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

Intrusion Detection System using Osprey Optimization Algorithm with Bidirectional Gated Recurrent Unit Technique

Anitha Parabathina*[1], Madhavi Dabbiru[2], Venkata Rao Kasukurthi[3]

Submitted:10/03/2024 **Revised**: 25/04/2024 **Accepted**: 02/05/2024

Abstract: Nowadays, the Intrusion Detection System (IDS) is one of the most important application of security in mobile networks, and it is considered a significant method for exposing attacks and applying security measures to networks. For this reason, various types of IDS approaches have been established in conventional research that focus on recognizing intrusions from datasets with the help of a classification issue. However, conventional techniques are limited in identifying malicious attacks due to the issue of overfitting. To overcome this issue, the Osprey Optimization Algorithm with Bidirectional Gated Recurrent Unit (OOA-BiGRU) is proposed in this research for IDS classification. The OOA selects the set of best features by updating positions based on the chasing and attack behavior. The weights are assigned by a self-attention mechanism which enables the BiGRU to adopt attack patterns and enhance the classification accuracy. Various datasets such as CICIDS-2018, CICIDS-2017, UNSW-NB15 and NSL-KDD are preprocessed by label encoding and min-max normalization to convert categorical feature into integer format and normalize the features. The Synthetic Minority Over-sampling Technique (SMOTE) is the oversampling technique employed for balancing the dataset. The accuracy, precision, recall and f1-score are taken as parameters to estimate the model's performance. The OOA-BIGRU achieves the accuracies of 99.86%, 98.64%, 99.72% and 99.83% respectively on NSL-KDD, UNSW-NB15, CICIDS-2017 and CICIDS-2018 datasets, which is superior when compared to existing methods.

Keywords: *Bidirectional Gated Recurrent Unit, Label Encoding, Intrusion Detection System, Min-max Normalization, Osprey Optimization Algorithm, SMOTE*

1. Introduction

IDS are created to address complexity difficulty, supplying a way of keeping track of network web traffic coupled with recognizing prospective safety and security dangers [1]. As computer system networks continue to expand in complexity and dimension, the need for efficient security and safety measures is increasingly important [2]. IDS is identified as network IDS (NIDS) with host IDS (HIDS); NIDS covers web traffic vulnerable to attack when the HIDS system is hosted on any type of other network equipment [3-4]. These systems examine network web traffic along with recognizing possible safety dangers such as malware, network breaches, and also rejection of solution strikes [5]. However, the increasing sophistication and variety of network website traffic have made it difficult to accurately identify network website traffic using conventional rule-based IDS systems [6]. Machine learning (ML) methods to overcome these constraints have been extensively adopted in IDS for network website traffic types [7]. ML approaches are thoroughly made use of, to develop network breach discovery systems as a result of their capacity to comprehend brand-new breaches [8].

[1] *Department of Computer Science and Engineering, AUTDR Hub, Andhra University, Visakhapatnam, India*
[2] *Department of Computer Science and Engineering, Dr Lankapalli Bullayya College of Engineering, Visakhapatnam, India*
ORCID ID : 0000-0003-1159-7255
[3] *Department of Computer Science and Systems Engineering, College of Engineering (A), Andhra University, Visakhapatnam, India*
** Corresponding Author Email: anitha501p@gmail.com*

ML-based IDS systems offer several advantages over conventional policy-based systems, including higher accuracy and better scalability, along with greater resilience to developing network threats [9]. ML-based IDS is tested by security and safety experts to maintain one of the most current attack methods. It is also prepared due to the complexity of modern computer system networks and the frequent cyber threats [10]. Deep learning (DL) is a sub-field of ML that consists of multiple hidden layers that excel at solving problems with big data [11]. DL has indeed recently advanced intrusion detection, showing detection functions with elevated accuracy rates for intrusion detection [12]. Supervised learning focuses on prediction, while unsupervised learning explores data patterns that rely on training information with correct results [13]. Unsupervised detection is a strategy that relies on unlabeled information, which implies that the design has not previously understood the effects [14]. It is time-consuming and requires a sufficient amount of information to obtain high prediction accuracy [15]. The existing approaches are unable to recognize malicious attacks because of overfitting issues in intrusion detection. To overcome this issue, the OOA with Bi-GRU is proposed for IDS classification in this research. The major contributions are listed as follows;

- The OOA-BIGRU is proposed in this research for classifying IDS and mitigating the overfitting issues.
- The OOA selects a set of the best relevant features by

updating positions based on the chasing and attack behavior.

- The GRU delivers a better balance between efficiency and managing long-term dependencies, and the capability to focus on appropriate data in network traffic, also aiding for reduced overfitting.

The structure of this research is as follows: Section 2 analyzes the literature review, Section 3 expands the proposed method, Section 4 presents the results and discussion, Section 5 presents the current conclusion with future work, finally concluded with references.

