# Percentile-Based DoS Attack Detection and Severity Level Identification in Computer Networking Using SS-LH-BiLSTM

**Amaresan Venkatesan**

Email: *v.amaresan[at]gmail.com*

**Abstract:** ***By overwhelming the network functions with a flood of illegitimate requests, Denial of Service (DoS) attacks disrupt them. But, identifying the severity of the DoS attacks wasn't concentrated in any of the conventional studies. Thus, by utilizing Sinc-Softmin and Lecun-He-based Bidirectional Long Short-Term Memory (SS-LH-BiLSTM), a DoS attack detection and severity identification model is proposed. Primarily, the data are gathered and preprocessed. Next, the data are segmented as well as sorted in ascending order. Then, centered on the duration of data transmission as well as the total number of data packets, the percentile threshold is estimated. After that, by utilizing the percentile value, the normal and abnormal behavior of the data is determined. For the instances of abnormal data, features are extracted and optimal features are chosen for identifying the attack severity. By utilizing SS-LH-BiLSTM, the DoS attack types and their severity levels are detected. Rate limiting is processed if a low or medium-level attack is detected; or else, the data is blocked. As per the experimental analysis, the proposed system outperformed conventional systems by achieving 98.89% accuracy.***

**Keywords:** DoS, Percentile threshold, Attack Detection System (ADS), Computer Network (CN), Sliding Window (SW), SZPi-Fuzzy Inference System (SZP-FIS), Glorot initialization-based Seagull Optimization Algorithm (GSOA), and Weighted Round Robin-based Token Bucket Algorithm (WRR-TBA).

## 1. Introduction

Currently, the backbone of modern society is the Computer Network (CN) (Zhang et al., 2021). To ensure reliable communication among the CNs, the cyber-physical system uses advanced technologies (Hussain et al., 2020). Yet, the integration of modern communication with advanced technologies exposed a wide range of security threats (Liu et al., 2020). Among all, DoS is regarded as one of the most severe threats on the internet (Kurniawan & Yazid, 2020). This is because DoS hinders the whole network services and makes it unavailable for its legitimate users (Yue et al., 2020). The network is overwhelmed with a flood of illegitimate traffic requests by DoS attackers, thus leading to server downtime (Wang et al., 2020).

Prevailing works developed several methodologies to differentiate normal and abnormal network traffic (Kshirsagar & Kumar, 2021) (Gao et al., 2019). Deep Belief Network (DBN), Deep Neural Network (DNN), and Long Short-Term Memory (LSTM) are some methodologies (Fauzi et al., 2020) (Zhijun et al., 2020). However, the severity of the DoS attacks was not detected in any of the traditional works. Hence, this framework proposes an efficient DoS attack detection and its severity identification model by utilizing SS-LH-BiLSTM.

### 1.1. Problem statement:

- None of the existing works focused on identifying the severity levels of the DoS attacks.
- (Ramesh et al., 2021) employed the static threshold-centric ADS that reduced the system's flexibility and reliability.
- (Baig et al., 2020) didn't employ any control mechanisms while performing penetration testing and large-scale ethical hacking.
- Choosing the most suitable feature set is one of the promising tasks in (Qu et al., 2019).

### 1.2 Objectives

- The proposed framework identifies the severity levels of various types of DoS attacks.
- The proposed model uses the percentile threshold-based DoS attack detection system.
- To control the traffic flow, the rate-limiting centered on the proposed WRR-TBA is employed.
- By using the proposed GSOA method, the optimal features are selected.

The paper's structure is given as: the literature survey is presented in Section 2, the proposed technique is explicated in Section 3, the results are given in Section 4, and the paper is concluded in Section 5.

## 2. Literature Survey

(Ramesh et al., 2021) established an optimized DNN-centric DoS attack detection model in a wireless multimedia sensor network. According to the evaluation outcomes, the developed model consumed minimum network resources. However, this system utilized static threshold-centric attack detection.

(Baig et al., 2020) explored the adoption and redefinition of the averaged one-dependence as well as two-dependence-based DoS detection in Internet-of-Things networks. The analysis outcomes proved this model's superiority regarding detection accuracy. Still, this model didn't use any controlling mechanisms during real-time testing.

(Qu et al., 2019) explained a DoS detection model utilizing statistics-enhanced direct batch growth self-organizing mapping. According to the experimental outcomes, this model was less error-prone. But, this model didn't choose the most suitable features efficiently.

(Gao et al., 2020) propounded the packet filter and flow rule management-centric DoS attack detection as well as mitigation in a software-defined network. The evaluation outcomes proved that the DoS attack was precisely identified and mitigated by this model. Nevertheless, this model was highly resource-intensive. (Tang et al., 2021) presented a self-adaptive density-centric spatial clustering of applications with a noise algorithm-centric low-rate DoS attack detection model. The analysis outcomes stated that this model was efficiently adapted to large-scale complex networks. Still, this model had a time complexity issue.

## 3. Proposed Methodology for DoS Attack Detection and Severity Identification System using SS-LH-BiLSTM

Here, an effective DoS attack detection and severity identification model is proposed by utilizing SS-LH-BiLSTM. Figure 1 explains the proposed model's architecture.

