International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Levy Flight – Pelican Optimization Algorithm Based Refined Long Short-Term Memory for Network Intrusion Detection System

Valavan Woothukadu Thirumaran*[1], Nalini Joseph[2], Umarani Srikanth[3]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: Intrusion Detection System (IDS) is a method of observing and tracking events on computer systems, which is utilized for identifying signs of security problems and activities monitored by event-based methods and security data. Network IDS (NIDS) is performed intrusion detection by partial packet data of fixed size, but the existing methods suffer to maximize the detection rate and reduce the false alarm rate. In this research, a Levy Flight – Pelican Optimization Algorithm (LF-POA) based Refined Long Short-Term Memory (RLSTM) is proposed for network intrusion detection. The datasets used for evaluating the proposed method are CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT. One-Hot encoding and min-max normalization methods are used as pre-processing techniques, while the feature selection process is performed by POA which is enhanced by Levy flight. The RLSTM method is used for classifying the intrusion as normal or attack. The performance of the proposed technique is analyzed on the basis of accuracy, precision, recall, f1-score, detection rate and false alarm rate. The proposed method attains a detection rate of 99.75%, 95.31%, 98.25% and 93.94% on CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT datasets, respectively. The proposed technique performs better than other existing techniques like Convolutional Neural Network – Long Short-Term Memory (CNN-LSTM) and AdaBoost based method.

Keywords: *Intrusion Detection System, Levy Flight, Pelican Optimization Algorithm, Refined Long Short-Term Memory and Security*

1. Introduction

Intrusion Detection System (IDS) plays an essential part in safeguarding networks from malicious nodes. There are majorly two kinds of IDS namely, Signature-based and Anomaly-based detection [1]. Signature depended detection is performed through extracting a signature of traffic and then is compared with those in the base of pre-built knowledge [2]. As an outcome, it is efficient in detecting known attacks but does not detect attacks out of the knowledge base [3]. Anomaly-based detection identifies the deviations from excessed normal traffic method [4]. Any threat affects the whole network. Encryption and firewall methods are classical security algorithms which face difficulties where attackers keep creating complex attacks [5]. Additionally, cybersecurity identifies the significance of creating effective network IDS (NIDS) to ensure secured networks [6]. IDS provides availability, integrity and confidentiality for data transmission in networked computers through protecting unauthorized access in the network, providing much significant ability for detecting known and unknown attacks and issues with higher accuracy and lesser false alarm rate [7-9].

[1] *Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research (BIHER), Chennai, India*
ORCID ID : 0009-0006-5065-6333
[2] *Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research (BIHER), Chennai, India*
ORCID ID : 0000-0001-6938-9235
[3] *Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India*
ORCID ID : 0000-0002-7359-2495
* *Corresponding Author Email: wtvalavan@gmail.com*

Signature-based detection methods have lesser false alarm rate and higher detection accuracy [10]. With an extension of networks and services, unknown attacks are created through enemies that make the methods vulnerable for these attacks [11,12]. To give security for the networks, intrusion detection should be efficient and intelligent in detecting and protecting unknown and known attacks like anomaly detection [13]. In spite of the huge FAR, anomaly detection identifies the known and unknown attacks [14]. Classification and detection of network traffic in Machine Learning (ML) is dependent on manual extraction feature. While Deep Learning (DL) with the Neural Network (NN) itself extract features after a dataset and performs classification and detection [15]. DL methods maximize and enhance the accuracy of detection when compared to ML. Depending on different techniques and learning approaches, several methods are developed for creating an efficient intrusion detection system [16]. The previous techniques have less precision, less detection and huge false alarm rates. The significant contributions of the research are given below:

- The Levy Flight – Pelican Optimization Algorithm is proposed for feature selection process which selects best features and reduce computational complexity.
- The Refined Long Short-Term Memory (RLSTM) is used for intrusion detection system which classifies the intrusions as normal or attack.
- The proposed technique is analyzed with standard datasets of CIC-IDS 2017, UNSW-NB15, NSL-KDD

and Bot-IoT, which shows high detection rate and low false alarm rate.

The remaining section of manuscript is given in the following format: Section 2 defines the literature survey, section 3 defines the details of the proposed methodology, section 4 discusses the results and discussion, while section 5 presents the conclusion.

2. Literature Review

Halbouni et al. [17] implemented a Convolutional Neural Network – Long Short-Term Memory (CNN-LSTM) for network intrusion detection. CNN has the ability for extracting temporal features for developing hybrid a intrusion detection method. Batch normalization and dropout layers in the method was employed for maximizing its performance. Additionally, depending on binary and multiple class classification, this method was trained on CIC-IDS 2017, UNSW-NB15 and WSN-DS datasets. However, the implemented method was unable to resulted in high Detection Rate (DR) in some kinds of attacks like from worms, backdoors, etc.