2. Literature Review

This section discusses the recent literature review based on Machine Learning (ML) and Deep Learning (DL) techniques for IDS for classifying traffic into multiple classes.

Thirimanne et al. [16] presented a real-time intrusion detection system (RT-IDS) based on a deep neural network (DNN). Data pre-processing steps included feature scaling, categorical data encoding using one-hot encoding, and feature selection excluding 13 content features. The machine learning (ML) pipeline with sequential components was developed for data encoding and scaling before feeding real-time data to the DNN model. The RT-IDS near the gateway router provided network protection for the entire organization, and personal network protection when deployed on a single host. However, the performance of the developed method was different depending on the specific characteristics of the network traffic and the types of intrusions encountered.

Aljehane et al. [17] presented the golden jackal optimization algorithm (GJOA) with DL-assisted IDS for network security (NS). GJOA is utilized for feature selection which simulated the hunting behavior of golden jackals to select an optimal subset of features. The attention-based Bidirectional Long Short Term Memory (Bi-LSTM) model for ID enhanced the system's ability for recognizing and classifying the intrusions efficiently. Automating feature selection and extraction through deep learning reduced the need for manual intervention, thereby improving efficiency in intrusion detection. Nonetheless, the GJOADL-IDSNS technique faced challenges in dealing with highly dynamic and evolving cyber threats due to the static nature of feature selection methods.

Eljialy et al. [18] developed a novel framework for an intrusion detection system using multiple feature selection methods based on DL. The multiple-feature selection procedure was developed followed by classification. The software-defined networking (SDN) dataset was used for training and testing in the developed model. This model applied the multiple-feature selection approach for selecting the high-scoring features from a set of features. The multiple classification algorithms were applied to candidate datasets for building the models. The proposed model also exhibited considerable enhancement in the detection of attacks with high accuracy and low positive rates. Nevertheless, the developed model focused on selecting high-scoring features, but still missed relevant crucial features that affected the overall model's performance.

Devendiran and Turukmane [19] suggested the deep learning-based network intrusion detection system using a chaotic optimization strategy. The data cleansing and M-squared normalization were used for pre-processing the data to make balanced datasets from unbalanced datasets by the Extended Synthetic Sampling Technique. After balancing, features of datasets were taken out by using kernel-assisted principal component analysis, the optimal features were selected by the Chaotic Honey Badger optimization (CHBO) algorithm. After all required features had been extracted, the attacks were classified by the gated attention dual long short-term memory (Dugat-LSTM). However, the developed approach utilized the CHBO for feature selection that attained limitations in computational complexity and scalability, potentially affecting the efficiency and applicability to large-scale networks.

Bakro et al. [20] developed an IG-CS-PSO with an ML classifier in the cloud for IDS. Initially, the pre-processing phase was established by encoding numerical values and numerical scaling data. To identify and categorize various kinds of attacks, the Random Forest (RF) was employed. This approach evaluated a subset of an optimal feature that not only decreased the performance of the approach, but also contained features that highly associated with the data and target variable. However, the developed technique had some significant features that were bypassed, causing overfitting.

Sharma and Singh [21] presented a Feed-Forward Deep Neural Network (FFDNN) approach by utilizing a filter-based feature selection for cloud IDS. The FS goal was to identify and choose the greater appropriate attribute subsets from the score of feature importance for training the presented approach. The FFDNN approach performed effectively without the usage of FS. Still, the training of FFDNN dealt with noisy and high-dimensional features was challenging to interpret due to the class imbalance and difficulty in selecting relevant patterns.

3. Proposed Methodology

In this research, the OOA-BiGRU is proposed for IDS classification, and four datasets are considered to assess the proposed method's performance. Label encoding and min-max normalization are employed for pre-processing to convert the categorical feature into integer format, as well as normalize the features. The OOA is used for selecting the

best set of features by using the chasing and attack behavior. Lastly, the selected features are provided as input to BiGRU, in which the weights are assigned by self-attention mechanism, enabling the BiGRU to adopt attack patterns and enhance the classification accuracy. Fig. 1 describes the workflow of the proposed methodology.