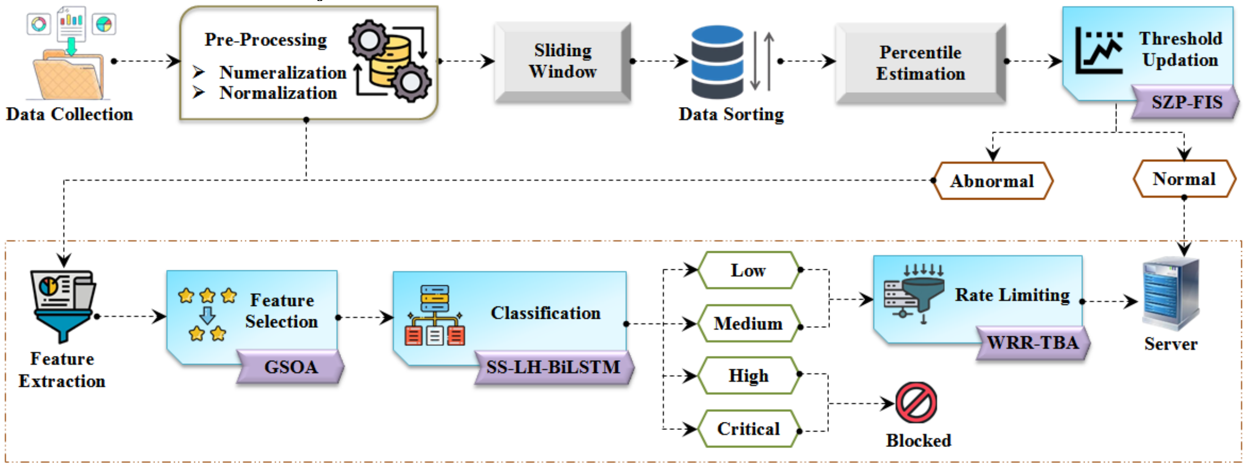


**Figure 1:** Architecture of the proposed model

### 3.1 Data collection

Primarily, a $l$ number of data $(Z)$ is gathered from the dataset, and it is given as,

$$Z = \{Z_1, Z_2, \ldots\ldots, Z_l\} \tag{1}$$

Next, the gathered $Z$ are processed further.

### 3.2. Preprocessing

Then, the gathered $Z$ are preprocessed under numeralization and normalization processes. The preprocessed data $(Z_{pre})$ are segmented further.

### 3.3. Sliding window

After that, $Z_{pre}$ is segmented into a particular time range $(\varepsilon)$ by using the SW technique based on the following expression,

$$Z_{sli} = \varepsilon(Z_{pre}) \tag{2}$$

Where, $Z_{sli}$ is the segmented data.

### 3.4. Data sorting

Next, to evaluate the percentile index, the segmented $Z_{sli}$ is sorted in ascending order. Hence, the sorted data is signified as $Z_{sort}$.

### 3.5. Percentile estimation

After that, the percentile index $(\mathrm{P})$ is estimated based on the duration of data transmission $(\mathrm{H})$ and the total number of data packets $(\mathrm{K})$ in $Z_{sort}$. The percentile threshold is an adaptive threshold that could automatically adapt to the normal data pattern changes by recalculating the percentile periodically. The percentile indicates the upper bound of the normal traffic behavior, and it is expressed as,

$$\mathrm{P} = \frac{\mathrm{H}}{100} \times (\mathrm{K} + 1) \tag{3}$$

Where, the $\mathrm{P}^{th}$ percentile determines the threshold for normal and abnormal data.

### 3.6. Threshold updation

After finding $\mathrm{P}$, the initial threshold is generated for the normal and abnormal data by utilizing SZP-FIS. The FIS is simple, easy to implement, and computationally efficient. However, it had difficulty in tuning the fuzzy sets. Thus, the SZPi-shaped membership function is deployed. Firstly, the conditions are set in the rule-base of SZP-FIS, and it is presented as,

$$\aleph = \begin{cases} Normal, & if\ (\kappa < \mathrm{P}) \\ Abnormal, & else \end{cases} \tag{4}$$

Where, $\kappa$ is the incoming data from $Z_{sort}$. The fuzzification $(\varsigma)$ block of SZP-FIS is utilized for mapping the input $(\aleph)$ into a fuzzy value $(\hat{\aleph})$ and is given as,

$$\aleph \xrightarrow{\varsigma} \hat{\aleph} \tag{5}$$

Next, the inference engine is responsible for making decisions by utilizing the SZPi-shaped membership function $(\vartheta)$, which is signified as,

$$\vartheta = \begin{cases} 0, & if\ \hat{\aleph} \le a' \\ 2\left(\dfrac{\hat{\aleph}-a'}{b'-a'}\right)^2, & if\ a' < \hat{\aleph} \le \dfrac{a'+b'}{2} \\ 1-2\left(\dfrac{b'-\hat{\aleph}}{b'-a'}\right)^2, & if\ \dfrac{a'+b'}{2} < \hat{\aleph} \le b' \\ 1, & if\ b' < \hat{\aleph} \le c' \\ 1-2\left(\dfrac{\hat{\aleph}-c'}{d'-c'}\right)^2, & if\ c < \hat{\aleph} \le \dfrac{c'+d'}{2} \\ 2\left(\dfrac{d'-\hat{\aleph}}{d'-c'}\right), & else\ \dfrac{c'+d'}{2} < \hat{\aleph} \le d' \end{cases} \tag{6}$$

Here, $a', b'c'\ and\ d'$ symbolize the function parameters. The last defuzzification $(\tilde{\varsigma})$ functional block converts the fuzzy quantity into crisp values, and it is given as,

$$\hat{\aleph} \xrightarrow{\tilde{\varsigma}} \aleph \tag{7}$$

Hence, the output acquired is indicated as,

$$\aleph = \{\aleph_1, \aleph_2\} \tag{8}$$

Where, $\aleph_1\ and\ \aleph_2$ are the normal and abnormal data, correspondingly. The input condition $(\aleph)$ is periodically updated based on the incoming data $(\kappa)$. If $\aleph_1$ is obtained, then the data is further proceeded to run on its application; or else, the following process is carried out. The pseudocode for SZP-FIS is given below.

```
Input: Percentile $(\mathrm{P})$
Output: Normal and abnormal data $(\aleph_1\ and\ \aleph_2)$
Begin
    Initialize thresholds for data behavior
    For each $\mathrm{P}$ do
        Generate $\aleph = \begin{cases} Normal, & if\ (\kappa < \mathrm{P}) \\ Abnormal, & else \end{cases}$
        Compute $\aleph \xrightarrow{\varsigma} \hat{\aleph}$
        Evaluate SZPi-shaped membership function $(\vartheta)$
        Process $\hat{\aleph} \xrightarrow{\tilde{\varsigma}} \aleph$
    End for
    Return $\aleph_1\ and\ \aleph_2$
End
```

### 3.7. DoS types and severity identification

For the instance of $\aleph_2$, the type of DoS attack and its severity levels are known for taking appropriate mitigation measures.