Ahmad et al. [18] introduced a AdaBoost based method for NIDS. The feature selection technique was dependent on the correlation matrix measured among features. The features which were hugely correlated with others, also eliminated for important variance and maximized method difficulty. The introduced method depended on decision for classifying normal and probable issues, alongside monitoring network traffic and classifying that to intrusion or non-intrusion. However, the introduced method required numerous amounts of data to acquire better performance.

Park et al. [19] presented a Generative Adversarial Networks (GAN) method as a network IDS. In the presented method, through integrating the state-of-the-art GAN method, it produced plausible synthetic information and calculated the training convergence, representing that the presented method performed well. The presented method outperformed the previous Artificial Intelligence (AI) dependent NIDS with a good detection rate. But, the presented method had overfitting and vanishing gradient problems.

Aliehane et al. [20] suggested a Golden Jackal Optimization Algorithm with Deep Learning Assisted IDS for Network Security (GJOADL-IDSNS) in network IDS. The suggested method soared in effective recognition and classification of intrusions for ensuring network security. Normalization of data was done for scaling input data to a suitable format. The GJOA based feature selection method was assigned for selecting the optimal subset features. Further, for hyperparameter tuning an Attention based Bi-directional LSTM (A-BiLSTM) was deployed. However, the suggested method had lesser efficiency in handling imbalance datasets and the model's learning rate was low.

Osa et al. [21] developed Deep Neural Network (DNN) method for NIDS. The developed method had six hidden layers, introducing Rectified Linear Unit (ReLU) activation function. The output layer of neural network was introduced by the softmax activation more than two classes considered in this method. The Adam optimizer was utilized for optimizing the learning rate when Sparse Categorical cross entropy was utilized for evaluating the developed method. The developed method recognized patterns in network traffic and tracked the process of attack over a certain period, but, the method was computationally complex.

Bijua and Franklin [22] implemented an Evaluated Bird Swarm Optimization based Deep Belief Network (EBSO-DBN). Initially, a relevant format was produced through the preprocessing networking information. Next, the implemented EBSO-DBN method was utilized on alert generation for identification, and also classified intrusions in the IoT environment. However, in implemented method false alarm rate was high.

Han and Pak [23] introduced a Hierarchical LSTM method for network intrusion detection system. Initially, the LSTM method had two LSTM methods and produced features using the packet information. Two LSTMs were created to effectively process the whole packet information through accepting different packet lengths as input and reduced zero-padding which degraded the performance of classification. The introduced method facilitated the development of huge quality packet features through reducing the packet data loss. Nonetheless, the introduced method was sensitive for imbalance data in the network dataset.

3. Proposed Methodology

In this research, the introduced a LF-POA based RLSTM for NIDS. The datasets used for evaluating the proposed method are CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT. One-Hot encoding and min-max normalization methods are used as pre-processing techniques, and the feature selection process is performed by POA which is enhanced by Levy flight. The RLSTM method is used for classifying the intrusion as normal or attack. Fig. 1 represents the overall process of the proposed technique.

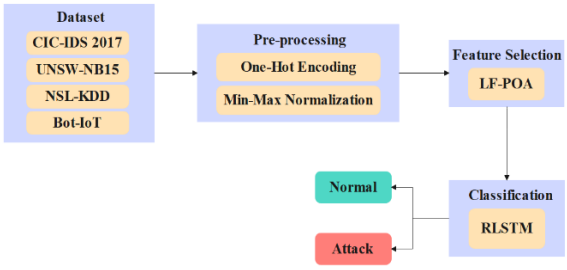

Fig. 1 Overall Process of proposed technique

3.1. Dataset

The datasets used for the intrusion detection are CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT. These are network intrusion datasets which are used for IDS. The description of the datasets is explained below.

3.1.1. CIC-IDS 2017 dataset

The dataset encompasses 11 new attacks which includes brute force, portscan, Dos, and web attacks encompassing XSS and SQL Injection, SSH and FTP-Patator. The dataset is created by Canadian Institute for Cybersecurity, wherein 80 of the 115 features are utilized for monitoring malicious and benign traffic.

3.1.2. UNSW-NB15 dataset

The dataset has files of benign traffic and other kinds of attacks like Backdoor, Fuzzers, DoS exploits. Australian Centre for Cyber Security (ACCS) was developed in 2015. The files are gathered through 3 real-world websites including BID (Symantec Corporation), Common Vulnerabilities Exposures (CVE), and Microsoft Security Bulletin (MSD).