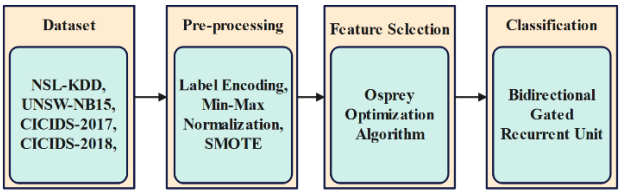

Fig. 1. Workflow of the proposed method

3.1. Dataset Collection

In this research, four diverse datasets: CICIDS-2018, CICIDS-2017, UNSW-NB15 and NSL-KDD are employed for the detection of intrusions. The brief explanation of datasets is described as below:

- **CICIDS-2018:** This dataset includes the network traffic and log files of each machine from the victim side, along with 80 network traffic features extracted from the captured traffic using CICFlowMeter-V3. This dataset also includes 7 diverse attack situations such as DoS, Brute-force, Heartbleed, Botnet, DDoS, Web attacks, and Infiltration [22].
- **CICIDS-2017:** This data includes benign and common attacks in the cyber intrusion field, and contains 78 features belonging to various classes. This dataset contains 9 different classes, brute force FTP, brute force SSH, DoS, heartbleed, web attacks, infiltration, botnet, normal, and DDoS attacks [23].
- **UNSW-NB15:** This dataset is developed by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). It is developed to generate a hybrid of real modern normal activities and synthetic contemporary attack behaviors. It contains 47 features and 10 classes like Shellcode, Exploits, Backdoors, Generic, Fuzzers, Reconnaissance, Analysis, DoS, Worms and normal [24].
- **NSL-KDD:** This dataset is one of the most popular datasets, and is used in the evaluation of IDS frameworks. Also, it is an enhanced version of KDD99 that eliminates the duplicated and redundant data from training and testing sets. It is comprised of 41 features and 5 classes namely, Normal, User to Root (U2R), Denial of Service (DoS), Probe, and Remote to Local (R2L) [25].

3.2. Pre-processing

The collected four datasets are pre-processed over pre-processing approaches such as label encoding, min-max normalization and SMOTE. Label encoding is used to convert the categorical feature into numerical representations, also assigning a unique integer label to every category within in feature. The resulting numeric labels allow learning models to work with categorical data, as most algorithms operate on numerical inputs.

The dataset has features in different scales, for example, one feature ranges from 0-100 and another feature ranges from 0-1000. By applying min-max normalization, all features are converted into a common scale that avoids certain features dominating others because of their large scale [26]. The converted integer features have various scales; hence normalized to a specific range. Here, the min-max normalization is used to normalize feature values in the scale of 0 and 1 which is mathematically given in (1).

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

Where, x_i' is a normalized feature, x_i is an actual feature, x_{min} and x_{max} are the minimum and maximum number of features. After normalization, SMOTE is applied to address the class imbalance. It generates synthetic samples for the minority class by interpolating between existing instances [27]. SMOTE selects a random instance and generates new samples by connecting it with its k nearest neighbors in the feature space. The synthetic samples are then added to the training dataset, effectively increasing the representation of the minority class.

3.3. Feature Selection

The preprocessed functions have some non-important properties such as highly relevant functions, and loud features for the type of IDS to be removed. Here, an optimization formula is used to choose the best attributes among the possible combinations of functional areas minimizing meaningless attributes from preprocessed functions. The osprey's approach to finding fish and bringing the fish to the best place to eat them are all effective, with all-natural actions that form the basis for a brand-new optimization formula. Through optimization, Osprey helps reduce false positive alerts generated by IDS. By fine-tuning detection parameters and thresholds, the algorithm improves the accuracy of intrusion detection while reducing unnecessary alerts that increase security. Finding fish and bringing them to the best place to feed are all intelligent natural processes that form the basis of a brand-new optimization formula. OOA is a population-based technique that provides an optimal solution based on the search power of its population participants in a problem-solving area through an iterative process. The initiation of OOA is explained first, followed by the process of optimizing the organization of ospreys in both stages of travel and exploitation based on the simulation of all-natural osprey habits. At the start of the OOA application, setting

ospreys in the search area automatically use (2).

$$x_{i,j} = lb_j + r_{i,j} \cdot \left(ub_j - lb_j\right), i = 1,2,\dots,N \; and \; j = 1,2,\dots,m \quad (2)$$

Where, X is the populace matrix of osprey's areas, X_i is the i^{th} osprey (a prospect option) x_{ij} is its j^{th} measurement. N is the numerous ospreys, m is the variety of trouble variables, $r_{i.j}$ are arbitrary numbers in the period [0,1]. lb_j, and ub_j are the lower bound and upper bound of the j th problem variable. Because each osprey is a potential problem, each osprey represents an unbiased feature review. For the unbiased aspect of the problem, the reviewed values are implied using a vector according to (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_2) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (3)$$

Where, F is the vector of unbiased feature values and Fi is the unbiased feature value for the i^{th} osprey. The validated values for the unbiased aspect are the primary requirements for evaluating the high quality of prospective services. Consequently, the best value obtained for the unbiased feature represents the best chance solution, and the worst value obtained for the unbiased feature indicates the worst chance service. Considering that the placement of ospreys in the search area has been improved with each version, the more useful opportunity service should be additionally improved in each model.

