International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I10P118

Volume 72 Issue 10, 180-190, October 2024

Original Article

Enhancing Intrusion Detection System Using Osprey Optimization Algorithm with Ensemble Learning Model

Swapna Sunkara[1], T. Suresh[2], V. Sathiyasuntharam[3]

[1,2]*Department of Computer Science and Engineering, Annamalai University, Chidambaram.*
[3]*CSE - Cyber Security, CMR Engineering College, Hyderabad.*

[1]*Corresponding Author : swapna.sun@gmail.com*

Received: 18 May 2024 Revised: 27 September 2024 Accepted: 08 October 2024 Published: 25 October 2024

Abstract *- Networks play crucial roles in the modern world, and cybersecurity has evolved into a vital research field. An Intrusion Detection System (IDS) is a significant cybersecurity system that monitors the status of hardware and software operating within the network. Even after decades of development, current IDSs still encounter problems enhancing recognition accuracy, decreasing the False Alarm Rate (FAR), and identifying unknown attacks. To resolve the above challenges, several researchers have concentrated on emerging IDSs that exploit Machine Learning (ML) approaches. ML techniques automatically learn the crucial differences between normal and abnormal data with maximum accuracy. Moreover, ML approaches have strong generalizability, enabling them to effectively identify unknown or novel attacks, further bolstering their capabilities in cybersecurity. Deep Learning (DL), a subcategory of ML that utilizes Neural Networks (NNs), is gaining attention for its outstanding performance. This study introduces an Enhanced IDS using the Osprey Optimization Algorithm with Ensemble Learning (EIDS-OSOAEL) model. The EIDS-OSOAEL technique mainly focuses on the design of an ensemble classifier that integrates the output from multiple classes. Primarily, the EIDS-OSOAEL technique involves a min-max scalar for scaling the input data into a uniform format. Besides, the Tandem-Twirl Modified Bacterial Foraging Optimization Algorithm (TT-MBFOA) approach is employed to achieve the optimal Feature Selection (FS). For intrusion detection, the EIDS-OSOAEL approach undergoes an ensemble of three classifiers, namely AutoEncoder (AE), Feed-Forward NN (FFNN), and Elman NN (ENN). The OS-OA approach is utilized to adjust the hyperparameter values of these models. The experimental results of the EIDS-OSOAEL technique are evaluated under a standard dataset. The performance validation of the EIDS-OSOAEL technique showed a superior 99.76% over recent approaches.*

Keywords *- Intrusion detection system, Osprey optimization algorithm, Network security, Ensemble learning, Feature selection.*

1. Introduction

The world is advancing in computer technology, and as a result, the modern era offers the Internet of Things (IoTs) and networking technology for everyday usage [1]. Therefore, the networking infrastructure stores an abundance of government information and military, commercial, and personal data [2]. Network security is considered a significant concern in internet applications since intellectual properties that are stored easily on the net are copied from the internet. Antivirus firewalls and IDSs are some solutions for safeguarding network environments [3]. Intrusion detection is more widespread to provide security for social networks and networking systems. By deploying a strategy on intrusion detection-based network security, the user can protect their system from intruders [4]. As an active defence technology, Network IDS (NIDS) has become a crucial area of research [5]. It mainly analyses and detects information in the audit files, network system and related logs to define whether the behaviour breaches computer system security and security policies [6]. NIDS is a classification problem used to resolve model optimization, data dimensionality reduction, and classifier construction to improve detection efficiency. The objective of FS is to choose the optimum subset and decrease the data dimensionality to enhance the intrusion detection performance [7]. ML-based IDSs can obtain better detection accuracy once the training dataset is sufficiently accessible, and the ML model has a better generalization ability to identify novel attacks and attack variants. Moreover, ML-based IDSs do not heavily rely on domain information; thus, it is easy to construct and design [8]. DL is a subdivision of ML techniques that can attain remarkable success. DL techniques are better at dealing with big data than classical ML methods. Furthermore, DL techniques can learn representative features automatically from raw information and later output the results; they are practical and work end-to-end [9]. As technology grows and interconnected devices proliferate, the risk of cyber threats escalates, emphasizing the urgent requirement for robust safety measures. Protecting sensitive

data in this digital landscape is paramount, necessitating innovative solutions that efficiently detect and respond to intrusions. By employing advanced optimization methods and ensemble learning, the efficacy of IDSs is enhanced, confirming a safer network environment [10]. This study introduces an Enhanced IDS using the Osprey Optimization Algorithm with Ensemble Learning (EIDS-OSOAEL) model. Primarily, the EIDS-OSOAEL technique comprises a min-max scalar for scaling the input data into a uniform format. Besides, the Tandem-Twirl Modified Bacterial Foraging Optimization Algorithm (TT-MBFOA) technique is employed for the optimum choice of features. For intrusion detection, the EIDS-OSOAEL technique undergoes an ensemble of three classifiers, namely AutoEncoder (AE), Feed-Forward NN (FFNN), and Elman NN (ENN). The OS-OA approach is utilized to adjust the hyperparameter values of these models. The experimental results of the EIDS-OSOAEL approach are evaluated under a standard dataset.