3.1.3. NSL-KDD dataset

This dataset is a polished format of KDDcup 99 dataset that has training and testing datasets such as KDD Train and KDD Test, along 12973 columns and 22544 rows. In every data point, there are 41 features with 3 nominal, 6 binary and 32 numerical features that represent various features of network flow, while the label presents an attack or normal behavior. To attack type, there are 4 distinct attack profiles such as DoS, Remote to Local (R2L), Probing and User to root (U2R).

3.1.4. Bot-IoT dataset

The dataset is a recent intrusion detection dataset in IoT networks, and it is produced by Cyber Range Lab of UNSW Canberra in a real network environment. The botnet and normal traffic are merged for network environment, and the whole dataset has 72 million files. 364,562 files are utilized in training the dataset, while it is reduced to 243,043 files in testing. The dataset includes four types of attacks: Reconnaissance, Distributed DoS, DoS and Theft.

3.2. Pre-processing

The pre-processing is an essential stage which cleans the data, converts the data into numerical format, and also normalizes the data. The pre-processing techniques used in the research are one-hot encoding and min-max normalization.

3.2.1. Data encoding

The stage is used to encode labels in the dataset, the labels in every dataset are not in numerical format so, utilizing One-Hot encoder that encodes a label column through modifying a value from malicious or benign, to their respective numerical values.

3.2.2. Data Normalization

Normalization regulates numerical data in the dataset by using a basic scale without difference in the actual value ranges or data loss. This is done through developing new values which control the resource information proportion and distribution. Normalization is utilized for protecting values of the whole method's numerical columns. Though, Min-Max normalization is deployed in this research for normalizing the data in the dataset. Additionally, the dataset has maximum and minimum values for every characteristic. It is the method specifically used for normalizing data in dataset. The mathematical formula for min-max normalization is given as (1),

$$x = \frac{x_i - min}{max - min} \quad (1)$$

Where, x_i represents the numerical feature of ith sample, min and max represent minimum and maximum values of numerical features. After converting data, it is converted into a numerical format, then the data is normalized from 0 to 1, and given as input to the feature selection process.

3.3. Feature Selection

After preprocessing the data, it is given as input to the feature selection process which selects the relevant features for the process of neural network and reduces the dimensionality. Generally, optimization algorithms are used for feature selection which identify the best features among probable combinations of feature subsets, minimizing computational complexity.

3.3.1. Levy Flight – Pelican Optimization Algorithm (LF-POA)

In this research, Pelican Optimization algorithm (POA) is used which is enhanced by levy flight to improve the ability of global optimization and population diversity of the POA algorithm. POA is a population-based optimization algorithm inspired by pelicans. This is a simulated process of evolution in the ecosystem, wherein the pelicans are considered as individuals in the population. Every specific population describes the possible solution and provides optimization which is obtained by an problem variable for position of every individual in a search space. During population initialization, to ensure the population diversity and ability of global search, every member is initialized randomly in particular upper and lower bounds of issues. This process is mathematically formulated as given in (2).

$$x_{(i,j)} = l_j + rand \cdot (u_j - l_j), i = 1,2,\dots,N, j = 1,2,\dots,m \quad (2)$$

Where, $x_{i,j}$ represents the value of jth variable in the ith candidate solution, N represents all the members. The

amount of issue variables is represented as m, $rand$ represents the random number generated within the range of 0 and 1. l_j and u_j represent the lower and upper bounds of jth issue variable, and is essential to control the range of solution space. The initial locations of population in the search space are distributed regularly, thereby contributing for maximizing the global search ability of the algorithm, along with its search efficiency. The traditional POA initializes the population randomly to minimize diversity of the population, so in this research logistic-sine chaotic map is use to initialize the population. Logistic-sine chaotic mapping integrates features of logistic and sine mapping. The mapping variant by sinusoidal and logical mapping is used because of its high chaotic interval. The mathematical formula of logistic map, sine map and logistic sine map are given in (3) - (5),

$$Z_{i+1} = \mu Z_i(1 - Z_i) \quad (3)$$

$$Z_{i+1} = \sin(\pi Z_i) \quad (4)$$

$$Z_{i+1} = \left(\mu Z_i(1 - Z_i) + \frac{(4-\mu)\sin(\pi Z_i)}{4}\right)(mod1) \quad (5)$$

Where, μ represents chaos multiplier and Z represents the series of numbers developed randomly. The formation of logistic and sine map is given as (6),

$$x_i = l_b + (u_b - l_b) * Z_i \quad (6)$$

The levy flight is a non-Gaussian stochastic procedure, called Levy motion that performs random walks acquired in maintenance of levy. The balance between exploitation and exploration is obtained in accordance with levy flight depending on jumps that allow pelicans clasp many fish in the hunting field. The spreading follows a formula of power law $L(s) \sim |s|^{-1-\beta}$, where $0 < \beta < 2$ describes index and s represents the step length. The numerical representation for step length is given in (7).