3.3.1. Phase 1 – Position identification and hunting the fish (exploration)

Initially, fish under the sea is found, thanks to the solid vision of Ospreys who are great hunters. After sensing the location of the fish, they attack and go under the sea in search of the fish. The initial stage of population development in OOA is modeled on the simulation of these natural processes of osprey. Modeling osprey attacks on fish results in substantial changes in osprey composition in the search area, increasing the ability to determine the optimal location of OOA and departure from regional optima. Each osprey is defined using a fish collection (4).

$$FP_i = \{X_k | k \in \{1,2,\dots,N\} \wedge F_k < F_i\} \cup \{X_{best}\} \quad (4)$$

Where, FP_i is the collection of fish locations for the i^{th} osprey, and Xbest is the most effective chance option (the most effective osprey). The osprey instinctively recognizes these fish and attacks them. Based on the simulation of the osprey's activity in the direction of the fish, a new system for the matching osprey is numerically calculated in (5-6). If this location improves the value of the unbiased feature, it replaces the previous location of the osprey according to (7).

$$x_{i,j}^{P_1} = x_{i,j} + r_{i,j} \cdot \left(SF_{i,j} - I_{i,j} \cdot x_{i,j}\right) \quad (5)$$

$$x_{i,j}^{p1} = \begin{cases} x_{i,j}^{p1}, lb_j \le x_{i,j}^{p1} \le ub_j; \\ lb_j, x_{i,j}^{p1} < lb_j; \\ ub_j, x_{i,j}^{p1} > ub_j \end{cases} \quad (6)$$

$$X_i = \begin{cases} X_i^{p_1}, F_i^{P_1} < F_i; \\ X_i, \; else \end{cases} \quad (7)$$

Where, X_i^{P1} is the new position of i^{th} osprey based on initial phase of OOA, $x_{i,j}^{P1}$ is its j^{th} dimension, F_i^{P1} is value of the objective function, SF_i is the selected fish for i^{th} osprey, $SF_{i,j}$ is its j^{th} dimension, $r_{i,j}$ are random numbers in the interval [0,1], and I_{ij} are random numbers from the set {1,2}.

3.3.2. Phase 2 – Carrying the fish to the suitable position (exploitation)

After hunting a fish, the osprey drags it to a suitable (and non-threatening) setting to consume it there. The subsequent stage of population optimization in OOA is designed based on the simulation of these natural habits of the fish. Modeling to bring the fish into a suitable setting results in small changes in osprey placement in the search chamber. This leads to an increase in the exploitative power of OOA in regional search, and is included in the direction of better solutions. In the OOA style, a brand-new spontaneous employment opportunity for each of the people is calculated as an ideal system for consuming fish is mathematically represented in (8-9) to initially adopt these natural habits. Then, if the value of the neutral feature is raised in this brand new system, it replaces Osprey's previous system which fits according to (10).

$$x_{i,j}^{p_1} = x_{i,j} + \frac{lb_j + r \cdot \left(ub_j - lb_j\right)}{t}, i = 1,2,\dots,N, j = 1,2,\dots,m, t = 1,2,\dots,T \quad (8)$$

$$x_{i,j}^{p2} = \begin{cases} x_{i,j}^{p2}, lb_j \le x_{i,j}^{p2} \le ub_j; \\ lb_j, x_{i,j}^{p2} < lb_j; \\ ub_j, x_{i,j}^{p2} > ub_j \end{cases} \quad (9)$$

$$X_i = \begin{cases} X_i^{p_2}, F_i^{P_2} < F_i; \\ X_i, \; else \end{cases} \quad (10)$$

Where, X_i^{P2} is the novel location of i^{th} osprey based on the second phase of OOA, $x_{i,j}^{P2}$ is its j^{th} dimension, F_i^{P2} is its unbiased feature worth r_{ij} are random numbers in time [0,1], t is iteration counter of the algorithm, T is the total number of iterations. The flowchart of OOA is represented in Fig. 2.

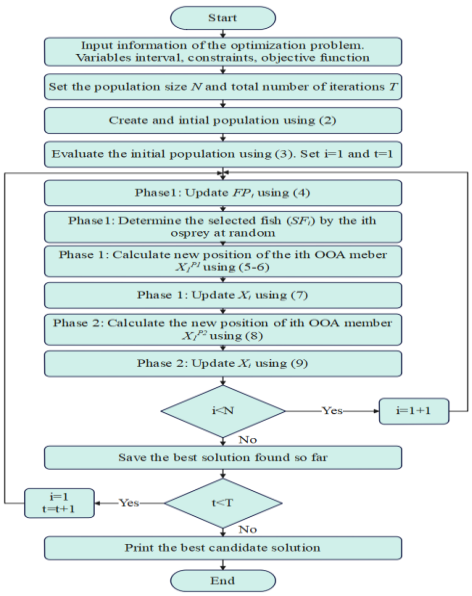

Fig. 2. Flowchart of OOA algorithm

The OOA fitness function is classifier accuracy and the number of selected features. Therefore, the fitness function of the individual solution is mathematically expressed in (11),

$$fitness = \gamma \times ER + (1-\gamma) \times \frac{selected\ features}{Total\ number\ of\ features} \quad (11)$$

Where, the ER is an error rate that denotes classifier error which utilizes selected features and it is estimated as an incorrect classification percentage into the number of classifiers in the range of $[0, 1]$. Further, γ is a weighting factor used for managing the importance of subset length and classifier quality. The selected features are provided as input to the classification of IDS.