2. Related Works

Al Essa and Bhaya [11] develop a novel IDS trust on fusion ensemble and FS classifiers. A fusion FS technique contains dual models, such as hard voting and mean, and also uses three dissimilar FS methods. Next, a hard-voting model and a mean method are employed. In [12], a hybrid ensemble model utilizing Bagging and AdaBoosting for IDS is projected, which contains 3 phases. The first phase is pre-processing. The 2nd phase includes using Bagging and AdaBoosting models by four diverse classifiers: SVM, Naïve Bayesian (NB), KNN, and RF. Then, the AdaBoosting classification model is united to work in the Bagging model. At last, the voting method is applied. Mogollón-Gutiérrez et al. [13] developed a network traffic identification system for dissimilar classes dependent upon numerous AI models. In the initial task, binary techniques were used to discriminate clearly between every kind of traffic. An ensemble method has been projected in dual stages, which permits the separation of illegitimate and legitimate traffic (stage 1) and classifies the kind of illegitimate traffic (stage 2). In [14], a novel structure termed BoostIDS is intended to influence ensemble learning. BoostIDS includes dual significant modules: (i) A data observing and FS module that uses an effectual Boosting FS method to pick the finest SG-based feature and (ii) An EL-based attacks recognition method that executes a Lightweight Boosting Algorithm (LBA) to determine SG-assisted threats.

Saheed and Misra [15] present an innovative EL model based on a Grey Wolf Optimizer (GWO). The method uses a voting method and a hybrid of FS and feature extraction models. Then, GWO was applied to enhance the parameters of EL methods. The hybrid of Principal Component Analysis (PCA) and Information Gain (IG) was used to reduce dimensionality. The research employed an innovative GWO-EL technique that combined a decision tree, RF, KNN, and multi-layer perceptron for identification. Le et al. [16] presented an ML-based IDS model with DT and RF classifiers dependent upon the ensemble trees method. Besides, dual massive datasets are employed for the experimentation estimation of the projected technique over the feature set. Moreover, the SHAP is used in the XAI approach to clarify and take the classification results of RF and DT techniques. Yao et al. [17] present an NIDS utilizing a One-Class BiGRU-AE and EL approach. Initially, a One Class Classification method dependent upon a BiGRU-AE model has been given.

Then, a multi-classification detection model based on EL has been projected. Kumar et al. [18] examine the Chebyshev Osprey Optimization-based LSTM (ChOs_LSTM) for intrusion detection. The ChOs model fine-tunes LSTM parameters, integrating Chaotic Chebyshev mapping to improve randomness and avoid local optima. Alghanam et al. [19] present an improved Pigeon-Inspired Optimization (PIO) methodology, integrating a Local Search Algorithm (LS-PIO) model. Furthermore, it utilizes an ensemble learning method with diverse 1C classifiers to improve the accomplishment. Yao et al. [20] propose a novel leukaemia diagnosis methodology utilizing an optimized Capsule NN (CapsNet). Kagade and Vijayaraj [21] introduce a Clustering-based IDS (CIDS) for Wireless Sensor Networks (WSNs). It employs a hybrid approach, OCCOA, for optimal selection of Cluster Head (CH). It also implements a Multilayer Perceptron-Recurrent NN (MLP-RNN) method for intrusion detection, with weights fine-tuned by OCCOA to improve performance. Kushwah and Prasad [22] propose a Mobile Agent (MA)-based IDS technique.

In [23], an IDS employs a stack-based ensemble learning model. To address computational resource constraints, FS and hyperparameter optimization methods are integrated. Zuo et al. [24] introduce "GSOOA-1DDRSN," a methodology for detecting network traffic anomalies by utilizing a Deep Residual Shrinkage Network (DRSN) model. It employs an improved OS-OA to choose significant features and reduce dimensionality, improving detection performance with a one-dimensional DRSN (1DDRSN) as the classifier. The limitations of the existing studies comprise dependence on diverse FS models, which enhances complexity and negatively impacts real-time performance. Binary classification methods may oversimplify traffic types, while ensemble learning can restrict adaptability to dynamic environments. Moreover, sensitivity to parameter tuning may affect stability, and overfitting on smaller datasets can mitigate generalizability. There are also risks of losing significant data during feature reduction, which encompasses anomaly detection accuracy. Despite enhancements in IDS, there remains a gap in efficiently handling dynamic network environments and adapting to growing attack patterns. Current methodologies may face difficulty with balancing complexity and performance, and maintaining accuracy across diverse datasets. Furthermore, the integration of FS and optimization techniques requires additional exploration to improve real-time detection capabilities.