$$s = \frac{u}{|v|^{1/\beta}} \quad (7)$$

Where, normal distribution described as source to u and v. The numerical representation is given as (8),

$$uN(0, \sigma_u^2), vN(0, \sigma_v^2) \quad (8)$$

Further, the numerical representation for calculating σ_u is given as (9),

$$\sigma_u = \left\{\frac{\Gamma(1+\beta)\times sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\times\beta\times 2^{\left(\frac{\beta-1}{2}\right)}}\right\}^{\frac{1}{\beta}}, \quad \sigma_v = 1 \quad (9)$$

In POA algorithm, levy function is added when winging. The mathematical formula for new position of pelican is measured by using (10),

$$X_i = \begin{cases} X_i^{P_2} + \alpha \oplus Levy, & if\ F_i^{P_2} < F_i \\ X_i, & else \end{cases} \quad (10)$$

The numerical denotation for calculating α is given as (11),

$$\alpha = 0.01 \times s \times \left(X_i^{P_2} - X_{best}\right) \quad (11)$$

The Logistic-Sine chaotic mapping technique is assigned for maximizing initialization of random solutions that allows for the production of consistently distrbuted and unrepetation intial solution. Levy flight mechanism is used for maximizing the ability of global optimization and for enriching the population diversity of POA algorithm. By using LF-POA, the best features are selected from the dataset, therefore minimizing the computational complexity.

3.4. Long Short-Term Memory

The existing Deep Neural Networks (DNNs) and CNN, LSTM have the advantages of producing results of similar shape, meanwhile input shapes are of different sizes. This is because of the LSTM's extended architecture of Recurrent Neural Network (RNN) produces results of similar dimension for inputs of different sizes. This characteristic of LSTM allows NIDS to separate every packet through a fized size, as well as give every part to every cell of LSTM so that all packets of different sizes are utilized as input for the classifier. Fig. 2 represents the gate architecture of LSTM.

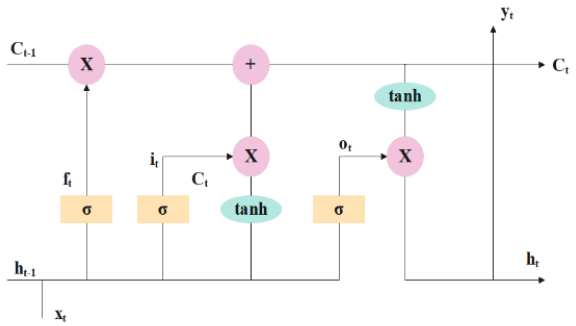

Fig. 2 Gate Architecture of LSTM

3.4.1. Refined – LSTM (RLSTM)

The RLSTM is a higher rated RNN varaiant which has the ability for tackling a issue of long-term dependency of RNN. Additionally, it generates long-term memory and also has the ability of addresing the issue of vanishing gradients which occurs in classical RNN training. It processes the whole sequence of information, rather than individual data points. The RLSTM prevents backpropagated errors from exploiding. The RLSTM has advantages of being relatively insensitive for gap length. RLSTM elements consist of cell, input, output and forget gate. The cell is responsible for remembering the values at an accurate time, while the three gates manage how data enters and cell leaves.

Input gate (i_t) denotes how much input data is required to be stored in cell state at the present moment (t), whereas the intermediate value (u_t) is utilized for updating the cell state process. The numerical formula for input gate is given as

(12) and (13).

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

$$u_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (13)$$

Forget gate (f_t) denotes how many cell states are required to retain from the past moment $(t-1)$ to present moment (t). The mathematical formula for forget gate is given as (14).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \quad (14)$$

Cell state is updated from C_{t-1} to C_t through eliminating certain previous data and updating a filtered intermediate value (u_t). The mathematical formula for cell state is given as (15),

$$C_t = f_t * C_{t-1} + i_t * u_t \quad (15)$$

Output gate (O_t) controls the number of present cell states required for output to the new hidden state. The numerical denotation for output gate is given as (16) and (17),

$$O_t = \sigma(W_O \cdot [h_{t-1}, x_t] + b_O) \quad (16)$$

$$h_t = O_t * \tanh(C_t) \quad (17)$$

In the above (12)-(17), x_t represents the input at t time, C_t and C_{t-1} represents method output at time t and $t-1$, while h_{t-1} and h_t represent the outputs of hidden layer at time t and $t-1$, u_t represents the input of cell state at time t. $f_t, i_t \; and \; O_t$ represent the results of forget, input and output gates at time t. $W_f, W_i, W_o \; and \; W_c$ denote the weights merging $h_{t-1} \; and \; x_t$ for forget, input, output, and cell input, $b_f, b_i, b_O \; and \; b_c$ are its respective bias terms.