3.4. Classification

The selected feature by the OOA algorithm is used for IDS classification by using Bi-GRU which classifies the network traffic into multiple classes. In contrast to long-short-term memory (LSTM), GRU simplifies the security device, reduces network parameters, and is much less likely to generate overfitting. In addition, GRU achieves better results with homogeneous models, and so GRU builds a network architecture. At existing GRU has been extensively utilized. GRU consists of an upgrade gateway, together with a reset entrance which establishes the retention coupled with disposing of information specifically. The GRU memory device incorporates the neglecting entrance f as well as input entrance i from the LSTM to the upgrade gateway z that not only remembers important functions amongst them, but also solves the lengthy dependency problem. However, these facilities are as easy as LSTM. At time n, the specific computational handling to yield the secret layer of the input X_n GRU effect h_n is numerically specified in (12).

$$\begin{aligned} Z_n &= \sigma(W_z \cdot [h_{n-1}, x_n]) \\ r_n &= \sigma(W_r \cdot [h_{n-1}, x_n]) \\ \tilde{h}_n &= \tanh(W \cdot [r_n * h_{n-1}, x_n]) \\ h_n &= (1 - z_n) * h_{n-1} + z_n * \tilde{h}_n \end{aligned} \quad (12)$$

A conventional RNN uses prior information based on a previewed input sequence, but does not consider conforming information. According to the issues of RNN, the BiRNN method remembers the above information and the subsequent information. The output is then combined with a homogeneous resulting layer, and the bidirectional context data is recorded for the feature array. The BiGRU strategy is achieved by trading hidden layer neurons from BiRNN with GRU memory systems. At this time, the BiGRU effect is the hidden layer of h_n, the computation procedure is mathematically expressed in (13-15).

$$\vec{h}_n = \sigma\left(W_{x\vec{h}} x_n + W_{\vec{h}\vec{h}} \overrightarrow{h_{n-1}} + b_{\vec{h}}\right) \quad (13)$$

$$h'_n = \sigma\left(W_{x\overleftarrow{h}} x_n + W_{\overleftarrow{h}\overleftarrow{h}} h'_{n-1} + b_{\overleftarrow{h}}\right) \quad (14)$$

$$h_n = \vec{h} \oplus h'_n \quad (15)$$

Where, W indicates the weight matrix connecting both layers, b stands for the bias vectors, σ as well as $tanh$ refer to the activation feature. $\overrightarrow{h_n}$ and $\overleftarrow{h_n}$ are represented as positive and negative outputs of GRU, while $\oplus$ signifies element-wise. To reduce the overfitting in BiGRU, dropout regularization is effectively applied, and a dropout regularization strategy is often used in neural networks to randomly drop out a fraction in the system to avoid overfitting during training. A hierarchical model is created and a BiGRU layer is added with dropout specified as 0.2 and input dropout and dropout repeated. During training, 20% of the input units and 20% of the iteration units are randomly set to zero at each update to help prevent overfitting.

4. Experimental Results and Discussion

The proposed model is simulated in python 3.8 with the system configurations of intel i7, RAM 16GB and Windows 10 operating system. The assessment parameters like accuracy, precision, recall and f1-score are taken to estimate model performance. The mathematical formula is given in (16-19),

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (16)$$

$$Precision = \frac{TP}{TP+FP} \quad (17)$$

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

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

Where, TP, TN, FP and FN denote True Positive, True Negative, False Positive and False Negative, respectively.

4.1. Performance Analysis

The performance evaluation of OOA-BIGRU is analyzed by NSL-KDD, UNSW-NB15, CICIDS-2017 and CICIDS-2018. In IDS, malicious attack identification is difficult due to overfitting issues. The OOA is used to select a set of best features by updating individual positions based on chasing and attack behavior. The BiGRU produces a balance between efficiency, handling long-term dependencies and the ability to focus on suitable data in network traffic. The proposed OOA-BiGRU approaches attained better performance in every class of each dataset.

