Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 4, August 2026, pp. 3469~3482
ISSN: 2302-9285, DOI: 10.11591/eei.v15i4.11169

Feature selection based SOOA-SDRGRU for DDoS attack detection and classification

Teena Kodapalu Balakrishna, Swati Sharma
Presidency School of Computer Science & Engineering, Presidency University, Bangalore, India

Article Info

Article history:

Received Aug 7, 2025
Revised May 2, 2026
Accepted Jul 9, 2026

Keywords:

Convergence speed
Distributed denial-of-service
Local optima
ShakeDrop refined gated recurrent unit
Somersault Orangutan optimization algorithm

ABSTRACT

Distributed denial-of-service (DDoS) attacks monitor network traffic to determine unusual patterns that represent malicious flooding. However, DDoS attacks mimic legitimate traffic which makes it difficult to accurately detect and classify between normal and attack. This research proposes Somersault Orangutan optimization algorithm for feature selection with ShakeDrop refined gated recurrent unit (SOOA-SDRGRU) to detect and classify DDoS. In OOA, somersault foraging behavior is incorporated to solve local optima issue which enhances solution diversity and rapid convergence speed toward global optima. GRU captures temporal patterns and dependencies in sequential network data, whereas ShakeDrop assist to minimize overfitting, which leads to more accurate and reliable detection of diverse attack patterns. Compared to existing methods like bacterial colony optimization (BCO) with optimized back propagation neural network (BPNN), the proposed SOOA-SDRGRU obtains high accuracy of 0.9989 and 0.9992 on CIC-IDS2017 and CIC-DDoS2019 datasets which shows robust detection method for evolving DDoS patterns.

Corresponding Author:

Teena Kodapalu Balakrishna
Presidency School of Computer Science & Engineering, Presidency University
Rajanakunte, Yelahanka, Bangalore, India
Email: teenakb1@gmail.com

1. INTRODUCTION

Distributed denial-of-service (DDoS) attacks are an important threat to organisations worldwide as overwhelm networks and render services unavailable to legitimate users. These attacks have the potential to paralyse networks that make them inaccessible to users and cause severe disruptions in service integrity and availability. Detecting and mitigating DDoS attacks has become crucial to maintain the resilience and security of critical infrastructure [1]. Attacking devices declare routers, computers, or internet of things (IoT) devices, which are infected with malware attack [2], [3]. DDoS attacks originate from anywhere in the world, making them challenging to identify and difficult to avoid due to their widespread nature [4]. DDoS attack causes important harm like financial losses, reputational harm, even legal consequences [5]-[7]. Conventional security solutions like intrusion detection systems and firewalls based on manually set thresholds that struggle in different attack scenarios and leads to false positives (FP). To address this, researchers introduce methods to detect malicious traffic in DDoS attacks effectively [8]. If an attack targets data plane, it causes an enormous amount of meaningless traffic leading to DDoS, which consumes more processing resources and network bandwidth. Such traffic disrupts normal forwarding of legitimate traffic and forces the software defined networking (SDN) controller to address unwanted flow tables to switches [9], [10]. This not only consumes valuable SDN switch storage but also maximises the overall burden on the data plane. A primary detection technique is commonly categorised as a statistics and deep learning (DL) based method [11]-[13].

Flow modelling is used in statistics-based detection method to evaluate particular network protocols for detecting DDoS [14], [15]. Moreover, these methods are effective, simple, and manage new attack scenarios, which enhances overall security in IoT [16]-[18]. Furthermore, DL adapt to hidden attack patterns from intricate network environments [19], [20]. Arunadevi and Sathya [21] presented an optimised back propagation neural network (BPNN) that employs bacterial colony optimization (BCO) to detect DDoS. However, BCO-BPNN struggled with overfitting due to a tendency to excessively fine-tune network weights, which results in suboptimal performance. Zhao *et al.* [22] established a convolutional neural network-bidirectional long short-term memory (CNN-BiLSTM) to detect DDoS attacks effectively. Nevertheless, CNN-BiLSTM suffer from limited generalization to new and evolving DDoS attack patterns due to dependence on historical data. Hussan *et al.* [23] suggested an optimized Elman recurrent neural network (ERNN) with chaotic bacterial colony optimization (CBCO) for DDoS attack detection. However, CBCO-ERNN face instability in a dynamic DDoS attack environment due to its reliance on chaotic behavior for optimization, which leads to unpredictable results. Roopak *et al.* [24] developed an unsupervised approach for detecting zero-day DDoS attacks in IoT. By using random projection, a zero-day DDoS attack was detected to minimise data dimensionality, and an ensemble model was applied by integrating a Gaussian mixture model (GMM), k-means, and one-class support vector machine (SVM) with a hard voting approach for classification. Tikhe and Patheja [25] introduced a wrapper feature selection-based hybrid DL (WF-HDL) to detect DDoS attacks. Z-score normalization method was used, followed by employing pelican optimization algorithm (POA) to select appropriate features. At last, a DDoS attack was identified accurately by utilising deep auto encoder-convolutional gated recurrent unit (DAE-CGRU). However, DAE-CGRU face challenges in generalizing to DDoS attack variants due to reliance on learned features from training data.

Alshdadi *et al.* [26] established a ResNeSt model with split-attention, which was enhanced by Jaya algorithm (RSG-MJ) and augmented by gated recurrent unit (GRU) to detect DDoS attacks in IoT. Nevertheless, Jaya Algorithm optimization converges prematurely that minimise adaptability in rapidly evolving DDoS attack patterns. Guo *et al.* [27] presented a topological and flow feature-based DL approach (GLD-Net) for detecting DDoS attack. Long short-term memory (LSTM) was associated behind GAT to acquire node neighborhood relationships and fully connected (FC) layer was used for classification. GLD-Net supports to determine distribution of traffic attack source effectively. Shieh *et al.* [28] suggested a Bi-LSTM with GMM for detecting unknown DDoS attacks. Bi-LSTM with GMM effectively distinguish novel instances from samples in trained methods. Ma *et al.* [29] developed a feature and model selection (FAMS) to detect DDoS attacks. Initially, feature coding, duplicate elimination outlier, normalization, and data balancing were used to pre-process the data. At last, linear regression (LR), gradient descent (GD), and SVM act as ensemble method to detect DDoS attack. Wei *et al.* [30] established a auto encoder-multi layer perceptron (AE-MLP) for detecting and classifying DDoS attack. AE was used to extract the features for determining most appropriate feature sets automatically. MLP utilized the compressed and minimized feature sets generated by AE as input and categorize attacks into diverse DDoS attack types.

Aamir and Zaidi [31] implemented a clustering-based ML to classify DDoS attack. Clustering methods like Agglomerative and k-means under principal component analysis (PCA) were used for labeling data to distinguish attacks from normal traffic. After labeling, k-nearest neighbor (KNN), random forest (RF), and SVM were used for classification. Shroff *et al.* [32] suggested a generative adversarial network (GAN) for detecting DDoS attack. GAN generate and categorize synthetic malignant and benign instances which were similar to real instances through similarity scores. This offers towards establishing robust and generic detector for DDoS attack. Kaliyaperumal *et al.* [33] developed a basic autoencoder (BAE)-deep autoencoder (DAE) for binary classification and hierarchical density-based spatial clustering of applications with noise (HDBSCAN) for multi classification of DDoS attacks. The developed method has efficient threat classification ability which attains robust and better performance. Existing DDoS detection methods exhibit research gap which suffer from overfitting, limited generalization to evolving attack patterns, and premature convergence during feature selection that significantly minimize detection ability. Moreover, DDoS attacks mimic legitimate traffic which makes difficult to accurately detect and classify between normal and attack.