3.4.2. Adam Optimizer

Adam Optimizer is a gradient descent technique used in this research as it has the best property of adaptively adjusting the learning rate, generally utilized for measuring weight matrix. Adam optimizer integrates the benefits of RMSProp and AdaGrad optimization algorithms with feasible implementation, huge executional effectiveness and low executing resources. Moreover, the update of parameters in Adam are not affected through transformation of gradient, which is suitable to unstable noise datasets. Overfitting is a general problem in LSTM that results in higher accurcay in training and less accuracy in testing. Nevertheless, it is important for preventing overfitting in training. In this research, a dropout is used for preventting overfitting through dropping certain neurons from network with some possibility in every training. Dropout resolves the overfitting issues through avoiding feature detectors for minimizing difficult relationships among neurons, therefore resulting in the neural network learning good features. After feature selection, it gives the preferable features for neural network for NIDS. The feature selection process minimizes the computational complexity of the model. The RLSTM with adam optimizer detects the intrusion detection in network with high detection rate and less false positive rate.

4. Experimental Analysis

The proposed technique is simulated through python with system requirements of i7 processor and 6 GB RAM. The performance of the suggested technique is analyzed with various performance metrics of accuracy, precision, recall, f1-score, detection rate and false alarm rate. Accuracy is defined as the proportion of true predictions of files. Precision is defined as the ability for avoiding the mislabeling of negative files as positive. Recall is defined as the division of data with positive values that are truly predicted. F1-score is defined as the harmonic average of precision and recall. Detection rate is defined as the ability for predicting positive files in its entirety. False alarm rate is defined as the proportion of misclassification in normal traffic. The mathematical denotations for performance metrics are given from (18) – (23),

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

$$Precision = \frac{TP}{TP+FP} \quad (19)$$

$$Recall = \frac{TP}{TP+FN} \quad (20)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (21)$$

$$Detection\ Rate = \frac{TP}{TP+FN} \quad (22)$$

$$False\ Alarm\ Rate = \frac{FP}{FP+TN} \quad (23)$$

4.1. Quantitative and Qualitative Analysis

The outcomes of the proposed technique is analyzed with performance measures of accuracy, precision, recall, f1-score, DR and FAR. The datasets used for evaluating the proposed technique are CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT. Various tables are described below to show the effectiveness of proposed technique.

Table 1. Performance of Proposed Optimization algorithm

Optimization Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	False Alarm rate (%)
ChOA	92.39	92.02	91.72	91.24	89.19	7.7
ROA	93.71	92.61	92.19	91.82	90.26	7.3
RSA	94.37	93.26	93.01	92.71	90.64	6.9
POA	95.67	94.81	94.16	93.28	91.02	6.4
LF-POA	97.23	96.35	95.76	95.92	91.56	5.2

In table 1, the performance of the introduced optimization algorithm is presented. The existing optimization algorithms used for analyzing the proposed algorithm are Chimp Optimization Algorithm (ChOA), Remora Optimization Algorithm (ROA), Reptile Search Algorithm (RSA) and Pelican Optimization Algorithm (POA). The proposed LF-POA reaches the highest 97.23% of accuracy, 96.35% of precision, 95.76% of recall, 95.92% of f1-score, 91.56% of DR, and 5.2% of FAR, which is more efficient than other optimization algorithms.

Table 2. Performance of RLSTM Neural Network

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	False Alarm rate (%)
DNN	94.64	93.50	92.04	92.82	89.87	7.9
CNN	95.00	94.93	93.51	94.34	90.38	7.0
RNN	96.17	95.27	94.92	95.01	91.42	6.4
LSTM	97.38	96.45	95.27	95.92	92.03	5.9
RLSTM	98.21	97.73	96.62	97.26	92.57	5.1

In table 2, the performance of proposed RLSTM neural network is presented. The existing neural networks used for evaluating the proposed algorithm are Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and LSTM. The proposed RLSTM reaches the highest 98.21% of accuracy, 97.73% of precision, 96.62% of recall, 97.26% of f1-score, 92.57% of DR, and 5.1% of FAR, hence being more efficient than other neural networks.