Table 1. OOA-BiGRU performance for NSL-KDD dataset

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Probe	99.80	99.90	99.90	99.86
DoS	99.84	99.95	99.94	99.92
U2R	99.87	99.85	99.83	99.80
R2L	99.88	99.82	99.73	99.75
Normal	99.91	99.86	99.84	99.86
Average	99.86	99.87	99.84	99.83

Table 1 displays the OOA-BiGRU performance for the NSL-KDD dataset. The different results in terms of 5 classes as Probe, DoS, U2R, R2L and normal are taken based on metrics like precision, recall and f1-score for analyzing the performance of the proposed method. The average result of OOA-BIGRU accomplishes a precision of 99.87%, accuracy of 99.86%, recall of 99.84%, and f1-score of 99.83%.

Table 2. OOA-BiGRU performance for UNSW-NB15 dataset

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Shellcode	98.41	98.31	97.31	97.31
Exploits	98.56	98.15	98.27	98.27
Backdoors	98.46	98.42	96.41	96.41
Generic	98.53	98.34	97.05	97.05
Fuzzers	98.37	98.19	97.29	97.29
Reconnaissance	98.39	98.54	97.45	97.02
Analysis	98.42	98.51	97.16	97.14
DoS	98.47	98.41	98.25	98.25
Worms	98.64	98.36	97.37	97.37
Normal	98.55	98.55	98.38	98.38
Average	98.48	98.37	97.49	97.44

Table 2 demonstrates the OOA-BiGRU performance for UNSW-NB15 dataset. The different results in terms of 10 classes such as Shellcode, Exploits, Backdoors, Generic, Fuzzers, Reconnaissance, Analysis, DoS, Worms and normal are taken based on metrics like precision, accuracy, recall and f1-score. The average result of OOA-BIGRU accomplishes precision of 99.01%, accuracy of 98.48%, recall of 99.49% and f1-score of 99.44%.

Table 3. OOA-BiGRU performance for CICIDS-2017 dataset

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
BruteForceFTP	99.74	99.74	99.76	99.79
BruteForceSSH	99.45	99.82	99.66	99.66
DoS	99.67	99.67	99.77	99.48
Web Attack	99.59	99.75	99.68	99.79
Infiltration	99.68	99.54	99.65	99.55
Heartbleed	99.45	99.72	99.78	99.76
Botnet	99.67	99.58	99.57	99.79
DDoS	99.78	99.73	99.69	99.65
Normal	99.92	99.89	99.88	99.86
Average	99.66	99.71	99.71	99.7

Table 3 displays the OOA-BIGRU performance for the CICIDS-2017 dataset. The different results in terms of 9 classes of BruteForceFTP, BruteForceSSH, DoS, Web Attack, Infiltration, Heartbleed, Botnet, DDoS and normal are assessed in terms of metrics of precision, recall and f1-score. The average result of OOA-BIGRU accomplishes precision at 99.71%, recall at 99.71%, accuracy at 99.66% and f1-score at 99.70%.

Table 4. OOA-BiGRU performance for CICIDS-2018 dataset

Classes	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
BruteForce	99.83	99.81	99.87	99.62
Infiltration	99.79	99.74	99.74	99.64
Botnet	99.75	99.78	99.76	99.76
Heartbleed	99.64	99.84	99.84	99.74
DoS	99.84	99.79	99.68	99.65
DDoS	99.79	99.83	99.83	99.83
Web Attack	99.64	99.86	99.85	99.65
Normal	99.47	99.96	99.91	99.91
Average	99.71	99.82	99.81	99.72

Table 4 exhibits the OOA-BIGRU performance for the CICIDS-2018 dataset. The different result in terms of 8

classes such as Brute-force, Infiltration, Botnet, Heartbleed, DoS, DDoS, Web attacks and normal are considered with respect to metrics of accuracy, precision, recall and f1-score. The average result of OOA-BIGRU accomplishes precision at 99.82%, accuracy at 99.71%, recall at 99.81% and f1-score at 99.72%. Table 5 displays the OOA-BIGRU performance with state-of-art methods for all datasets such as NSL-KDD, UNSW-NB15, CICIDS-2017 and CICIDS-2018. The RSO with RNN, LSTM, GRU and SAGRU performances are assessed and compared with that of OOA-BIGRU. The OOA-BIGRU accomplishes 99.86% accuracy, 99.84% precision, 99.83% recall and 99.81% f1-score for the NSL-KDD dataset. The OOA-BIGRU accomplishes 98.64% accuracy, 98.37% precision, 97.49% recall, and 97.44% f1-score for UNSW-NB15 dataset. The OOA-BIGRU accomplishes 99.74% accuracy, 99.71% precision, 99.71% recall and 99.70% f1-score for the CICIDS-2017 dataset. The OOA-BIGRU accomplishes 99.83% accuracy, 99.82% precision, 99.81% recall, and 99.72% f1-score for the CICIDS-2018 dataset.