Despite the significant progress in DDoS detection using ML and DL techniques, several limitations remain in existing studies. Many existing approaches suffer from premature convergence during feature selection, overfitting and limited generalization to evolving DDoS attack patterns. In addition, several models rely on high-dimensional traffic features which increased computational overhead and reduced detection efficiency in large-scale IoT environments. Therefore, there is a need for an effective framework which simultaneously performs optimal feature selection and robust temporal attack pattern learning to improve detection accuracy, computational efficiency. To address this gap, this research proposes advanced DDoS attack detection and classification using Somersault Orangutan optimization algorithm with ShakeDrop refined gated recurrent unit (SOOA-SDRGRU). SOOA is used to select the most appropriate features for

DDoS by solving the local optima issue, whereas SDRGRU captures temporal dependencies in network traffic data while minimizing overfitting and enhancing robustness.

The main contribution of this research is as follows:

- The manuscript contributes a novel SOOA-SDRGRU by integrating SOOA based feature selection with SDRGRU to ensure accurate and robust DDoS attack detection and classification. This integration enhance generalization, minimize computational overhead through selecting most appropriate features, and enable effective detection of DDoS attack in IoT environment.
- To select the most appropriate features, SOOA is used, which enhances the search for optimal feature subsets by solving the local optima issue. Its effective exploration and exploitation balance assists in capturing different attack patterns more accurately.
- GRU captures temporal dependencies in network traffic, whereas ShakeDrop enhances regularization, which minimises overfitting and enhances generalisation for different attack scenarios. This results in more accurate and robust detection and classification of DDoS attacks.
- Label encoding converts categorical data into numerical form, which enables the model to process and analyze the data effectively, whereas min-max scales features to a consistent range that enhances model convergence from varying feature magnitudes.

This research paper is organised as follows: section 2 illustrates a proposed method. Section 3 details the research method and section 4 analyses results and discussion; and the conclusion of this research paper is given in section 5.

2. PROPOSED METHOD

This research proposes SOOA-SDRGRU to detect and classify DDoS attacks accurately. Initially, CIC-IDS2017 and CIC-DDoS2019 datasets are used to evaluate model performance. Label encoding is used to convert categorical data into numerical data, whereas min-max normalization is employed to transform all numerical data into a fixed range that enhances model convergence and stability. Later, SOOA is performed to select the most appropriate features as well, and SDRGRU is used for DDoS attack detection and classification. Figure 1 demonstrates a detailed process of proposed method for DDoS detection and classification process.

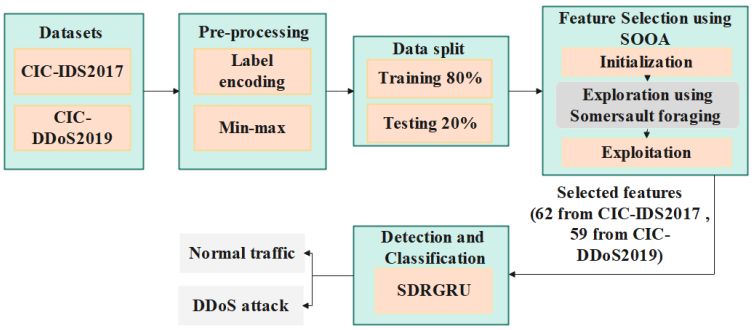


Figure 1. Detailed workflow of proposed method for DDoS detection and classification process

2.1. Datasets

This research employs CIC-IDS2017 [21] and CIC-DDoS2019 [23] datasets to determine model performance in DDoS attacks. These comprehensive datasets provide diverse traffic patterns, which enable a thorough assessment of model performance in identifying different types of DDoS threats. A detailed description of these datasets is explained below.

CIC-IDS2017: the data is obtained over five days in 2017, starting at 9:00 AM on Monday, July 3, 2017, and ending at 5:00 PM on Friday, July 7, 2017, during cyberattacks. Non-attack days that contain both regular and DDoS attack data. It has 2,491,689 samples with both normal and attack data, with 78 network flow features.

CIC-DDoS2019: it is a completely labelled dataset with 50,063,112 instances along with 80 features. It involves normal traffic and various malicious DDoS attack types with 11 class labels to accurately detect different DDoS attacks. The obtained dataset is split into 80% training and 20% testing. Also, k-fold cross-validation is used as k=5 respectively. Table 1 indicates a detailed dataset description for DDoS detection and classification, and these data are passed through the pre-processing stage.

Table 1. Detailed description of the dataset used in this research for detection and classification

Datasets	Instances/samples	Features	Class
CIC-IDS2017	2,491,689	78	8
CIC-DDoS2019	50,063,112	80	11

2.2. Pre-processing

After obtaining data, label encoding and min-max normalization is used in the pre-processing stage. Label encoding is employed to convert categorical features into numerical values, whereas min-max normalization is utilized to scale all feature values into a consistent range. A detailed procedure for these methods is outlined below.

a. Label encoding: the datasets contain categorical data, hence that data is converted into numerical data to make DL model to process and analyze effectively by assigning each unique category a distinct integer. This transformation enables DL model to interpret categorical features as numbers. This process conserves uniqueness of each category without implying any ordinal relationships.
b. Min-max: it is a scaling method that transforms numerical features to a fixed range between zero and one. It is processed by subtracting the minimum feature value and dividing by range, ensuring all features contribute equally during model training, which prevents features with larger scales from dominating and enhances model convergence as well as stability. The mathematical formula for min-max [34] normalization is formulated in (1). Where X_i indicates feature input value, $X_{minimum}$ and $X_{maximum}$ denotes feature's minimum and maximum value. Later, pre-processed input is passed through a feature selection process to select to most appropriate features.

$$X_{i\_normalize} = \frac{X_i - X_{minimum}}{X_{maximum} - X_{minimum}} \quad (1)$$

3. RESEARCH METHOD

3.1. Feature selection

After pre-processing, SOOA select most appropriate features to detect DDoS attack. Feature selection is essential for DDoS attack for identifying most relevant network traffic attributes which minimize noise and irrelevant data. This enhance model's accuracy and efficiency by focusing on significant features that truly indicate attacks. Other optimization algorithms like Gorilla-troop optimization algorithm (GTOA) and chimp optimization algorithm (ChOA), OOA [35] is advantageous, which mimics the adaptive and cooperative foraging behavior of orangutans. This makes effective exploration and exploitation of feature space to identify key features for DDoS attack detection. This ensures that only most relevant and discriminative features are selected which leads to rapid, more accurate, and reliable detection of malicious activity. This improve system's ability to differentiating among normal and abnormal traffic patterns, improving overall network. The following are step-by-step process explanation for the SOOA method.

