Comput. Syst. Sci. Eng.*, vol. 45, no. 2, pp. 1323–1342, 2023, doi: 10.32604/csse.2023.030786.

[18] D. Krishnamoorthy *et al.*, "An effective congestion control scheme for MANET with relative traffic link matrix routing," *Arab. J. Sci. Eng.*, vol. 45, pp. 6171–6181, 2020, doi: 10.1007/s13369-020-04511-9.

[19] R. Saraswathi, J. Srinivasan, and S. Aruna, "An Energy Efficient Routing Protocol and Cross Layer Based Congestion Detection Using Hybrid Genetic Fuzzy Neural Network (HGFNN) Model for MANET," *J. Algebr. Stat.*, vol. 13, no. 2, pp. 1007–1019, 2022, doi: 10.52783/jas.v13i2.255.

[20] S. Kanthimathi and P. JhansiRani, "Optimal routing-based load balanced congestion control using MAODV in WANET environment," *Int. J. Adv. Comput. Sci. Appl.*, vol.12, no. 3, 2021, doi: 10.14569/IJACSA.2021.0120348.

[21] K. Muthulakshmi, K. Kalirajan, J. Jean Justus, and P. Sivamalar, "QoS Aware Data Congestion Control Routing in Mobile Ad Hoc Networks for Intelligent Transportations Systems," *Tehnički Vjesnik*, vol. 31, no. 1, pp. 240–246, 2024, doi: 10.17559/TV-20230427000582.

[22] N. Rajendran and E.R. Naganathan, "Collaborative Opportunistic Zone Division Clustering and Routing Algorithm for Congestion Control and Load Balancing on Manet," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, pp. 1111–1120, 2020, doi: 10.34218/IJARET.11.10.2020.107.

[23] S. Mohan and D. Vimala, "Enhancing Congestion Control and QoS Scheduling using Novel Rate Aware-Neuro-Fuzzy Algorithm in MANET," *J. Theor. Appl. Inf. Technol.*, vol. 101, no. 11, pp. 4819-4832, 2023.

[24] M.P. Nallayam Perumal and C.K. Selvi, "Improved priority aware mechanism for enhancing QoS in MANET," *Wirel. Pers. Commun.*, vol. 122, no. 1, pp. 277–292, 2022, doi: 10.1007/s11277-021-08898-y.

[25] M.K. Dholey and D. Sinha, "Acolbr: Aco Based Load Balancing Routing in Manet," *Wirel. Pers. Commun.*, vol. 126, no. 3, pp. 2483–2511, 2022, doi: 10.1007/s11277-022-09804-w.

[26] M. Arun and R. Jayanthi, "An adaptive congestion and energy aware multipath routing scheme for mobile ad-hoc networks through stable link prediction," Measurement: *Sensors,* vol. 30, p. 100926, 2023, doi: 10.1016/j.measen.2023.100926.

[27] B. Devika and P.N. Sudha, "Power optimization in MANET using topology management," *Eng. Sci. Technol*., vol. 23, no. 3, pp. 565–575, 2020, doi: 10.1016/j.jestch.2019.07.008.

[28] P.J. Shermila, A. Ahilan, M. Shunmugathammal, and J. Marimuthu, "DEEPFIC: Food item classification with calorie calculation using dragonfly deep learning network," *Signal Image Video Process.*, vol. 17, no. 7, pp. 3731–3739, 2023, doi: 10.1007/s11760-023-02600-4.

[29] D. Vanathi, K. Anusha, A. Ahilan, and A.S.E. Suniram, "Software cost and effort estimation using dragonfly whale optimized multilayer perceptron neural network," *Alex. Eng. J.*, vol. 103, pp. 30–37, 2024, doi: 10.1016/j.aej.2024.04.043.