Table 5. OOA-BiGRU performance with state-of-art methods

Dataset	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NSL-KDD	RNN	92.65	92.61	92.54	92.65
	LSTM	94.51	94.32	94.64	94.25
	GRU	96.54	96.74	96.81	96.53
	BiGRU	97.89	97.41	97.43	97.35
	OOA-BiGRU	99.9	99.86	99.84	99.82
UNSW-NB15	RNN	92.94	92.84	92.67	92.59
	LSTM	94.25	94.23	94.24	94.19
	GRU	95.44	95.65	95.34	95.51
	BiGRU	97.64	97.67	97.59	97.32
	OOA-BiGRU	98.71	98.65	98.42	98.37
CICIDS-2017	RNN	93.84	93.75	93.51	93.45
	LSTM	95.89	95.87	95.74	95.65
	GRU	97.53	97.49	97.45	97.32
	BiGRU	98.34	98.32	98.36	98.27
	OOA-BiGRU	99.85	99.74	99.72	99.82
CICIDS-2018	RNN	93.45	93.41	93.36	93.41
	LSTM	95.51	95.45	95.41	95.34
	GRU	96.74	96.41	96.34	96.26
	BiGRU	98.68	98.61	98.53	98.41
	OOA-BiGRU	99.94	99.85	99.83	99.86

4.2. Comparative Analysis

The proposed model is analyzed comparatively with the existing techniques such as RT-IDS-DNN [16], GJOADL-IDSNS [17], CHBO-LSTM [19], and FFDNN [21]. The accuracy, precision, recall and f1-score are taken as the evaluation parameters used to estimate the model's performance. Table 6 indicates the comparative analysis for NSL-KDD, UNSW-NB15, CICIDS-2017, and CICIDS-2018 datasets.

Table 6. Comparative Analysis

Dataset	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NSL-KDD	RT-IDS-DNN [16]	97.1	96	99.1	97.6
	GJOADL-IDSNS [17]	99.81	99.76	99.81	99.79
	CHBO-LSTM [19]	97.5	95.7	99	96.7
	FFDNN [21]	90.88	93.58	96.54	95.04
	OOA-BiGRU	99.86	99.84	99.83	99.81
UNSW-NB15	RT-IDS-DNN [16]	86.9	88.1	88.1	88.1
	GJOADL-IDSNS [17]	85.55	86.24	85.55	85.61
	OOA-BiGRU	98.64	98.37	97.49	97.44
CICIDS-2017	GJOADL-IDSNS [17]	99.7	99.68	99.7	99.69

	CHBO-LSTM [19]	99.63	99.64	99.63	99.63
	OOA-BiGRU	99.72	99.71	99.71	99.7
	CHBO-LSTM [19]	99.78	99.77	99.78	99.7
CICIDS-2018	FFDNN [21]	83.57	89.87	92.57	91.2
	OOA-BiGRU	99.83	99.82	99.81	99.72

4.3. Discussion

The existing IDS techniques such as RT-IDS-DNN [16] are differed depending on the specific characteristics of the network traffic and the types of intrusions encountered. The GJOADL-IDSNS [17] face challenges in dealing with highly dynamic and evolving cyber threats due to the static nature of feature selection methods. CHBO-LSTM [19] attain limitations in computational complexity and scalability that potentially also affect the efficiency and applicability of large-scale networks. FFDNN [21] faces noisy and high-dimensional features, making it challenging to interpret due to the class imbalance and difficulty in selecting relevant patterns. To overcome these drawbacks, OOA-BIGRU is proposed in this research to provide balance among efficacy, handling long-term dependencies and ability to focus on appropriate features to reduce overfitting.

5. Conclusion

In this research, the OOA-BiGRU is proposed for classifying IDS and mitigating the class imbalance issues. The OOA selects a set of best relevant features by updating its positions based on its chasing and attack behavior. The GRU offers a preferable balance between efficacy, managing long-term dependencies, and the ability to focus on appropriate data in network traffic. Less attention is provided when the attack traffic is based on the majority class, and as the high attention is provided when the minority class is attained. In BiGRU, regularization is assigned by dropout which allows to adopt of attack patterns and enhances the classification accuracy. The OOA-BiGRU accomplishes 99.86%, 98.64%, 99.72% and 99.83% for NSL-KDD, UNSW-NB15, CICIDS-2017 and CICIDS-2018 datasets, correspondingly. In the future, the improved or hybrid optimization algorithm can be employed in IDS to improve the feature selection to improve detection accuracy.

Author contributions

Anitha Parabathina: Conceptualization, Methodology, Software, Field study **Madhavi Dabbiru:** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Venkata Rao Kasukurthi:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] V. Hnamte and J. Hussain, "DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system," *Telematics and Informatics Reports*, vol. 10, p. 100053, Jun. 2023, https://doi.org/10.1016/j.teler.2023.100053.