#### 3.7.1. Feature extraction

The features, such as timestamp, severity, IP, protocol, port, hostname, tag, autonomous system number, country, region, city, North American industry classification system code, hostname_source, sector, detail, and application layer protocol are extracted from $Z_{pre}$. Hence, the extorted features are signified as $Z_{ext}$.

#### 3.7.2. Feature selection

Then, the optimal features are chosen from $Z_{ext}$ by utilizing GSOA. Exploration and exploitation are efficiently balanced by the seagull optimization algorithm. However, SSA easily falls in the local optimal solutions. Thus, for overcoming this issue, the initial seagull's position is generated by using Glorot initialization.

**(a) Initialization:** Primarily, the population of the seagull (i.e., extracted features) is initialized as,

$$S = \{S_1, S_2, \ldots\ldots, S_O\} \tag{9}$$

Where, $O$ signifies the total number of seagulls in the population. The fitness function $(\zeta)$ for the population achieves maximum $(\max)$ classification accuracy $(ac)$, and it is given as,

$$\zeta = \max\{ac\} \tag{10}$$

By utilizing the Glorot initialization approach, the initial position of the seagull $(S)$ is generated, which is given as,

$$S = \mu\left(-\sqrt{\frac{6}{\psi_{up}+\psi_{lo}}}, \sqrt{\frac{6}{\psi_{up}+\psi_{lo}}}\right) \tag{11}$$

Where, $\mu$ is the uniform distribution and $\psi_{up}$ *and* $\psi_{lo}$ signify the search place's upper and lower bounds, correspondingly.

**(b) Migration behavior:** After that, the movement of the seagull during the migration process is represented under three conditions, such as

***(i) Collision avoidance:*** For avoiding collision with other seagulls, the individual seagull's position is updated using an additional variable $(Q)$, and it is given as,

$$\overrightarrow{S_C} = Q \times \vec{S}(t) \tag{12}$$

$$Q = v - \left(t \times \left[\frac{v}{t_{\max}}\right]\right) \tag{13}$$

Here, $\overrightarrow{S_C}$ symbolizes the new position of a seagull, which doesn't collide with other seagulls, $\vec{S}(t)$ is the seagull's current position, and $v$ signifies the hyperparameter.

***(ii) Move in the direction of the best neighbor:*** An individual seagull moves in their best neighbor's direction for avoiding overlapping with other seagulls, which is given as,

$$\overrightarrow{S_M} = W \times \left(\overrightarrow{S_{best}}(t) - \vec{S}(t)\right) \tag{14}$$

$$W = 2 \times Q^2 \times rand \tag{15}$$

Where, $\overrightarrow{S_M}$ is the direction of the best neighbor, $\overrightarrow{S_{best}}(t)$ symbolizes the best seagull's position, $rand$ is the random number, and $W$ signifies another controlling parameter based on $Q$.

***(iii) Approach to the best seagull's location:*** Next, the individual seagull moves to the global optimal direction $\left(\overrightarrow{S_G}\right)$, which is expressed as,

$$\overrightarrow{S_G} = \left|\overrightarrow{S_C} + \overrightarrow{S_M}\right| \tag{16}$$

Moreover, this $\overrightarrow{S_G}$ determines the distance between the current seagull and the optimal seagull.

**(c). Attacking behavior:** The seagulls depend on their wings as well as weight for attacking the prey. While the seagull attacks, its motion in 3-dimensional planes $(X, Y, and\ Z)$ is given as,

$$X = s \times \cos(q) \tag{17}$$

$$Y = s \times \sin(q) \tag{18}$$

$$Z = s \times q \tag{19}$$

$$s = U \times \exp(q \cdot V) \tag{20}$$

Where, $s$ is the flight radius, $q$ signifies a random number within the range of $[0, 2\pi]$, $\exp$ is the standard exponential function, as well as $U$ *and* $V$ are the constants utilized for defining the spiral shape. Hence, the updated seagull's position is expressed as,

$$\overrightarrow{S_{upd}}(t) = \left(\overrightarrow{S_G} \times X \times Y \times Z\right) + \overrightarrow{S_{best}}(t) \tag{21}$$

Here, $\overrightarrow{S_{upd}}(t)$ is the updated seagull's position in $t$ and it is the combination of migration and attacking behavior to find the optimal seagull's position. The updated seagull's position $\left(\overrightarrow{S_{upd}}\right)$ is the optimal feature acquired from GSOA.

### 3.7.3. Classification

Then, based on $\overrightarrow{S_{upd}}$, the types of DoS attacks and their severity level are detected by using SS-LH-BiLSTM. The input sequence is processed in both forward and backward directions by BiLSTM. But, BiLSTM is susceptible to overfitting and vanishing gradient issues. Thus, a Sinc-Softmin activation and Lecun-He-based weight initialization techniques are proposed. Figure 2 explains the proposed SS-LH-BiLSTM's structure.