Table 3. Performance of Proposed technique with CIC-IDS 2017 dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	False Alarm rate (%)
LF-POA based DNN	95.29	95.18	94.38	94.41	94.02	4.5
LF-POA based CNN	96.03	95.82	95.62	95.4	94.92	3.7
LF-POA based RNN	96.77	96.48	96.49	96.38	95.52	2.2
LF-POA based LSTM	97.54	97.59	97.28	97.6	96.48	1.4
LF-POA based RLSTM	99.85	99.77	99.42	99.61	99.75	0.7

In table 3, the results of the proposed LF-POA based RLSTM is described with respect to the CIC-IDS 2017 dataset. The existing techniques used for evaluating the proposed technique are LF-POA based DNN, LF-POA based CNN, LF-POA based RNN and LF-POA based LSTM. The proposed LF-POA based RLSTM reached highest 99.85% of accuracy, 99.77% of precision, 99.42% of recall, 99.61% of f1-score, 99.75% of DR and 0.7% of FAR which is efficient than other existing techniques.

Table 4. Performance of Proposed technique with UNSW-NB15 dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	False Alarm rate (%)
LF-POA based DNN	94.18	93.71	93.27	93.45	92.16	6.9
LF-POA based CNN	94.93	94.27	94.78	95.02	92.73	6.1
LF-POA based RNN	95.52	96.62	96.59	97.67	93.05	5.6
LF-POA based LSTM	97.66	97.35	97.41	98.31	93.67	4.9
LF-POA based RLSTM	99.52	99.41	99.03	99.26	95.31	4.2

In table 4, the performance of proposed LF-POA based RLSTM is described with UNSW-NB15 dataset. The existing techniques used for analyzing the proposed technique are LF-POA based DNN, LF-POA based CNN, LF-POA based RNN and LF-POA based LSTM. The proposed LF-POA based RLSTM reaches the highest 99.52% of accuracy, 99.41% of precision, 99.03% of recall, 99.26% of f1-score, 95.31% of DR and 4.2% of FAR, hence proving to be superior than other existing techniques.

Table 5. Performance of Proposed technique with NSL-KDD dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	False Alarm rate (%)
LF-POA based DNN	96.54	96.03	95.83	95.94	94.18	5.7
LF-POA based CNN	97.02	96.71	96.16	96.48	95.28	5.2
LF-POA based RNN	97.69	97.63	97.04	97.43	96.59	4.8
LF-POA based LSTM	98.18	98.02	97.62	98.28	97.27	4.0
LF-POA based RLSTM	99.13	99.04	98.82	98.91	98.25	3.4

The results of the proposed LF-POA based RLSTM are described with respect to NSL-KDD dataset in table 5. The existing techniques used for evaluating the proposed technique are LF-POA based DNN, LF-POA based CNN, LF-POA based RNN and LF-POA based LSTM. The proposed LF-POA based RLSTM reaches the highest 99.13% of accuracy, 99.04% of precision, 98.82% of recall, 98.91% of f1-score, 98.25% of DR and 3.4% of FAR, therefore outperforming other existing techniques.

Table 6. Performance of Proposed technique with Bot-IoT dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	False Alarm rate (%)
LF-POA based DNN	92.39	92.02	91.72	90.76	89.93	7.0
LF-POA based CNN	93.92	92.56	92.04	91.62	90.55	6.3
LF-POA based RNN	94.83	93.67	93.18	92.83	91.38	5.9
LF-POA based LSTM	95.48	94.29	93.72	94.57	92.47	5.1
LF-POA based RLSTM	97.75	96.49	95.73	96.18	93.94	4.5

The performance of the introduced method is described with respect to Bot-IoT dataset in table 6. The existing techniques used for analyzing the proposed technique are LF-POA based DNN, LF-POA based CNN, LF-POA based RNN and LF-POA based LSTM. The proposed LF-POA based RLSTM reaches the highest 97.75% of accuracy, 96.49% of precision, 95.73% of recall, 96.18% of f1-score, 93.14% of DR and 4.5% of FAR which is higher than other existing techniques.

4.2. Comparative Analysis

The outcomes of proposed LF-POA based RLSTM technique is compared with other existing techniques namely, CNN-LSTM [17], AdaBoost based DT classifier [18], GJOADL-IDSNS [20], DNN [21], EBSO-DBN [22] and Hierarchical LSTM [23] with UNSW-NB15, CIC-IDS 2017 and NSL-KDD datasets. The performance metrics used for comparing the proposed technique are accuracy, precision, recall, f1-score, detection rate and FAR. Table 7 represents the comparative analysis of proposed technique with existing techniques, and showcases its effective performance in contrast to other techniques.