3.1.1. Initialization

SOOA is a bio-inspired metaheuristic method which draws inspiration from orangutan's natural behavior. Orangutans act as population members and all orangutans indicate a potential solution to optimization issue. As a group, orangutans from the population of SOOA is expressed using matrix form in (2):

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,1} & \cdots & x_{i,1} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,1} & \cdots & x_{N,1} \end{bmatrix}_{N\times m} \quad (2)$$

Initially, each orangutan's position in search space is determined randomly which ensures diverse coverage of the search space and enhances exploration ability. The population's initialization process is represented in (3) where orangutan each dimension position is computed depending on random values within a predefined range. After initialization, the position of all orangutan's corresponds to variable set which are determined by utilizing the objective function, which is formulated in (4). Where X demonstrates SOOA population matrix, X_i indicates i^{th} orangutan (candidate solution), $x_{i,d}$ represents d^{th} dimension in search space, N determines number of orangutans, r provides random number in (0,1) range, m illustrates number of decision variables, lb_d and ub_d shows lower and upper bound of d^{th} decision variable, F represents vector of objective function, and F_i defines computed objective function depending on i^{th} orangutan. Mean

square error (MSE) is used as fitness function due to it measures the predicted and actual class labels that enable precise evaluation of feature subsets. Minimizing MSE guides SOOA toward selecting features which enhance classification accuracy with consistent and stable convergence. Also, population size of 30 search agents and 100 iteration is adopted to balance exploration efficiency and computation overhead.

$$x_{i,d} = lb_d + r.(ub_d - lb_d) \tag{3}$$

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{4}$$

3.1.2. Exploration

In the initial stage of SOOA, each orangutan's position is updated to simulate foraging behavior. For modeling, the movement mathematically, the new position is computed for all orangutans by employing (5). This movement enables orangutans to adjust location which explores fully distant regions that resulting in a significant process in the objective function value. The mathematical formula for the new position is confirmed and updated using (6).

$$x_{i,d}^{P1} = x_{i,d} + r.(CFS_{i,d} - I.x_{i,d}) \tag{5}$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \le F_i \\ X_i, & else \end{cases} \tag{6}$$

Where X_i^{P1} indicates new position of i^{th} orangutan depending on initial stage of SOOA, $x_{i,d}^{P1}$ represents d^{th} dimension, F_i^{P1} demonstrates objective function value, and $CFS_{i,d}$ illustrates d^{th} dimension of chosen food source for i^{th} orangutan. However, traditional OOA gets stuck in local optima due to limited movement diversity in complex search space. To solve this, somersault foraging behavior is used in the exploration stage, which makes abrupt and large-scale movements in the search space to escape local traps and obtain better global performance. This integration enables a more robust selection of features that enhance detection accuracy and reliability. After orangutans determine the location of the local optimum, orangutans perform somersaults using a specific point as pivot and a distance coefficient to jump out of the local optimal region, that enabling them to escape and continue searching for global optimum. The prey is considered as fulcrum in hunting behavior and during each iteration, the position is updated between the present position and the symmetrical counterpoint which is opposite to the fulcrum position. A mathematical formula for somersault foraging behavior is formulated in (7):

$$x_i^d(t+1) = x_i^d(t) + S.\left(r_1.x_{bset}^d(t) - r_2.x_i^d(t)\right), i = 1, \dots, N \tag{7}$$

For the i^{th} orangutan, somersault procedure is explained below:

- In this research, S a somersault factor indicates the position opposite to prey. Here, $S = 2$ is considered, choice of S value is identified by size of solution space of optimization issue. Typically, larger feasible regions of optimization issue, S value is greater, and the selection of S is an empirical value.
- The x_{bset}^d represents prey position, N indicates number of orangutans, d illustrates dimension, and r_1 and r_2 defines random numbers in (0,1) range. In each iteration, the present orangutan $x_i^d(t)$ compares its fitness with orangutans after jump $x_i^d(t+1)$. Whether $f\left(x_i^d(t+1)\right) < f(x_i^d(t))$, somersault is successful, and position information is updated; else, the somersault is not performed.

As the distance between the optimal solution and individual position become smaller, present position fluctuation also decreases, which eventually allowing each orangutan to approach the optimal solution. Hence, the somersault foraging range is reduced adaptively with an increase in number of iterations. In each iteration, the present orangutan compares its fitness with the fitness of orangutan after jumping to the fulcrum. If it falls into a local optimum, it is replaced by individual jumping fulcrum. As the number of iterations increases, probability of replacement becomes higher and the effectiveness of escaping local optimum becomes more obvious, which leads to better model convergence.

3.1.3. Exploitation

During the 2nd stage of SOOA, the orangutan moves nearby tree to nest, which is modelled by making a new position for the orangutan depending on its current location. The (8) simulates the movement

towards trees and if objective function value maximize, new position replaces prior one as defined in (9). Where X_i^{P2} indicates new position of i^{th} orangutan depending on 2nd stage of SOOA, F_i^{P2} demonstrates objective function value, and t demonstrates iteration count. Overall, model enhances global search ability by assisting individuals to jump out of local optima. This increases convergence by dynamically adjusting positions based on the fulcrum. Subsequently, the selected features are fed as input to the detection and classification process.

$$x_{i,j}^{P2} = x_{i,j} + r.\left(1 - 2r_{i,j}\right).\frac{ub_j - lb_j}{t} \tag{8}$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \leq F_i \\ X_i, & else \end{cases} \tag{9}$$

3.2. Detection and classification

After selecting the appropriate features, SDRGRU is used to detect and classify DDoS attacks accurately. GRU [36] is efficient for modelling sequential data by capturing long-term dependencies compared to LSTM, which leads to rapid training, less computational complexity, and reduced memory consumption. Incorporating ShakeDrop enhances the model by dynamically dropping neurons during training, which enhances robustness and generalization. This integration makes the network to better learn complex temporal patterns and minimise the overfitting issue. Unlike LSTM, GRU involves 2 gates, namely update and reset gates. Update gate integrates function of forget and input gate of LSTM which helps model in determining how much information from prior time steps is carried forward in (10). Moreover, the reset gate determines how much past information is discarded as formulated in (11). Unlike LSTM, GRU does not have a separate cell state; instead, it transfers information based on hidden state. Each candidate activation vector is calculated using (12) and internal gate within the GRU cell updates hidden state for each new input at each time t. A final output of GRU cell and next hidden state are acquired by employing the update gate vector in (13).

$$z_t = \sigma(W_z X_t + U_z H_{t-1} + b_z) \tag{10}$$

$$r_t = \sigma(W_r X_t + U_r H_{t-1} + b_r) \tag{11}$$

$$\widehat{h_t} = tanh\ (W_h X_t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{12}$$

$$h_t = z_t \otimes \widehat{h_t} + (1 - z_t) \otimes H_{t-1} \tag{13}$$

Where X_t indicates external input data at time step t, H_{t-1} represents hidden state at prior time step $t-1$, and r_t demonstrates reset gate vector at time step t. The z_t defines output of update gate at time step t, $\widehat{h_t}$ illustrate candidate activation vector at time step t, W_z, W_r, and W_h indicates weight matrix for update gate, reset gate, and candidate hidden state corresponding to input X_t, U_z, U_r, and U_h represents weight matrix for update gate, reset gate, and candidate hidden state corresponding to hidden state H_{t-1}, b_z, b_r, and b_h demonstrates bias vector for update gate, reset gate, and candidate hidden state, $\otimes$ indicates element wise product, and $\sigma(.)$ determines sigmoid activation function. ShakeDrop is integrated into GRU's recurrent output via stochastic residual mapping which follows linear decay schedule from higher dropout at shallow layers to near-zero at deeper ones that ensure stable convergence. This integration enables SDRGRU for retaining significant temporal dependencies and minimizing overfitting, which enhance robustness against diverse and evolving DDoS attack behaviors. ShakeDrop regularization enhances model robustness by dropping layers and neurons during training, which prevents co-adaptation and overfitting. It offers the network to learn more diverse and generalized features that enhance performance on unseen data and stabilise training. The mathematical formula for ShakeDrop is expressed in (14):

$$G(x) = \begin{cases} x + (b_l + \alpha - b_l\alpha)F(x), & in\ train - forward \\ x + (b_l + \beta - b_l\beta)F(x) & in\ train - backward \\ x + E[b_l + \alpha - b_l\alpha]F(x) & in\ test \end{cases} \tag{14}$$