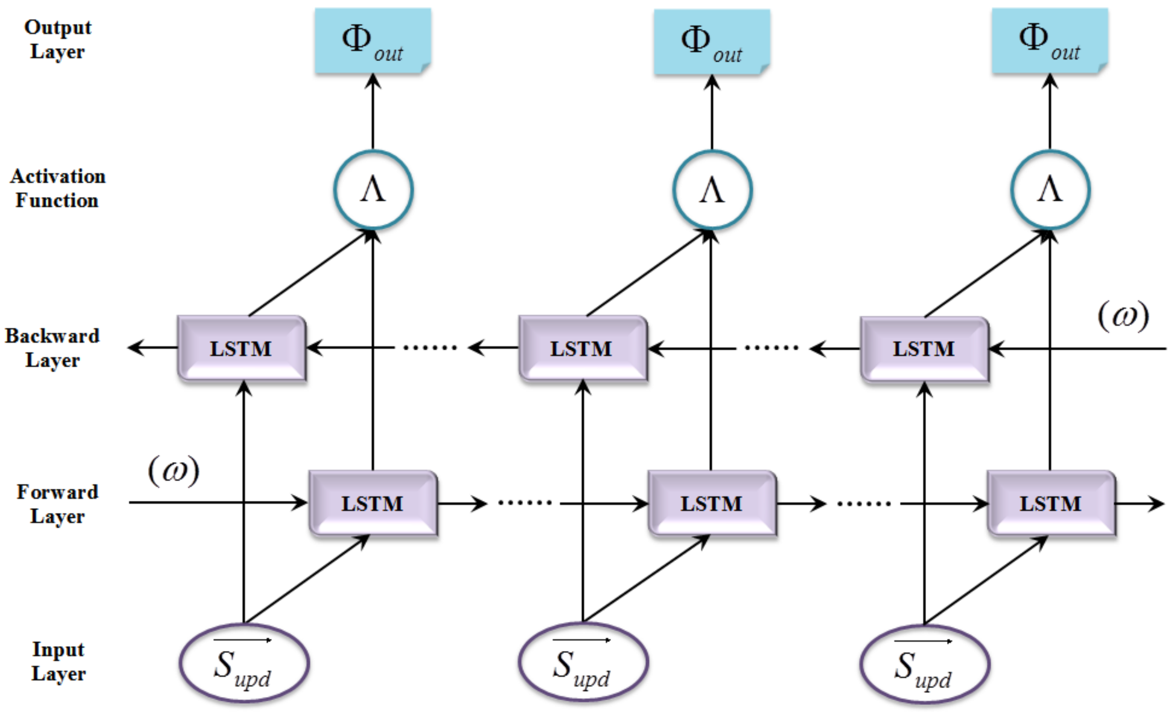


**Figure 2:** Structure of SS-LH-BiLSTM

Initially, $\overrightarrow{S}_{upd}$ is given to the input layer $(\Phi_{inp})$. Next, it is forwarded to the bidirectional functional layer in which $\overrightarrow{S}_{upd}$ is processed in ascending order in the forward layer and descending order in the backward layer. After that, the forget gate $(\Phi_{for})$ removes the unused information in the previous cell state $(\Phi_{cell-1})$ and is given as,

$$\Phi_{for} = \Lambda \cdot \left(\omega . \left[w_{T-1}, \overrightarrow{S}_{upd}\right] + \beta_{for}\right) \qquad (22)$$

Here, $\beta_{for}$ is the bias factor of the forget gate, $w_{T-1}$ is the previous hidden state, and $\Lambda$ is the Sinc-Softmin activation function, which is signified as,

$$\Lambda = \frac{e^{\left(-\frac{\sin\left(\pi \overrightarrow{S}_{upd}\right)}{\pi \overrightarrow{S}_{upd}}\right)}}{\sum e^{\left(-\frac{\sin\left(\pi \overrightarrow{S}_{upd}\right)}{\pi \overrightarrow{S}_{upd}}\right)}} \qquad (23)$$

Where, $\pi$ is the scaling parameter, and $e$ is the standard exponential function. The weight factor $(\omega)$ is determined by utilizing the Lecun-He-based weight initialization method, which is given as,

$$\omega = \eta\left(0, \frac{3}{2 \cdot \ell\left(\overrightarrow{S}_{upd}\right)}\right) \qquad (24)$$

Here, $\eta$ is the normal distribution and $\ell\left(\overrightarrow{S}_{upd}\right)$ is the number of input units. Then, the input layer controls the new information as given below,

$$\Phi_{inp} = \Lambda \cdot \tilde{\Phi}_{cell}\left(\omega . \left[w_{T-1}, \overrightarrow{S}_{upd}\right] + \beta_{inp}\right) \qquad (25)$$

Here, $\beta_{inp}$ is the bias factor of the input gate and $\tilde{\Phi}_{cell}$ is the vector function of the cell state $(\Phi_{cell})$. This cell state is the memory of the network and is expressed as,

$$\tilde{\Phi}_{cell} = \tanh \cdot \left(\omega \cdot \left[w_{T-1}, \overrightarrow{S}_{upd}\right] + \beta_{cell}\right) \qquad (26)$$

$$\Phi_{cell} = \Phi_{for} \circ \Phi_{cell-1} + \Phi_{inp} \circ \tilde{\Phi}_{cell} \qquad (27)$$

Lastly, the output gate $(\Phi_{out})$ is responsible for maintaining the information in $\Phi_{cell}$ and deciding the next hidden state $(w_T)$. It is given as,

$$\Phi_{out} = \Lambda \cdot \left(\omega \cdot \left[w_{T-1}, \overrightarrow{S}_{upd}\right] + \beta_{out}\right) \qquad (28)$$