[2] B. Deore and S. Bhosale, "Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection," *IEEE Access*, vol. 10, pp. 65611-65622, Jun. 2022, doi: 10.1109/ACCESS.2022.3183213.

[3] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *The Journal of Supercomputing*, vol. 79, no. 12, pp. 13241-13261, Aug. 2023, https://doi.org/10.1007/s11227-023-05197-0.

[4] 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.

[5] S. Yaras and M. Dener, "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm," *Electronics*, vol. 13, no. 6, p. 1053, Mar. 2024, https://doi.org/10.3390/electronics13061053.

[6] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE," *IEEE Access*, vol. 11, pp. 37131-37148, Apr. 2023, doi: 10.1109/ACCESS.2023.3266979.

[7] A. Paya, S. Arroni, V. García-Díaz, and A. Gómez, "Apollon: A robust defense system against Adversarial Machine Learning attacks in Intrusion Detection Systems," *Comput. Secur.*, vol. 136, p. 103546, Jan. 2024, https://doi.org/10.1016/j.cose.2023.103546.

[8] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, p. 29, Mar. 2023, https://doi.org/10.3390/jsan12020029.

[9] A. Alsarhan, M. Alauthman, E. Alshdaifat, A. R. Al-Ghuwairi, and A. Al-Dubai, "Machine Learning-driven optimization for SVM-based intrusion

detection system in vehicular ad hoc networks," *J. Ambient Intell. Hum. Comput.*, vol. 14, no. 5, pp. 6113-6122, May 2023, https://doi.org/10.1007/s12652-021-02963-x.

[10] R. B. Said, Z. Sabir, and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software Defined Networking with Hybrid Feature Selection," *IEEE Access*, vol. 11, pp. 138732-138747, Dec. 2023, doi: 10.1109/ACCESS.2023.3340142.

[11] J. F. C. Garcia and G. E. T. Blandon, "A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks," *IEEE Access*, vol. 10, pp. 83043-83060, Aug. 2022, doi: 10.1109/ACCESS.2022.3196642.

[12] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, p. 100306, Sep. 2023, https://doi.org/10.1016/j.array.2023.100306.

[13] V. Gowdhaman and R. Dhanapal, "An intrusion detection system for wireless sensor networks using deep neural network," *Soft Comput.*, vol. 26, no. 23, pp. 13059-13067, Dec. 2022, https://doi.org/10.1007/s00500-021-06473-y.

[14] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Comput. Commun.*, vol. 199, pp. 113-125, Feb. 2023, https://doi.org/10.1016/j.comcom.2022.12.010.

[15] N. Yadav, S. Pande, A. Khamparia, and D. Gupta, "Intrusion detection system on IoT with 5G network using deep learning," Wireless Communications and Mobile Computing, vol. 2022, p. 9304689, 2022, DOI: 10.1155/2022/9304689.

[16] S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep neural network based real-time intrusion detection system," *SN Comput. Sci.*, vol. 3, no. 2, p. 145, Jan. 2022, https://doi.org/10.1007/s42979-022-01031-1.

[17] 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.

[18] A. E. M. Eljialy, M. Y. Uddin, and S. Ahmad, "Novel Framework for an Intrusion Detection System Using Multiple Feature Selection Methods Based on Deep Learning," *Tsinghua Sci. Technol.*, vol. 29, no. 4, pp. 948-958, Aug. 2024, doi: 10.26599/TST.2023.9010032.

[19] R. Devendiran and A.V. Turukmane, "Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy," *Expert Syst. Appl.*, vol. 245, p. 123027, Jul. 2024, https://doi.org/10.1016/j.eswa.2023.123027.

[20] M. Bakro, R. R. Kumar, A. Alabrah, Z. Ashraf, M. N. Ahmed, M. Shameem, and A. Abdelsalam, "An Improved Design for a Cloud Intrusion Detection System Using Hybrid Features Selection Approach with ML Classifier," *IEEE Access*, vol. 11, pp. 64228-64247, Jun. 2023, doi: 10.1109/ACCESS.2023.3289405.

[21] H. S. Sharma and K. J. Singh, "A feed forward deep neural network model using feature selection for cloud intrusion detection system," *Concurrency Comput. Pract. Exper.*, vol. 36, no. 9, p. e8001, Apr. 2024, DOI: 10.1002/cpe.8001.

[22] CICIDS-2018 dataset link: https://registry.opendata.aws/cse-cic-ids2018/.

[23] CICIDS-2017 dataset link: https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset.

[24] UNSW-NB15 dataset link: https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15.

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

[26] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 9395-9409, Dec. 2022, https://doi.org/10.1016/j.aej.2022.02.063.

[27] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm," *EURASIP J. Adv. Signal Process.*, vol. 2022, p. 39, May 2022, https://doi.org/10.1186/s13634-022-00871-6.