Where b_l determines Bernoulli random variable, which is considered by the linear decay rule in each layer, α and β represents independent uniform random variable. In the training stage, and b_l manage the ShakeDrop behavior. If $b_l = 1$, then (14) is deformed as (15) i.e., ShakeDrop is equivalent to the original network. If $b_l = 0$, then (14) is deformed as (16). Where the calculation of $F(x)$ is perturbed by α and β. The hyperparameter values are selected based on grid search to ensure optimal performance. A learning rate of 0.0001 with Adam optimizer ensure controlled weight updates and rapid convergence. The tanh activation

binary cross-entropy loss function, 100 epoch, and 0.2 dropout enhance temporal feature learning by minimizing overfitting in DDoS detection and classification.

$$G(x) = \begin{cases} x + F(x), & in\ train - forward \\ x + F(x), & in\ train - backward \end{cases} \quad (15)$$

$$G(x) = \begin{cases} x + \alpha F(x), & in\ train - forward \\ x + \beta F(x), & in\ train - backward \end{cases} \quad (16)$$

Thus, the model effectively captures intricate patterns in network traffic while managing robustness against noise. As a result, SDRGRU significantly enhance detection and classification accuracy for DDoS attacks. Overall, it provides an efficient, reliable, and highly accurate solution for securing the network against these attacks. Table 2 shows a notation list description used in the proposed method with corresponding definitions and Algorithm 1 describes the pseudo code of proposed method to ensure reproducibility.

Table 2. Comprehensive list of mathematical symbols and notations employed in proposed method

Symbol	Description
X_i	Feature input value
$X_{minimum}$ and $X_{maximum}$	Feature's minimum and maximum value
X	SOOA population matrix
X_i	i^{th} orangutan (candidate solution)
$x_{i,d}$	d^{th} dimension in search space
N	Number of orangutans
r	Random number in (0,1) range
m	Number of decision variables
lb_d and ub_d	Lower and upper bound of d^{th} decision variable
F	Vector of objective function
F_i	Computed objective function depending on i^{th} orangutan
X_i^{P1}	New position of i^{th} orangutan depending on initial stage of SOOA
$x_{i,d}^{P1}$	d^{th} dimension
F_i^{P1}	objective function value
$CFS_{i,d}$	d^{th} dimension of chosen food source for i^{th} orangutan
S	Somersault factor
x_{bset}^{d}	Prey position
d	Dimension
r_1 and r_2	Random numbers in (0,1) range
X_i^{P2}	New position of i^{th} orangutan depending on 2nd stage of SOOA
F_i^{P2}	Objective function value
t	Iteration counter of algorithm
X_t	External input data at time step t
H_{t-1}	Hidden state at prior time step
r_t	Reset gate vector that after point-wise multiplication with H_{t-1} manage the number of past information to forget
z_t	Output of update gate sigmoid function
$\widehat{h}_t$	Candidate activation vector
Y_t	Output for input data X_t at time step
H_t	Next hidden state value
b_l	Bernoulli random variable which is considered by linear decay rule in each layer
α and β	Independent uniform random variable

Algorithm 1. Pseudo code of proposed method

```
Input: Datasets D
Output: Predicted class labels y

Start
Step 1: dataset and pre-processing
        i) Load datasets D
        ii) Apply label encoding to categorical attributes
        iii) Utilize min-max to numerical features
        iv) Split datasets into 80% training and 20% testing

Step 2: feature selection
        Initialize population of orangutans, set population size=30, iteration=100, and
employ MSE fitness function
        For iteration=1 to T do
                For each orangutan X_i ϵ P do
                        i) Evaluate fitness using MSE
                        ii) Apply somersault foraging behavior in exploration stage
```

```
                    iii) Perform exploitation stage
                    If fitness improves then
                            Update best solution
                    End If
            End For
    End For

Step 3: detection and classification
    Initialize GRU layers with hidden units, tanh activation, ad shakedrop
regularization
    Set learning rate, optimizer, and loss function
    For epoch = 1 to MaxEpochs do
            For each batch B ϵ training data do
                    i) Forward propagate batch through SDRGRU
                    ii) Calculate predicted output
                    iii) Compute loss function and apply ShakeDrop regularization during
                    training
                    iv) Update network weights using Adam optimizer
            End For
    End For

Step 4: inference
    For each sample  x ϵ testing data do
            i) Learn temporal feature sequence
            ii) Compute SDRGRU output probability P(y = 1|x)
            If  P(y = 1|x) ≥ τ  then
                    Assign label=DDoS attack
            Else
                    Assign label=Normal traffic
            End If
    End For

Return: predicted label y
End
```

4. RESULTS AND DISCUSSION

The SOOA-SDRGRU is simulated using Python 3.11 environment with an Intel i5 processor, Windows 11 operating system, 64 GB random access memory (RAM), 1 TB SSD, and 8 GB graphic processing unit (GPU) with TensorFlow and Keras library. Accuracy, recall, F1-score, specificity, and precision are employed to determine model effectiveness using (17) to (21). Where TN demonstrates true negative, FN illustrates false negative, FP represents false positive, and TP determines true positive respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

$$Recall = \frac{TP}{TP+FN} \quad (18)$$

$$F1-Score = \frac{2TP}{2TP+FP+FN} \quad (19)$$

$$Specificty = \frac{TN}{TN+FP} \quad (20)$$

$$Precision = \frac{TP}{TP+FP} \quad (21)$$

4.1. Performance analysis

Table 3 demonstrates performance evaluation of different feature selection. Compared to existing methods like GTOA, ChOA, and OOA, the SOOA obtains high accuracy of 0.9989 and 0.9992 on CIC-IDS2017 and CIC-DDoS2019 datasets due to effectively solving local optima issue. This mechanism enables candidate solutions to make abrupt and dynamic jumps in the search space that avoids premature convergence to local minima. Hence, SOOA refines optimal solutions and fosters cooperative learning, which enhance overall convergence. As a result, SOOA consistently achieves high accuracy compared to existing methods.

Table 4 evaluates the evaluation of different DL techniques. SDRGRU achieves a high accuracy of 0.9989 and 0.9992 on CIC-IDS2017 and CIC-DDoS2019 datasets due to it capture sequential dependencies in network traffic data without suffering from vanishing gradients. Its gating mechanism effectively manages the flow of information that enables it to remember significant patterns related to attack behavior over time.