$$w_T = \Phi_{out} \cdot \tanh\left(\Phi_{cell}\right) \qquad (29)$$

Where, $\beta_{out}$ is the bias factor of the output gate. Next, the loss function $(\Psi)$ is evaluated. Hence, the output gate is obtained with the DoS attack types and severity levels, namely low $(\lambda_1)$, medium $(\lambda_2)$, high $(\lambda_3)$, and critical $(\lambda_4)$. If $\lambda_3$ *and* $\lambda_4$ are obtained, then the data is blocked here; or else, the following process is proceeded further. The pseudocode for the proposed SS-LH-BiLSTM is given below.

```
Input: Optimal features $\left(\overrightarrow{S_{upd}}\right)$
Output: DoS attack type and severity level $\left(\Phi_{out}\right)$
Begin
    Initialize bias factor $(\beta)$ and previous cell state $\left(\Phi_{cell-1}\right)$
    For each $\overrightarrow{S_{upd}}$ do
        Formulate forget gate
        Activate $\Lambda = \frac{e^{\left(-\frac{\sin\left(\pi\overrightarrow{S_{upd}}\right)}{\pi\overrightarrow{S_{upd}}}\right)}}{\sum e^{\left(-\frac{\sin\left(\pi\overrightarrow{S_{upd}}\right)}{\pi\overrightarrow{S_{upd}}}\right)}}$
        Determine weight function
        Compute input gate
        Evaluate $\tilde{\Phi}_{cell} = \tanh\cdot\left(\omega\cdot\left[w_{T-1},\overrightarrow{S_{upd}}\right]+\beta_{cell}\right)$
        Process output gate
        Formulate $w_T = \Phi_{out}\cdot\tanh\left(\Phi_{cell}\right)$
        Estimate loss function $(\Psi)$
        If $(\Psi\ satisfy)${
                Stop
        } Else {
                Adjust the parameters
        }
        End If
    End For
    Return DoS attack type and severity level $\left(\Phi_{out}\right)$
End
```

### 3.7.4. Rate limiting

If $\lambda_1\ and\ \lambda_2$ are detected, these attacks might be mitigated by using WRR-TBA. The conventional TBA ensures smooth traffic flow. But, TBA had latency issues. For overcoming this issue, the Weighted Round-Robin technique is employed to schedule and allocate bandwidth in a way that can reduce the waiting time for the tokens. Initially, some random numbers of tokens are initialized in a bucket, and the bucket's capacity is signified as $\Re$. When the data from $\lambda_1\ and\ \lambda_2$ arrives for transmission, the WRR-TBA checks whether it has enough tokens to transmit the data, and it is expressed as,

$$p = \begin{cases} allow, & if\ p \le \Re \\ wait\,in\,queue, & otherwise \end{cases} \tag{30}$$

Where, $p$ is the sample packets from $\lambda_1\ and\ \lambda_2$. For reducing the waiting time of the packets in the queue, the weighted round-robin approach is employed by calculating the total weight of the queue as,

$$\delta_{tot} = \sum_{u=1}^{v}\delta_u \tag{31}$$

Here, $\delta_u$ is the weight of the queue, $v$ signifies the number of packets in the queue, and $\delta_{tot}$ is the total weight of the queue. Next, the bandwidth $(\alpha)$ of the queue is given as,

$$\alpha = \frac{\delta_u}{\delta_{tot}} \times \alpha_{tot} \tag{32}$$

Where, $\alpha_{tot}$ is the total available bandwidth. Then, the number of packets to be transferred is estimated in each round based on the expression given below,

$$p(\delta_u) = \frac{\delta_u}{\delta_{tot}} \times p_{tot} \tag{33}$$

Where, $p_{tot}$ is the total number of packets per round. Now, the tokens are initialized based on $p(\delta_u)$. Thus, the packets are controlled and smoothly transferred via the network. Therefore, by controlling the rate at which requests are processed by a server, the low and medium-level attacks are mitigated. Lastly, the attack-mitigated data are further processed in the network.

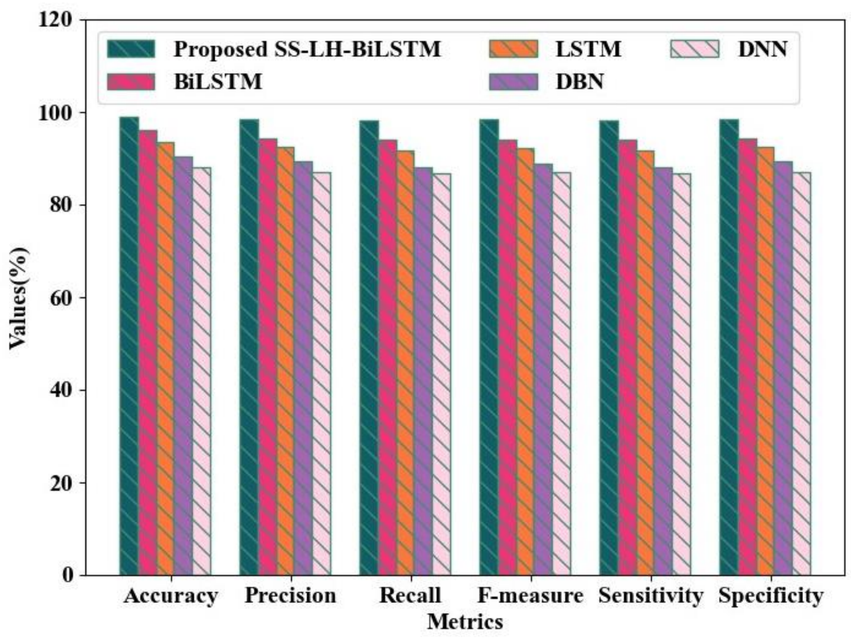
## 4. Result and Discussion

Here, the proposed model's robustness is proved by implementing it in the working platform of Python.