3. The Proposed Method

This paper proposes a novel EIDS-OSOAEL methodology. The technique mainly focuses on designing an ensemble classifier that integrates the output from multiple classes. It involves four stages of procedures: Min-Max Normalization, TT-MBFOA-based FS, Ensemble learning, and OS-OA-based tuning. Figure 1 portrays the structure of the EIDS-OSOAEL methodology.

3.1. Pre-Processing

Primarily, the EIDS-OSOAEL model involves a min-max scalar for scaling the input information into a uniform format. The term "data preparation" represents the action that should be carried out to encode or convert information so that the computer can understand and read it [25]. The model that underpins it should be capable of quickly analyzing data quality to generate precise and accurate model predictions. Assuming that the valuable information and quality of data resultant from it directly impacts the potential model for learning, the pre-processing of data before providing for into the method is significant.

The significant step of ML is data pre-processing. The data preparation process involves smoothing noisy data, filling in missing values, removing outliers, and resolving inconsistency, known as "filling in missing values". Normalizing a data component or group of independent variables is obtained using feature scaling.

During the data preparation phase, normalizing is frequently done in the data processing, known as data normalization. Min-max normalization or Min-max scaling is the simplest technique, which includes rescaling the range of factors to scale the interval of zero and one.

$$\frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

3.2. FS Using TT-MBFOA

The TT-MBFOA method is deployed to ensure the optimal FS. This approach goes back to the tandem twirl to improve its searchability and for simplification [26]. The TT-MBFOA technique is an efficient choice for FS due to its robust exploration capabilities and effectual convergence properties.

This methodology improves conventional bacterial foraging models by incorporating a twirling mechanism, which enhances the capability of the algorithm to navigate intrinsic feature spaces and avert local optima.

The TT-MBFOA model effectually balances exploration and exploitation, allowing it to detect the most relevant factors while minimizing redundancy. Furthermore, its population-based nature enables the simultaneous analysis of multiple solutions, accelerating the FS procedure.