Moreover, SDRGRU adaptively weight past and current inputs that assist in distinguishing between normal and malicious traffic. By concentrating on appropriate features, SDRGRU enhances detection performance, which leads to accurate detection and classification of DDoS attacks compared to existing methods like ShakeDrop Refined multi layer perceptron (SDRMLP), ShakeDrop refined recurrent neural network (SDRRNN), and ShakeDrop refined long short-term memory (SDRLSTM).

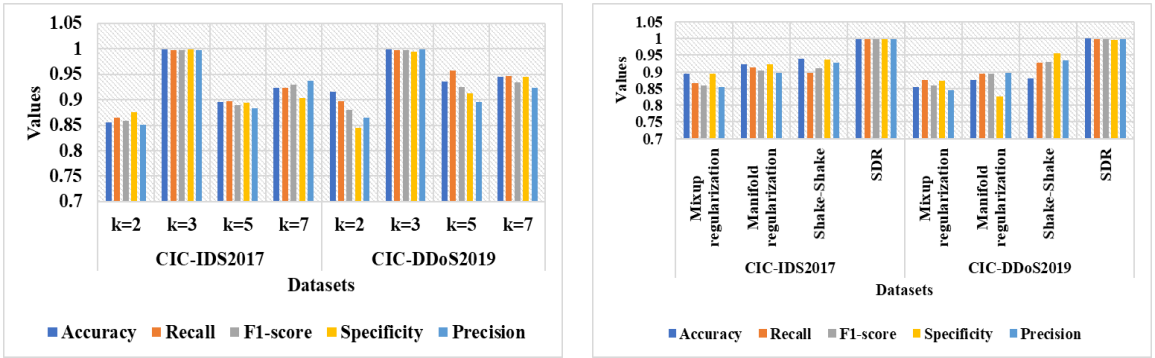
Table 3. Performance of diverse feature selection methods which represents superiority of SOOA in selecting optimal feature subsets for DDoS detection

Methods	Accuracy	Recall	F1-score	Specificity	Precision
CIC-IDS2017					
GTOA	0.8745	0.8547	0.8457	0.8423	0.8369
ChOA	0.8968	0.8965	0.8760	0.8369	0.8565
OOA	0.9245	0.9148	0.8986	0.8945	0.8830
SOOA	0.9989	0.9972	0.9973	0.9981	0.9976
CIC-DDoS2019					
GTOA	0.8471	0.8125	0.8370	0.8594	0.8632
ChOA	0.8965	0.8269	0.8579	0.9054	0.8915
OOA	0.9065	0.8549	0.8783	0.8934	0.9032
SOOA	0.9992	0.9973	0.9978	0.9945	0.9984

Table 4. Performance analysis of different DL methods for DDoS detection and classification which demonstrates effectiveness of proposed SDRGRU

Methods	Accuracy	Recall	F1-score	Specificity	Precision
CIC-IDS2017					
SDRMLP	0.8934	0.8697	0.8576	0.8934	0.8459
SDRRNN	0.9235	0.8932	0.8810	0.9215	0.8692
SDRLSTM	0.9502	0.9257	0.9310	0.9420	0.9365
SDRGRU	0.9989	0.9972	0.9973	0.9981	0.9976
CIC-DDoS2019					
SDRMLP	0.8626	0.8634	0.8794	0.8954	0.8962
SDRRNN	0.8923	0.8936	0.9083	0.9365	0.9236
SDRLSTM	0.9236	0.8257	0.8816	0.9578	0.9457
SDRGRU	0.9992	0.9973	0.9978	0.9945	0.9984

Figure 2 depicts a graphical representation of proposed model for different evaluations. Figure 2(a) represents k-fold validation. Compared to k=2, 3, and 7, k=5 yields a high accuracy due to striking a balance between overfitting and underfitting. Using five neighbors smooths decision boundaries that minimise the impact of noise in data, which ensures predictions are based on a broader and more representative sample of nearby points. Moreover, k=5 obtains better generalization by reducing dependence on irrelevant or distant data points, leading to more accurate and stable classification outcomes. Similarly, Figure 2(b) indicates a graphical representation of different regularisation methods. Compared to existing methods like mixup regularization, manifold regularization, and shake-shake methods, SDR achieves a better accuracy of 0.9989 and 0.9992 on CIC-IDS2017 and CIC-DDoS2019 datasets due to its involvement in random perturbations to branches during training that enhance the model's robustness. These perturbations avoid overfitting by learning a network that provides better generalization performance on unseen data.



(a) (b)

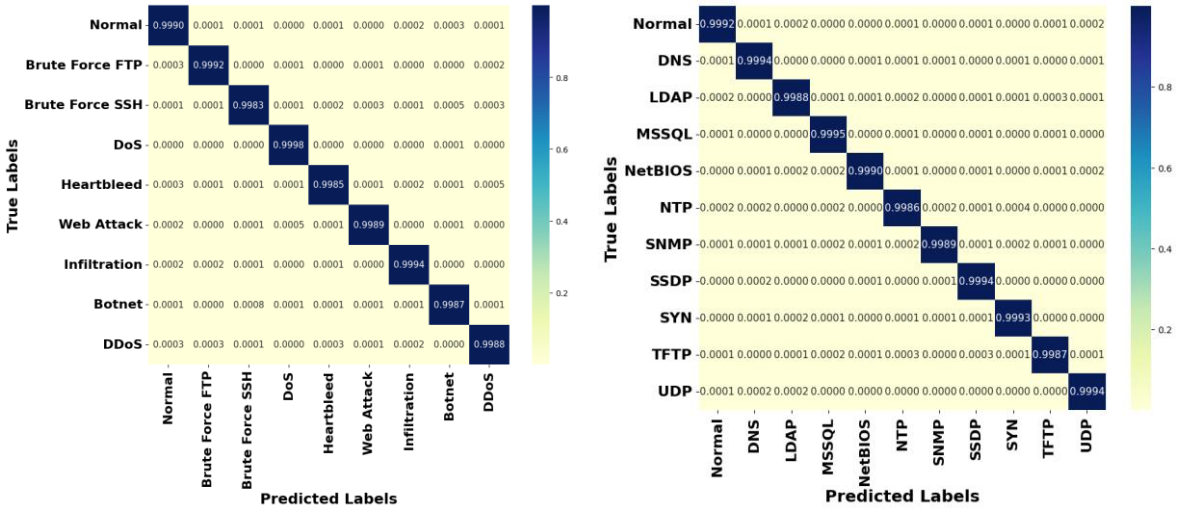
Figure 2. Graphical representation of; (a) k-fold validation process to ensure reliable evaluation and (b) different regularization methods which minimize overfitting compared to traditional methods

Table 5 represents the performance analysis of time complexity per epoch and statistical results. Compared to existing methods, SOOA achieves a shorter training time of 0.1486s and 0.1698s for both datasets due to an effective somersault movement strategy. Unlike traditional methods that require complex calculations, the SOOA involves straightforward updates to candidate solutions, which minimise complexity during each iteration. Moreover, the model's social interactions enhance convergence speed without adding additional complexity. Overall, this ensures in rapid training with fewer computational resources.