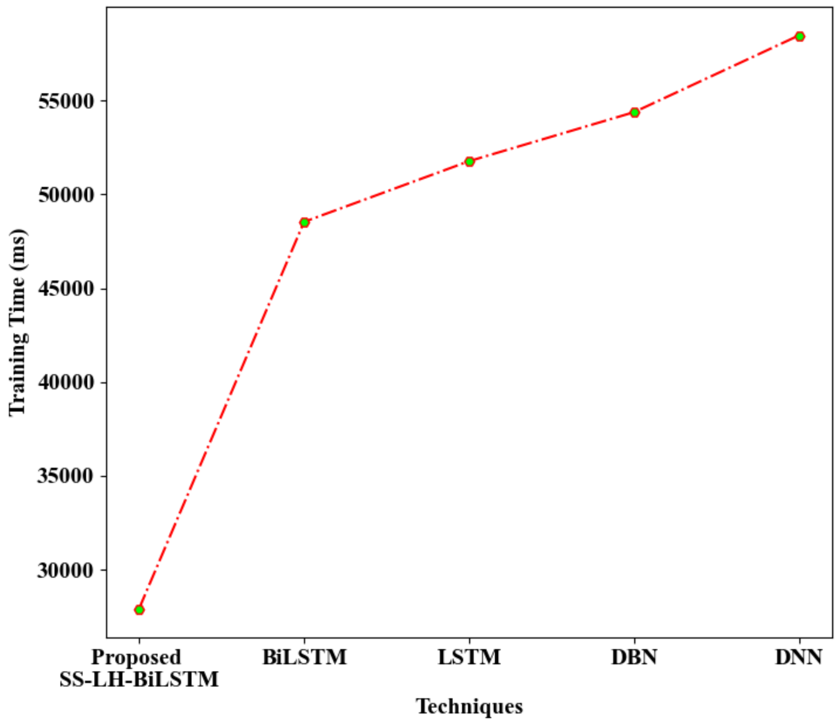
### 4.1 Dataset description

Here, the DoS attack data are gathered from the Loop DoS Report (LDoSR) data source. The used LDoSR contains attack types, attack vectors, target system, and impact. Among all the data in LDoSR, 80% of data is employed for training as well as 20% of data is utilized for testing the proposed work.

### 4.2 Performance evaluation



**(a)**



**(b)**

**Figure 3 (a) and (b):** Performance evaluation of proposed SS-LH-BiLSTM

In Figure 3, the performance analysis of the proposed SS-LH-BiLSTM and existing models, such as BiLSTM, LSTM, DBN, and DNN is described. The overfitting issues are reduced by the proposed model; thus, it achieved 98.89% accuracy, 98.37% precision, 98.29% recall, 98.33% f-measure, 98.29% sensitivity, 98.37% specificity, and 27926ms of training time. But, the prevailing models achieved relatively lower performance than the proposed model, thus proving the proposed SS-LH-BiLSTM's superiority.

**Table 1:** FPR and FNR analysis

| Techniques | FPR | FNR |
|---|---|---|
| Proposed SS-LH-BiLSTM | 0.013 | 0.031 |
| BiLSTM | 0.039 | 0.058 |
| LSTM | 0.065 | 0.072 |
| DBN | 0.088 | 0.098 |
| DNN | 0.107 | 0.119 |

False Positive Rate (FPR) and False Negative Rate (FNR) analyses of the proposed and prevailing models are indicated in Table 1. The FPR and FNR of the proposed SS-LH-BiLSTM are 0.013 and 0.031, correspondingly. But, the prevailing BiLSTM had 0.039 FPR and 0.058 FNR, and the other prevailing models also had lower performance. The proposed model attained better performance by reducing the vanishing gradient issue in the conventional model.

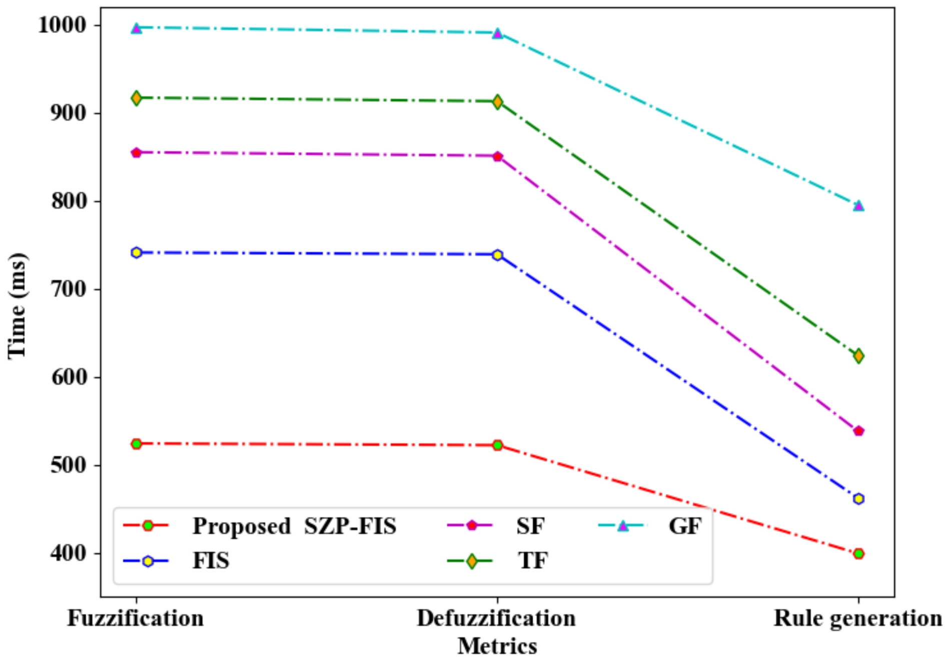


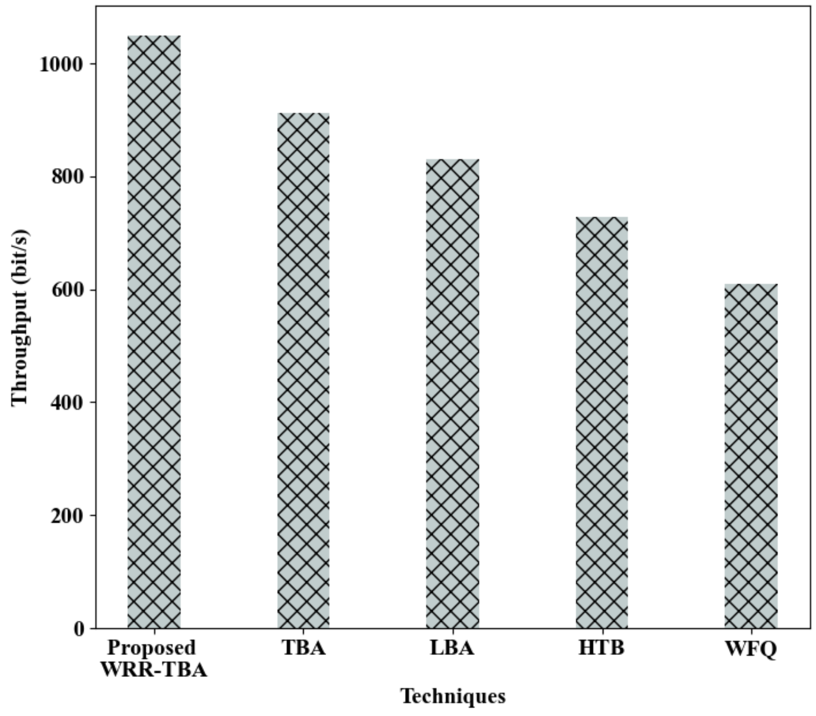
**Figure 4:** Performance assessment of proposed SZP-FIS