Table 7. Comparative Analysis

Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Rate (%)	FAR (%)
UNSW-NB15	CNN-LSTM [17]	93.75	N/A	N/A	N/A	94.53	6.0
	AdaBoost based DT classifier [18]	99.3	99.7	98.5	99.95	N/A	N/A
	Proposed LF-POA based RLSTM	99.52	99.41	99.03	99.26	95.31	4.2
CIC-IDS 2107	CNN-LSTM [17]	99.64	N/A	N/A	N/A	99.70	0.10
	GJOADL-IDSNS [20]	99.70	98.95	98.95	98.95	N/A	N/A
	DNN [21]	99.68	N/A	N/A	N/A	N/A	N/A
	Hierarchical LSTM [23]	99.61	99.52	99.88	99.70	N/A	N/A
	Proposed LF-POA based RLSTM	99.85	99.77	99.42	99.61	99.75	0.7
NSL-KDD	EBSO-DBN [22]	98.96	99.4	98.87	N/A	N/A	N/A
	Proposed LF-POA based RLSTM	99.13	99.04	98.82	98.91	98.25	3.4

4.3. Discussion

In this section, the advantages of the introduced method and limitations of existing methods are described here. The proposed method is compared with existing methods namely, CNN-LSTM [17], AdaBoost based DT classifier [18], GJOADL-IDSNS [20], DNN [21], EBSO-DBN [22] and Hierarchical LSTM [23]. These methods have limitations of high computational complexity, overfitting and vanishing gradient problem. To overcome these limitations, this research introduces a LF-POA for feature selection process which reduces computational complexity of the model. The RLSTM neural network with adam optimizer is used for intrusion detection which tackles the problem of overfitting and vanishing gradient. The proposed method attains a detection rate of 99.75%, 95.31%, 98.25% and 93.94% for CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT datasets, respectively. The proposed technique performs superiorly in contrast to the other existing methods.

5. Conclusion

In this research, NIDS maximizes the security and identifies the attacks in network for deleting malicious nodes in the network. The existing methods have limitations of high computational complexity, overfitting and vanishing gradient problem. The proposed LF-POA based RLSTM for NIDS is described in this manuscript. The datasets used for evaluating the proposed method are CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT. One-Hot encoding and min-max normalization methods are deployed as pre-processing techniques, while the feature selection process is performed by POA which is enhanced by Levy flight. The RLSTM method is used for classifying the intrusion as normal or attack. The proposed method accomplishes detection rate of 99.75%, 95.31%, 98.25% and 93.94% for CIC-IDS 2017, UNSW-NB15, NSL-KDD and Bot-IoT datasets, respectively. Therefore, the proposed technique is more robust in contrast to the previous techniques: CNN-LSTM and AdaBoost based method. In future, hyperparameter tuning or weight updation of LSTM can be done by

optimization algorithm to further improve the NIDS.

Author contributions

Valavan Woothukadu Thirumaran: Conceptualization, Writing-Original draft preparation, **Nalini Joseph:** Visualization, Investigation, Writing-Reviewing and Editing, **Umarani Srikanth**: Methodology, Software, Field study.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] M. Chalé and N. D. Bastian, "Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems," *Expert Syst. Appl*., vol. 207, p. 117936, Nov. 2022, https://doi.org/10.1016/j.eswa.2022.117936.

[2] N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, and Y. T. Hou, "Manda: On adversarial example detection for network intrusion detection system," *IEEE Trans. Dependable Secure Comput*., vol. 20, no. 2, pp. 1139-1153, Feb. 2023, doi: 10.1109/TDSC.2022.3148990.

[3] J. Mijalkovic and A. Spognardi, "Reducing the False Negative Rate in Deep Learning Based Network Intrusion Detection Systems," *Algorithms*, vol. 15, p. 258, Jul. 2022, https://doi.org/10.3390/a15080258.

[4] S. Shitharth, P. R. Kshirsagar, P. K. Balachandran, K. H. Alyoubi, and A. O. Khadidos, "An innovative perceptual pigeon galvanized optimization (PPGO) based likelihood Naïve Bayes (LNB) classification approach for network intrusion detection system," *IEEE Access*, vol. 10, pp. 46424-46441, May 2022, doi: 10.1109/ACCESS.2022.3171660.

[5] E. U. H. Qazi, M. H. Faheem, and T. Zia, "HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System," *Applied Sciences*, vol. 13, p. 4921, Apr. 2023, https://doi.org/10.3390/app13084921.

[6] M. B. Umair, Z. Iqbal, M. A. Faraz, M. A. Khan, Y. D. Zhang, N. Razmjooy, and S. Kadry, "A network intrusion detection system using hybrid multilayer deep learning model," *Big data*, Jun. 2022, https://doi.org/10.1089/big.2021.0268.