Table 5. Performance analysis of time complexity and statistical results which demonstrates efficiency and stability of proposed method

Methods	Training time per epoch (s)	Inference time per epoch (s)	Memory consumption (MB)	t-test
		CIC-IDS2017		
GTOA	0.2598	0.2458	359	0.015
ChOA	0.2469	0.2018	327	0.012
OOA	0.1236	0.0586	304	0.009
SOOA	0.1486	0.0365	263	0.005
		CIC-DDoS2019		
GTOA	0.2548	0.1487	348	0.012
ChOA	0.1965	0.1036	308	0.009
OOA	0.1879	0.0965	298	0.007
SOOA	0.1698	0.0265	214	0.004

Figure 3 shows a performance analysis of the confusion matrix for the classification process (Figure 3(a)) CIC-IDS2017 and CIC-DDoS2019 datasets (Figure 3(b)). In CIC-IDS2017, the model shows the system's accuracy in distinguishing between normal and various attacks, whereas CIC-DDoS2019 represents the system's ability in identifying normal traffic and DDoS attack types. Both confusion matrices demonstrate that the model obtains superior accuracy, with most predicted labels nearly aligning with true labels. Overall, the proposed SDRGRU shows robust performance in classifying network traffic for various attack scenarios.



(a) (b)

Figure 3. Analysis of confusion matrix for proposed SDRGRU; (a) CIC-IDS2017 and (b) CIC-DDoS2019 which shows correct and misclassified instances to evaluate classification reliability

Figure 4 shows an analysis of receiver operating curve (RoC), which trade-off among true positive rate (TPR) and false positive rate (FPR) for each class. From Figure 4, area under curve (AUC) values are near to 1, which represents better classification accuracy for all classes in both datasets. Figure 4(a) CIC-IDS2017 demonstrates the values of AUC ranging from 0.9987 to 0.9999, which shows consistent and high performance over different attacks. Likewise, Figure 4(b) CIC-DDoS2019 obtains around 1.0 AUC which indicates better reliability in classifying different DDoS attack types.

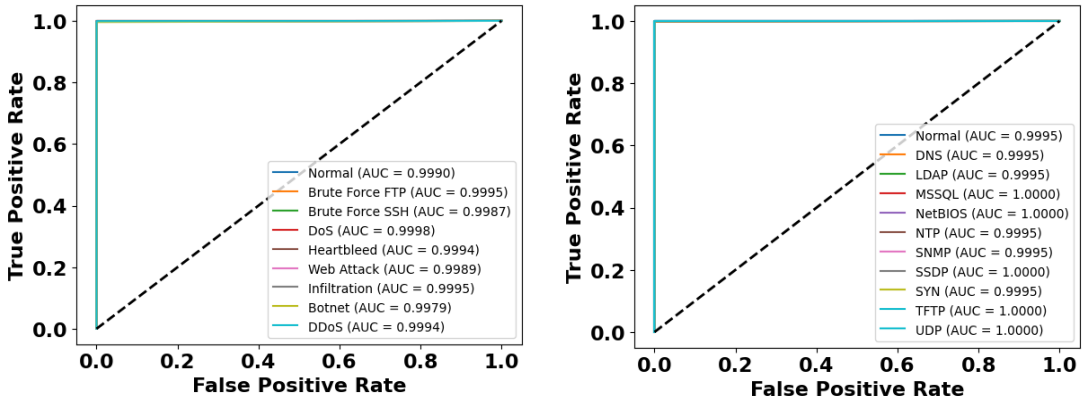


Figure 4. Analysis of AUC curve for proposed SDRGRU; (a) CIC-IDS2017 and (b) CIC-DDoS2019 demonstrating robust discriminative ability

4.2. Comparative analysis

Table 6 illustrates a comparative analysis of existing methods for both datasets. The existing methods, like [21]-[26], [30], [33], [35] are compared with the proposed SOOA-SDRGRU. The existing methods [30], [33], [35] result values are presented in percentage which is converted into decimal to match with proposed method value. Compared to these methods, the proposed method achieves better accuracy of 0.9989 and 0.9992 by leveraging SOOA's strong exploration and exploitation balance. The ShakeDrop method enhances GRU's generalization through regularization. This method minimises overfitting and improves learning stability. Therefore, refined GRU accurately captures temporal dependencies and subtle features in complex patterns.

Table 6. Comparative analysis of existing methods which shows superior performance for DDoS detection and classification

Methods	Datasets	Accuracy	Recall	F1-score	Specificity	Precision
BCO-BPNN [21]	CIC-IDS2017	0.9341	0.9592	0.9684	0.9711	0.9692
CNN-BiLSTM [22]		0.9567	0.9590	0.9586	N/A	0.9582
CBCO-ERNN [23]		0.9849	0.9963	0.9764	0.9947	0.9963
RSG-MJ [26]		0.950	0.960	0.960	0.960	0.930
RF [31]		0.9666	N/A	N/A	N/A	N/A
Proposed SOOA-SDRGRU		0.9989	0.9972	0.9973	0.9981	0.9976
CBCO-ERNN [23]	CIC-DDoS2019	0.9829	0.9957	0.9948	0.9818	0.9857
Ensemble model [24]		0.9450	0.953	0.943	N/A	0.933
DAE-GRU [25]		0.9969	0.9907	0.9905	N/A	0.9903
AE-MLP [30]		0.9834	0.9848	0.9818	N/A	0.9791
bAE+dAE+HDBSCAN [33]		0.9906	N/A	N/A	N/A	N/A
Proposed SOOA-SDRGRU		0.9992	0.9973	0.9978	0.9945	0.9984

4.3. Discussion

The benefits of proposed SOOA-SDRGRU and limitations of existing techniques are represented in detail in this section. The existing method's limitations like BCO-BPNN [21] struggled with overfitting due to the tendency to excessively fine-tune network weights, which results in suboptimal performance. CNN-BiLSTM [22] suffers from limited generalization to new and evolving DDoS attack patterns because of dependence on historical data. CBCO-ERNN [23] faces instability in a dynamic DDoS attack environment due to reliance on chaotic behavior for optimization that leads to unpredictable results. DAE-CGRU [25] struggles to generalise DDoS attack variants due to reliance on features learned from training data. Jaya algorithm [26] optimization converges prematurely, which minimising adaptability in rapidly evolving DDoS attack patterns. The proposed SOOA-SDRGRU overcomes these existing methods' limitations by selecting the most appropriate features and capturing long-term dependencies. A ShakeDrop regularization refines the GRU that provides better generalization by avoiding the co-adaptation of neurons. This method improves model's ability by capturing significant patterns in time-series data. Therefore, the proposed SOOA-SDRGRU offers superior accuracy, enhanced stability, and better management of non-linear dependencies. Overall, the proposed method outperforms the existing methods in detecting and classifying

DDoS attacks. The proposed SOOA-SDRGRU not only demonstrates superior accuracy and stability in detecting and classifying DDoS attack but also obtain significant practical implication. In IoT, which involve highly heterogeneous traffic patterns, model's effective feature selection and ShakeDrop regularization enable robust performance without computational overhead. By enhancing generalization to evolving DDoS attack, proposed SOOA-SDRGRU supports security measures which minimize FP and enhance network performance.