Figure 4 illustrates the fuzzification time, defuzzification time, and rule generation time analysis of the proposed and existing FIS, Sigmoid Fuzzy (SF), Trapezoidal Fuzzy (TF), and Gaussian Fuzzy (GF). Owing to efficient management of the fuzzy sets, the proposed SZP-FIS takes a minimum of 524ms, 522ms, and 399ms for fuzzification, defuzzification, and rule generation. But, the prevailing models take longer time than the proposed system. This proves the proposed model's robustness.

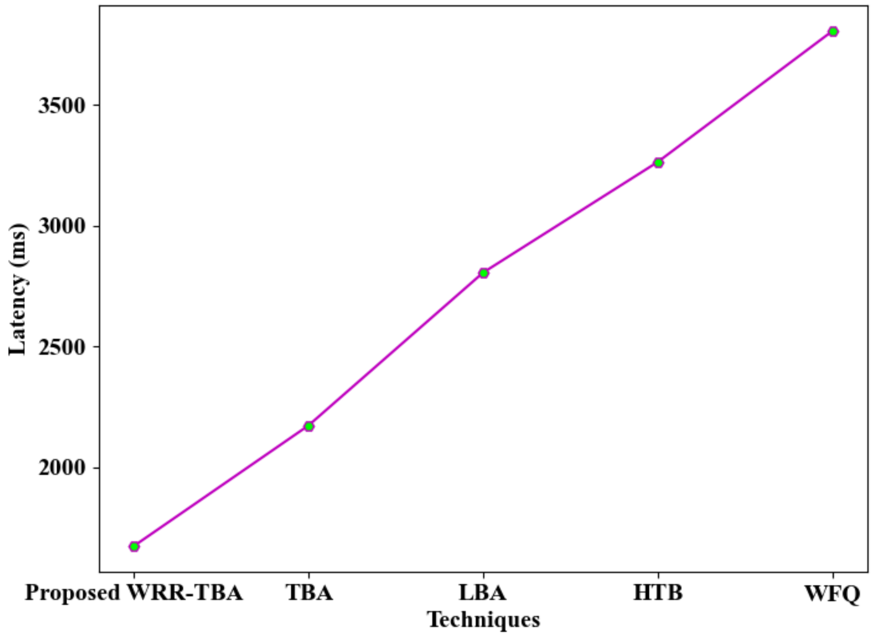
**Table 2:** Comparative analysis of proposed GSOA

| Methods | Average fitness (%) | Feature selection time (ms) |
|---|---|---|
| Proposed GSOA | 98.69 | 1627 |
| SOA | 96.27 | 2992 |
| RPOA | 93.96 | 3591 |
| GROA | 90.05 | 4927 |
| GAOA | 88.39 | 5849 |

In Table 2, the fitness analysis and selection time analysis of the proposed and the prevailing SOA, Red Panda Optimization Algorithm (RPOA), Gold Rush Optimization Algorithm (GROA), and Green Anaconda Optimization Algorithm (GAOA) are presented. The proposed model's average fitness and feature selection time are 98.69% and 1627ms, correspondingly. However, the prevailing SOA had 96.27% average fitness and 2992ms of feature selection time. This is because the existing SOA had fallen in the local optimal solutions. Therefore, the proposed system's efficiency is proven.



(a)



(b)

**Figure 5:** Throughput and latency analysis

The throughput and latency analysis of the proposed and existing TBA, Leaky Bucket Algorithm (LBA), Hierarchical Token Bucket (HTB), and Weighted Fair Queuing (WFQ) are given in Figure 5. The proposed work reduces the waiting time of the data; thus, it attains 1050 bit/s throughput and 1675ms latency. Still, the prevailing model had an average throughput and latency of 770.5 bit/s and 3013.25 ms, correspondingly. Hence, the analysis proves the proposed model's dynamic performance.

**Table 3:** Comparative analysis with related works

| Author name | Technique | Accuracy (%) |
|---|---|---|
| Proposed | SS-LH-BiLSTM | 98.89 |
| (Tang et al., 2020a) | Two-step cluster analysis | 98.06 |
| (SaiSindhuTheja & Shyam, 2021) | Recurrent neural network | 94.12 |
| (Wu et al., 2019) | Smith-waterman local sequence alignment | 86.88 |
| (Tang et al., 2020b) | Adaboost | 97.06 |
| (Amma et al., 2020) | Class scatter ratio and feature distance map | 95 |

In Table 3, the comparative analysis of the proposed and related works is presented. The proposed SS-LH-BiLSTM attained 98.89% accuracy in DoS attack detection and severity identification. But, when compared to the proposed system, the conventional works had considerably lower performances. Thus, the proposed model's significance is proved.

## 5. Conclusion

This work proposes a significant DoS attack detection and its severity identification framework using SS-LH-BiLSTM. The experimental analysis proves the proposed model's efficacy. As per the analysis outcomes, the proposed model attained 98.89% accuracy and 98.33% f-measure for attack type and severity detection, correspondingly. Moreover, the proposed system had a minimum of 27926ms training time, 0.013 FPR, and 0.031 FNR. The proposed model's average fitness was 98.69%. Moreover, the throughput and latency were 1050 bit/s and 1675ms, correspondingly. Thus, the analysis proves the proposed model's robustness.