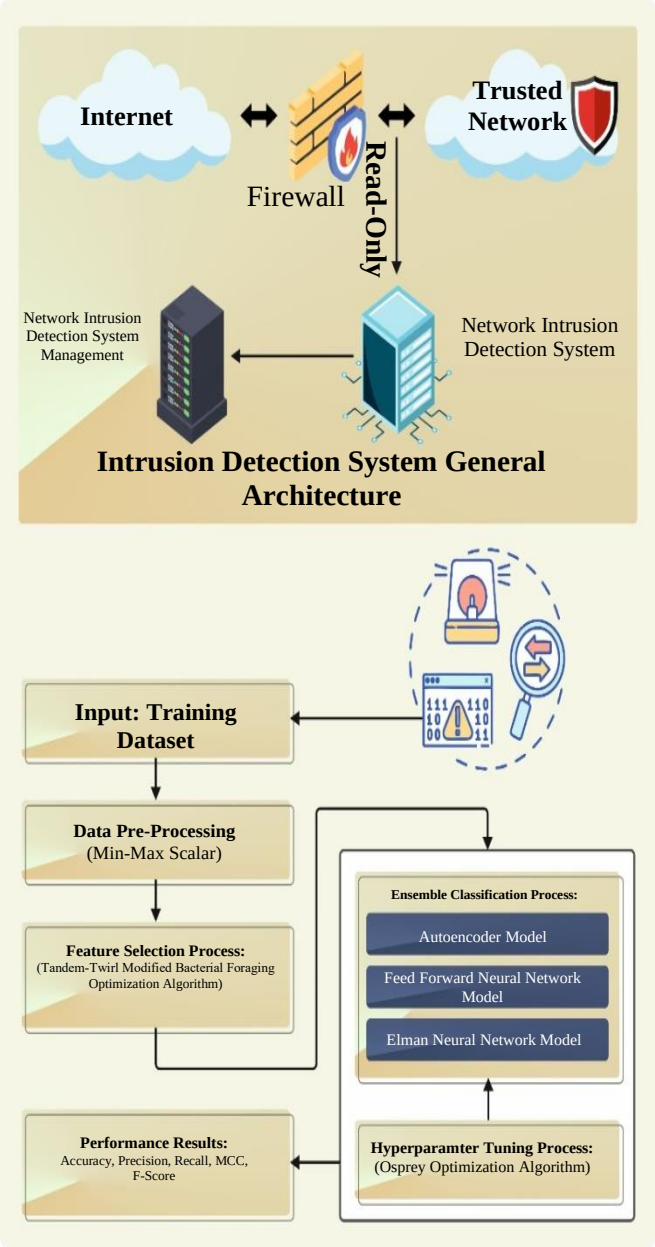

Fig. 1 Overall procedure of EIDS-OSOAEL methodology

The algorithm's adaptability makes it appropriate for high-dimensional datasets, confirming that the chosen features contribute meaningfully to the performance and interpretability of the model. On the whole, the TT-MBFOA model provides a compelling merit for FS tasks in diverse applications. A tandem twirl is applied within the chemotaxis process. The purpose of the first twirls is to complement the swarming operator by letting the bacterium search in the direction of arbitrarily chosen bacteria and other areas of search space. The second spin uses real twirls but with a minuscule step size value, which concentrates on the tiny bacterial motion. In the following, the proposed tandem twirls are briefly discussed.

3.2.1. Exploration Twirl

The initial twirl is calculated as follows:

$$\theta^i(j+1,G) = \theta^i(j,G) + \beta - 1\left(\theta^{r_1}(j,G) - \theta^{r_2}(j,G)\right) \quad (2)$$

In Equation (2), the user-defined value is more than 1 for the swarming operator of MBFOA. The swarm's arbitrarily chosen bacteria are $\theta^{r_2}(j,G)$ and $\theta^{r_2}(j,G)$ and $(i \neq r_1 \neq r_2)$, correspondingly. The twirl operator uses the position of both bacteria to choose the search direction and begin twirling from the bacteria $\theta^i(j,G)$.

The behaviours of the twirl operator with two decision variables range between -5 and 5. On the twirl completion, the new bacterium location is divided into the purple spot that bact1 and bact2 specify and are represented as $\theta^{r_2}(j,$ G) and $\theta^{r_2}(j,G)$, correspondingly. The best bacterium is inserted, which shows that these operators are working to obtain the search space (viz., not those existing in the neighbourhood of the present optimum solution as the swarm motion pushes it).

3.2.2. Exploitation Twirl

Based on random search direction, the second twirl reverts to the new twirl; however, it is combined with a lesser arbitrary step size value such that smooth motion might be accurately defined:

$$\theta^i(j+1,G) = \theta^i(j,G) + C(i,G)\varphi(i) \quad (3)$$

In Equation (3), the step size value consists of an n-dimension random vector represented as (i,G), calculated at all the generations as follows:

$$C(i,G)_k = R * \Delta_{(i)_k}, k = 1, \ldots, n \quad (4)$$

In Equation (4), $\Delta_{(t)}$ indicates the uniformly distributed random number in $[Lk, Uk]$ of search space k. R specifies a user-defined parameter to increase the size of the step, and its value should be nearly 0, for example, $5.00E-03$. In the 1st cycle, the step size calculation is performed by $\Delta_{(t)}$ to allow the bacteria in the early swarm to traverse in different directions and escape the attractor once the process begins.

This result is possible since TT- MBFOA involves a twirl for the exploration, a twirl to choose convergence, and a delicate twirl to increase the quality of the solution. The Fitness Function (FF) in the TT- MBFOA approach is designed for balancing the chosen features (lower) and the classification accuracy (superior) attained with those features. Equation (5) implies the FF for assessing performances.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (5)$$

Whereas, $\gamma_R(D)$ denotes the classifier error rate. $|R|$ signifies the cardinality of the chosen subset, and $|C|$ represents the overall feature counts from the dataset; α and β are the two parameters comparable to the impact of classifier quality and length of subset $\in$ [1,0] and $\beta = 1-\alpha$.

3.3. Ensemble Learning

For intrusion recognition, the EIDS-OSOAEL technique uses an ensemble of three classifiers: AE, FFNN, and ENN models, which offer various merits. AEs outperform unsupervised learning, enabling them to detect anomalies by reconstructing input data and emphasizing deviations efficiently. The FFNN model provides strong generalization capabilities, allowing them to learn intrinsic patterns in labeled data, while ENNs, with their recurrent architecture, are adept at capturing temporal dependencies in network traffic. This incorporation employs the merits of every technique, enhancing overall detection accuracy and robustness. Furthermore, the ensemble technique reduces the weaknesses of individual classifiers, resulting in more reliable performance in detecting diverse intrusion kinds. By incorporating these diverse architectures, the system can adapt to a wider range of attack scenarios and enhance its overall effectiveness in safeguarding network environments.