5. CONCLUSION

This research proposed SOOA-SDRGRU to detect and classify DDoS attacks accurately. The primary contribution lies in effective combination of SOOA for feature selection with SDRGRU for DDoS detection and classification. SOOA selects highly suitable features by balancing both the exploration and exploitation processes, which contributes to rapid convergence, solves local optima, and decreases computational complexity. Moreover, GRU is used to manage long-term dependencies in sequential data, while ShakeDrop enhances robustness by preventing overfitting. This ensures robust discrimination among malicious and legitimate traffic when attack patterns nearly resemble normal behavior that makes proposed method well effective for evolving DDoS scenarios. This research represents that proposed SOOA-SDRGRU obtains near-perfect DDoS attack detection performance on CIC-IDS2017 and CIC-DDoS2019 datasets with accuracy of 0.9989 and 0.9992, recall of 0.9972 and 0.9973, and F1-score of 0.9973 and 0.9978 respectively. In addition, the proposed approach reduces computational cost with training time of 0.1486 s and 0.1698 s per epoch while maintaining low memory consumption compared with existing approaches. Moreover, the outcomes confirm proposed method's reliability and efficiency compared to existing methods which shows robust detection of evolving DDoS attacks. Despite these promising results, the study has limitations, where the model is evaluated only on benchmark datasets and does not consider real-time network deployment, adversarial attack scenarios. This limitation may influence the scalability and robustness. Therefore, the future research will focus on real-time DDoS detection in edge-enabled IoT networks and adversarial robustness evaluation to evolving attack patterns and dynamic network traffic conditions, for improved scalability and practical applicability.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Teena Kodapalu Balakrishna	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Swati Sharma		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The datasets generated during the current study are available in the CIC-IDS2017 and CIC-DDoS2019 repository: [21] and [23].

REFERENCES

[1] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Comput. Secur.*, vol. 144, pp. 1-10, Sep. 2024, doi: 10.1016/j.cose.2024.103962.

[2] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (DDOS) attack detection using supervised machine learning algorithms," *Sci. Rep.*, vol. 15, no. 1, pp. 1-14, Apr. 2025, doi: 10.1038/s41598-024-84879-y.

[3] J. Kumar and A. L. P. J. Rose, "A Novel Approach to DDoS Detection in Multi-Controller SDNs: Adaptive Learning Models and Real-Time Feedback for Enhanced IoT Security," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 5, pp. 570–592, Oct. 2024, doi: 10.22266/ijies2024.1031.44.

[4] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2020, doi: 10.1109/ACCESS.2019.2963077.

[5] M. A. Hossain and M. S. Islam, "Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity," *Meas. Sensors*, vol. 32, pp. 1-12, Apr. 2024, doi: 10.1016/j.measen.2024.101037.

[6] S. Balasubramaniam *et al.*, "Optimization Enabled Deep Learning-Based DDoS Attack Detection in Cloud Computing," *Int. J. Intell. Syst.*, no. 1, Jan. 2023, doi: 10.1155/2023/2039217.

[7] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A New Framework for DDoS Attack Detection and Defense in SDN Environment," *IEEE Access*, vol. 8, pp. 161908–161919, 2020, doi: 10.1109/ACCESS.2020.3021435.

[8] B. Wang, Y. Jiang, Y. Liao, and Z. Li, "DDoS-MSCT: A DDoS Attack Detection Method Based on Multiscale Convolution and Transformer," *IET Inf. Secur.*, no. 1, Jan. 2024, doi: 10.1049/2024/1056705.

[9] Z. Liu, Y. Wang, F. Feng, Y. Liu, Z. Li, and Y. Shan, "A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks," *Sensors*, vol. 23, no. 13, pp. 1-24, Jul. 2023, doi: 10.3390/s23136176.

[10] A. T. K. Al-Khayyat and O. N. Ucan, "A Multi-Branched Hybrid Perceptron Network for DDoS Attack Detection Using Dynamic Feature Adaptation and Multi-Instance Learning," *IEEE Access*, vol. 12, pp. 192618–192638, 2024, doi: 10.1109/ACCESS.2024.3508028.

[11] O. Ebrahem, S. Dowaji, and S. Alhammoud, "Towards a minimum universal features set for IoT DDoS attack detection," *J. Big Data*, vol. 12, no. 1, pp. 1-38, Apr. 2025, doi: 10.1186/s40537-025-01146-1.

[12] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, and I. B. Dhaou, "Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model," *IEEE Access*, vol. 11, pp. 119862–119875, 2023, doi: 10.1109/ACCESS.2023.3327620.

[13] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated DDOS attack detection in software defined networking," *J. Netw. Comput. Appl.*, vol. 187, Aug. 2021, doi: 10.1016/j.jnca.2021.103108.

[14] M. J. Pasha, K. P. Rao, A. MallaReddy, and V. Bande, "LRDADF: An AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments," *Meas. Sensors*, vol. 28, pp. 1-11, Aug. 2023, doi: 10.1016/j.measen.2023.100828.

[15] B. Assadhan, A. Bashaiwth, and H. Binsalleeh, "Enhancing Explanation of LSTM-Based DDoS Attack Classification Using SHAP With Pattern Dependency," *IEEE Access*, vol. 12, pp. 90707–90725, 2024, doi: 10.1109/ACCESS.2024.3421299.

[16] S. Selvam and U. M. Balasubramanian, "UASDAC: An Unsupervised Adaptive Scalable DDoS Attack Classification in Large-Scale IoT Network Under Concept Drift," *IEEE Access*, vol. 12, pp. 64701–64716, 2024, doi: 10.1109/ACCESS.2024.3397512.

[17] F. Musumeci, A. C. Fidanci, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks," *J. Netw. Syst. Manag.*, vol. 30, no. 1, pp. 1-27, Jan. 2022, doi: 10.1007/s10922-021-09633-5.

[18] M. Y. Küçükkara, F. Atban, and C. Bayılmış, "Quantum-Neural Network Model for Platform Independent Ddos Attack Classification in Cyber Security," *Adv. Quantum Technol.*, vol. 7, no. 10, Oct. 2024, doi: 10.1002/qute.202400084.

[19] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intell.*, vol. 123, pp. 1-17, Aug. 2023, doi: 10.1016/j.engappai.2023.106432.

[20] A. Pawar and N. Tiwari, "A Novel Approach of DDOS Attack Classification with Optimizing the Ensemble Classifier Using A Hybrid Firefly and Particle Swarm Optimization (HFPSO)," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 4, pp. 201–214, Aug. 2023, doi: 10.22266/ijies2023.0831.17.

[21] M. Arunadevi and V. Sathya, "DDoS Attack Detection using Back Propagation Neural Network Optimized by Bacterial Colony Optimization," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 5, pp. 301–312, Oct. 2023, doi: 10.22266/ijies2023.1031.26.

[22] J. Zhao, Y. Liu, Q. Zhang, and X. Zheng, "CNN-AttBiLSTM Mechanism: A DDoS Attack Detection Method Based on Attention Mechanism and CNN-BiLSTM," *IEEE Access*, vol. 11, pp. 136308–136317, 2023, doi: 10.1109/ACCESS.2023.3334916.

[23] M. I. T. Hussan, G. V. Reddy, P. T. Anitha, A. Kanagaraj, and P. Naresh, "DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization," *Cluster Comput.*, vol. 27, no. 4, pp. 4469–4490, Jul. 2024, doi: 10.1007/s10586-023-04187-4.