***Future Recommendations***

Strong multiple layers of defense measures will be developed in the future for preventing and mitigating DoS attacks.

## References

[1] Amma, N. G. B., Selvakumar, S., & Velusamy, R. L. (2020). A Statistical Approach for Detection of Denial of Service Attacks in Computer Networks. *IEEE Transactions on Network and Service Management*, *17*(4), 1–12. https://doi.org/10.1109/TNSM.2020.3022799

[2] Baig, Z. A., Sanguanpong, S., Firdous, S. N., Vo, V. N., Nguyen, T. G., & So-In, C. (2020). Averaged dependence estimators for DoS attack detection in IoT networks. *Future Generation Computer Systems*, *102*, 198–209. https://doi.org/10.1016/j.future.2019.08.007

[3] Fauzi, M. A., Hanuranto, A. T., & Setianingsih, C. (2020). Intrusion Detection System using Genetic Algorithm and K-NN Algorithm on Dos Attack. *2nd International Conference on Cybernetics and Intelligent System*, 1–6. https://doi.org/10.1109/ICORIS50180.2020.9320822

[4] Gao, L., Li, Y., Zhang, L., Lin, F., & Ma, M. (2019). Research on Detection and Defense Mechanisms of DoS Attacks Based on BP Neural Network and Game Theory. *IEEE Access*, *7*, 43018–43030. https://doi.org/10.1109/ACCESS.2019.2905812

[5] Gao, S., Peng, Z., Xiao, B., Hu, A., Song, Y., & Ren, K. (2020). Detection and Mitigation of DoS Attacks in Software Defined Networks. *IEEE/ACM Transactions on Networking*, *28*(3), 1–15. https://doi.org/10.1109/TNET.2020.2983976

[6] Hussain, F., Abbas, S. G., Husnain, M., Fayyaz, U. U., Shahzad, F., & Shah, G. A. (2020). IoT DoS and DDoS Attack Detection using ResNet. *23rd IEEE International Multi-Topic Conference*, 1–6. https://doi.org/10.1109/INMIC50486.2020.9318216

[7] Kshirsagar, D., & Kumar, S. (2021). An efficient feature reduction method for the detection of DoS attack. *ICT Express*, *7*(3), 371–375. https://doi.org/10.1016/j.icte.2020.12.006

[8] Kurniawan, M. T., & Yazid, S. (2020). Mitigation and Detection Strategy of DoS Attack on Wireless Sensor Network Using Blocking Approach and Intrusion Detection System. *2nd International Conference on Electrical, Communication and Computer Engineering*, 1–5. https://doi.org/10.1109/ICECCE49384.2020.9179255

[9] Liu, L., Wang, H., Wu, Z., & Yue, M. (2020). The detection method of low-rate DoS attack based on multi-feature fusion. *Digital Communications and Networks*, *6*(4), 504–513. https://doi.org/10.1016/j.dcan.2020.04.002

[10] Qu, X., Yang, L., Guo, K., Ma, L., Feng, T., Ren, S., & Sun, M. (2019). Statistics-enhanced direct batch growth self-organizing mapping for efficient dos attack detection. *IEEE Access*, *7*, 78434–78441. https://doi.org/10.1109/ACCESS.2019.2922737

[11] Ramesh, S., Yaashuwanth, C., Prathibanandhi, K., Basha, A. R., & Jayasankar, T. (2021). An optimized deep neural network based DoS attack detection in wireless video sensor network. *Journal of Ambient Intelligence and Humanized Computing*, 1–14. https://doi.org/10.1007/s12652-020-02763-9

[12] SaiSindhuTheja, R., & Shyam, G. K. (2021). An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. *Applied Soft Computing*, *100*, 1–11. https://doi.org/10.1016/j.asoc.2020.106997

[13] Tang, D., Dai, R., Tang, L., & Li, X. (2020a). Low-rate DoS attack detection based on two-step cluster analysis and UTR analysis. *Human-Centric Computing and Information Sciences*, *10*(1), 1–20. https://doi.org/10.1186/s13673-020-0210-9

[14] Tang, D., Tang, L., Dai, R., Chen, J., Li, X., & Rodrigues, J. J. P. C. (2020b). MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost. *Future Generation Computer Systems*, *106*, 347–359. https://doi.org/10.1016/j.future.2019.12.034

[15] Tang, D., Zhang, S., Chen, J., & Wang, X. (2021). The detection of low-rate DoS attacks using the SADBSCAN algorithm. *Information Sciences*, *565*, 229–247. https://doi.org/10.1016/j.ins.2021.02.038

[16] Wang, Z., Cheng, W., & Li, C. (2020). DoS attack detection model of smart grid based on machine learning method. *Proceedings of IEEE International Conference on Power, Intelligent Computing and Systems*, 735–738. https://doi.org/10.1109/ICPICS50287.2020.9202401

[17] Wu, Z., Pan, Q., Yue, M., & Liu, L. (2019). Sequence alignment detection of TCP-targeted synchronous low-rate DoS attacks. *Computer Networks*, *152*, 64–77. https://doi.org/10.1016/j.comnet.2019.01.031

[18] Yue, M., Wang, H., Liu, L., & Wu, Z. (2020). Detecting DoS Attacks Based on Multi-Features in SDN. *IEEE Access*, *8*, 104688–104700. https://doi.org/10.1109/ACCESS.2020.2999668

[19] Zhang, D., Wang, Q. G., Feng, G., Shi, Y., & Vasilakos, A. V. (2021). A survey on attack detection, estimation and control of industrial cyber–physical systems. *ISA Transactions*, *116*, 1–16.

https://doi.org/10.1016/j.isatra.2021.01.036

[20] Zhijun, W., Wenjing, L., Liang, L., & Meng, Y. (2020). Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey. *IEEE Access*, *8*, 43920–43943. https://doi.org/10.1109/ACCESS.2020.2976609