[7] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Comput. Electr. Eng.*, vol. 102, p. 108156, Sep. 2022, https://doi.org/10.1016/j.compeleceng.2022.108156.

[8] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *Int. J. Inf. Secur.*, vol. 22, no. 1, pp. 235-247, Feb. 2023, https://doi.org/10.1007/s10207-022-00634-2.

[9] H. Alazzam, A. Sharieh, and K. E. Sabri, "A lightweight intelligent network intrusion detection system using OCSVM and Pigeon inspired optimizer," *Appl. Intell.*, vol. 52, no. 4, pp. 3527-3544, Apr. 2023, https://doi.org/10.3390/s23084141.

[10] H. Asad and I. Gashi, 'Dynamical analysis of diversity in rule-based open source network intrusion detection systems," *Empirical Software Eng.*, vol. 27, p. 4, Jan. 2022, https://doi.org/10.1007/s10664-021-10046-w.

[11] X. H. Nguyen, X. D. Nguyen, H. H. Huynh, and K. H. Le, "Realguard: A lightweight network intrusion detection system for IoT gateways," *Sensors*, 22(2), p.432, Jan. 2022, https://doi.org/10.3390/s22020432.

[12] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry*, vol. 14, no. 7, p. 1461, Jul. 2022, https://doi.org/10.3390/sym14071461.

[13] C. Zhang, X. Costa-Perez, and P. Patras, "Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms," *IEEE/ACM Trans. Networking*, vol. 30, no. 3, pp. 1294-1311, Jan. 2022, doi: 10.1109/TNET.2021.3137084.

[14] M. A. Haq, M. A. Rahim Khan, and T. AL-Harbi, "Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing," *CMC-Comput. Mater. Continua*, vol. 71, no. 1, pp. 1769-1788, Apr. 2022, https://doi.org/10.32604/cmc.2022.018708.

[15] M. Mehmood, T. Javed, J. Nebhen, S. Abbas, R. Abid, G. R. Bojja, and M. Rizwan, "A hybrid approach for network intrusion detection," *CMC-Comput. Mater. Continua*, vol. 70, no. 1, pp. 91-107, Jan. 2022, https://doi.org/10.32604/cmc.2022.019127.

[16] T. Sommestad, H. Holm, and D. Steinvall, "Variables influencing the effectiveness of signature-based network intrusion detection systems," *Inf. Secur. J.: Global Perspect.*, vol. 31, no. 6, pp. 711-728, Nov. 2022, https://doi.org/10.1080/19393555.2021.1975853.

[17] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837-99849, Sep. 2022, doi: 10.1109/ACCESS.2022.3206425.

[18] I. Ahmad, Q. E. Ul Haq, M. Imran, M. O. Alassafi, and R. A. AlGhamdi, “An efficient network intrusion detection and classification system,” *Mathematics*, vol. 10, no. 3, p. 530, Feb. 2022, https://doi.org/10.3390/math10030530.

[19] C. Park, J. Lee, Y. Kim, J. G. Park, H. Kim, and D. Hong, “An enhanced ai-based network intrusion detection system using generative adversarial networks,” *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2330-2345, Oct. 2022, doi: 10.1109/JIOT.2022.3211346.

[20] N. O. Aljehane, H. A. Mengash, M. M. Eltahir, F. A. Alotaibi, S. S. Aljameel, A. Yafoz, R. Alsini, and M. Assiri, “Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security,” *Alexandria Eng. J.*, vol. 86, pp. 415-424, Jan. 2024, https://doi.org/10.1016/j.aej.2023.11.078.

[21] E. Osa, P. E. Orukpe, and U. Iruansi, “Design and implementation of a deep neural network approach for intrusion detection systems,” *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 7, p. 100434, Mar. 2024, https://doi.org/10.1016/j.prime.2024.100434.

[22] A. Biju and S. W. Franklin, “Evaluated bird swarm optimization based on deep belief network (EBSO-DBN) classification technique for IOT network intrusion detection,” *Automatika*, vol. 65, no. 1, pp. 108-116, Jan. 2024, https://doi.org/10.1080/00051144.2023.2269646.

[23] J. Han and W. Pak, “Hierarchical LSTM-Based Network Intrusion Detection System Using Hybrid Classification,” *Applied Sciences*, vol. 13, no. 5, p. 3089, Feb. 2023, https://doi.org/10.3390/app13053089.

[24] CIC-IDS 2017 dataset: https://www.kaggle.com/datasets/cicdataset/cicids2017/data

[25] UNSW-NB15 dataset: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[26] NSL-KDD dataset: https://www.kaggle.com/datasets/hassan06/nslkdd

[27] Bot-IoT dataset: https://research.unsw.edu.au/projects/bot-iot-dataset