[24] M. Roopak, S. Parkinson, G. Y. Tian, Y. Ran, S. Khan, and B. Chandrasekaran, "An unsupervised approach for the detection of zero-day distributed denial of service attacks in Internet of Things networks," *IET Networks*, vol. 13, no. 5–6, pp. 513–527, Sep. 2024, doi: 10.1049/ntw2.12134.

[25] G. N. Tikhe and P. S. Patheja, "A Wrapper Feature Selection Based Hybrid Deep Learning Model for DDoS Detection in a Network with NFV Behaviors," *Wirel. Pers. Commun.*, vol. 133, pp. 481–506, Nov. 2023, doi: 10.1007/s11277-023-10775-9.

[26] A. A. Alshdadi, A. A. Almazroi, E. Alsolami, N. Ayub, and M. D. Lytras, "Enhanced IoT Security for DDOS Attack Detection: Split Attention-Based ResNeXt-GRU Ensembler Approach," *IEEE Access*, vol. 12, pp. 112368–112380, 2024, doi: 10.1109/ACCESS.2024.3443067.

[27] W. Guo, H. Qiu, Z. Liu, J. Zhu, and Q. Wang, "GLD-Net: Deep Learning to Detect DDoS Attack via Topological and Traffic Feature Fusion," *Comput. Intell. Neurosci.*, pp. 1–20, Aug. 2022, doi: 10.1155/2022/4611331.

[28] C.-S. Shieh, W.-W. Lin, T.-T. Nguyen, C.-H. Chen, M.-F. Horng, and D. Miu, "Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model," *Appl. Sci.*, vol. 11, no. 11, pp. 1-13, Jun. 2021, doi: 10.3390/app11115213.

[29] R. Ma, X. Chen, and R. Zhai, "A DDoS Attack Detection Method Based on Natural Selection of Features and Models," *Electronics*, vol. 12, no. 4, pp. 1-32, Feb. 2023, doi: 10.3390/electronics12041059.

[30] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021, doi: 10.1109/ACCESS.2021.3123791.

[31] M. Aamir and S. M. A. Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 4, pp. 436–446, May 2021, doi: 10.1016/j.jksuci.2019.02.003.

[32] J. Shroff, R. Walambe, S. K. Singh, and K. Kotecha, "Enhanced Security Against Volumetric DDoS Attacks Using Adversarial Machine Learning," *Wirel. Commun. Mob. Comput.*, no. 1, Jan. 2022, doi: 10.1155/2022/5757164.

[33] P. Kaliyaperumal, T. Karuppiah, R. Perumal, M. Thirumalaisamy, B. Balusamy, and F. Benedetto, "Enhancing cybersecurity in

Agriculture 4.0: A high-performance hybrid deep learning-based framework for DDoS attack detection," *Comput. Electr. Eng.*, vol. 126, pp. 1-30, Aug. 2025, doi: 10.1016/j.compeleceng.2025.110431.
[34] M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," *Decis. Anal. J.*, vol. 7, Jun. 2023, doi: 10.1016/j.dajour.2023.100233
[35] T. Hamadneh *et al*., "Orangutan Optimization Algorithm: An Innovative Bio-Inspired Metaheuristic Approach for Solving Engineering Optimization Problems," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 1, pp. 47–57, 2025, doi: 10.22266/ijies2025.0229.05.
[36] Y. Imrana, Y. Xiang, L. Ali, A. Noor, K. Sarpong, and M. A. Abdullah, "CNN-GRU-FF: a double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units," *Complex Intell. Syst.*, vol. 10, no. 3, pp. 3353–3370, Jun. 2024, doi: 10.1007/s40747-023-01313-y.

BIOGRAPHIES OF AUTHORS

Teena Kodapalu Balakrishna is working as an Assistant Professor in the Department of Information Science and Engineering at East Point College of Engineering and Technology, Bangalore. She holds an M.Tech. degree in Software Engineering from Sri Jayachamarajendra College of Engineering, Mysuru. Currently, she is pursuing Ph.D. in the area of network security at Presidency University, Bengaluru. She is a dedicated and committed faculty in Department of ISE, with 16 years of experience in teaching UG Students. Her journal publications include: i) "Real Time Precision Farming through Cloud Computer Vision and Deep Learning", International Journal for Research in Engineering Application and Management (IJREAM), pp. 124-127, Special Issue-NGEPT-2019, ISSN:2454-9150; ii) "A Review on Deep Learning Approaches to Real Time Network Intrusion Detection System", Test Engineering and Management, vol. 83, pp. 205–216, 2020, ISSN: 1934120; and iii) "Medical Wing Health Care System", International Journal of Scientific Research in Science, Engineering and Technology, pp. 118-120, vol. 9, no. 4, 2021, ISSN: 2394-4099. Her conference publications include: i) "Block Chain Based File Tracking and Management System for Pension Applications", IEEE International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE-2023); ii) "Detecting Diabetes Mellitus And Nonproliferative Diabetic Retinopathy Using Tongue Features", IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience and Nanotechnology (5NANO2023); and iii) "Bidirectional based LSTM RNN for detecting DDOS flooding attack", publishhed in International Conference ICMTCI-4.0 (2020). She can be contacted at email: teenakb1@gmail.com.

Swati Sharma education and experience: completed B.Tech. from Institute of Engineering and Technology, Bhaddal, in 2002. M.Tech. from Punjab University, Chandigarh in 2005. Ph.D. from SU, Rajasthan, in the year 2011. Professor, Presidency School of Computer Science & Engineering, Presidency University, Bangalore, May 2022-present; Team Lead, Clerisy Solutions Pvt. Ltd., Mohali, April 2020–May 2022; Sr. Faculty and Administration Roles, PTU-affiliated Faculty of Engineering, Punjab, March 2020-July 2005. The publications, webinars, and conferences: i) Springer Nature Singapore Pte Ltd.-Book Series-Lecture Notes in Networks and Systems; ii) National Webinar on Coding with Python-Exploring reasons behind its popularity, MS, India; iii) IEEE International Conference on Communication Systems and Network Technologies, Ahmedabad; and iv) International Conference on High Pressure Science and Technology organized by N.P.L., New Delhi. Honours and awards: she received an award for excellence in education in 2018 by Honourable Governor of Haryana, Indo Global Education Foundation, New Delhi. Projects: i) intelligent finger dexterity based MRS for guitar learning project aims at developing a guitar learning app designed on top of a unique dataset to be deployed on Android and iOS platforms; ii) Python based Greek alphabets tutor developed in Python for engineering domain learning applications, deployed in an academic environment; iii) data encryption based on probabilistic cryptography of alphabets to generate ciphertext developed for research schemes at higher technical education departments of Punjab State; text classification paradox in machine learning by exploring anomalies in NLP project aimed at challenging the trained machine learning model by introducing anomalies such as sarcasm and double negation texts; and iv) machine learning based information extraction and audio classification of speech using Keras aimed at identifying specific expressions out of a dataset by training the neural network model in Keras. The accuracy of the model is optimized by varying model compilation parameters like loss models and optimizers. She can be contacted at email: swati.sharma@presidencyuniversity.in.