3.3.1. AE Model

To overcome the backpropagation in an unsupervised context, input is employed as output since input and output are similar. Rumelhart et al. first introduced an AE, which is classified as self-supervised [27]. According to Goodfellow et al., an AE is a trained NN to copy its input to the output". The Hidden Layers (HLs) count and the architecture of AE differ based on the usage scenario and the domain. Consider an input x, an AE has trained to minimize the reconstructed error, as follows:

$$\varphi: \mathcal{X} \to \mathcal{F} \qquad \psi: \mathcal{F} \to \mathcal{X} \quad (6)$$

$$\varphi, \psi = arg\min_{\varphi,\psi} ||\mathcal{X} - (\varphi \circ \psi)\mathcal{X}||^2$$

Where φ and ψ are the encoder and decoder functions, correspondingly. The reconstructed error signifies the difference between x and x':

$$x' = g(f(x))$$

In the Equation, $f(x)$ denotes the encoder function, which constructs the encoded vector of x, and $g(x)$ shows the decoder function and restores x to its initial value. The common functions of MAE and MSE compute the reconstructed error as follows.

$$MSE = \sum_{i=1}^{N}(x'-x)^2 \quad (7)$$

$$MAE = \sum_{i=1}^{N}|x'-x| \quad (8)$$

The AE corresponds to PCA, assuming that the encoder function $f(x)$ is a single-layered network with a linear function. Figure 2 demonstrates the architecture of the AE method.

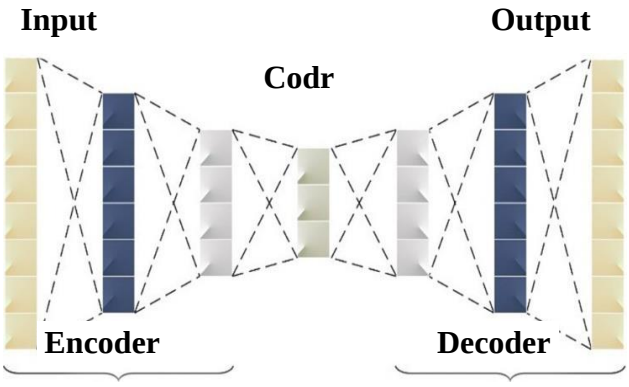

Fig. 2 AE structure

3.3.2. FFNN Model

This technique transfers data from the input through the HL to the output [28]. The NN and output layers are connected to the input and weight layers. The activation function of i^{th} hidden neurons are:

$$h_i = f(u_i) = f\left(\sum_{k=0}^{K} w_{ki}\, x_k\right) \quad (9)$$

Now, $f(u_i)$ shows the connection function that provides non-linearity amongst input and HLs, h_i indicates the i^{th} hidden neuron, x_k denotes the K input value and w_{ki} represents the weight in ki^{th} entry in $(K\times N)$ weight matrices:

$$y_j = f\left(u_j^1\right) = f\left(\sum_{i=1}^{N} w_{ij}\, h_i\right) \quad (10)$$

In Equation (10), y_j denotes the j^{th} output values.

3.3.3. ENN Model

An Elman network integrates context nodes. The context node attains input from the hidden nodes and transfers its output to the hidden node. As the context node relies on activating the hidden node from the prior input, the context node keeps the state data amongst inputs. This is mathematically modelled as below:

$$x(t) = |x_1(t), x_2(t), \dots, x_n(t)|^T \quad (11)$$

The output vector is described as:

$$y(t) = |y_1(t), y_2(t), \dots, y_n(t)|^T \quad (12)$$

The output vector of HL is represented as:

$$c(t-1) = |c_1(t-1), c_2(t-1), \dots, c_m(t-1)|^T \quad (13)$$

The output vector linked to the HL is:

$$x^2(t) = |x_1^2(t), x_2^2(t), \dots, x_m^2(t)|^T = c(t-1) \quad (14)$$

The input vector is:

$$x(t)[x_1(t), x_2(t), \dots, x_n(t); x_{n+1}^2(t), \dots, x_k^2(t)]^T = [x(t)]^T[x^2(t)]^T = [x_1(t), x_2(t), \dots, x_n(t), c_1^2(t-1), c_m^2(t-1)]^T \quad (15)$$

Where $k = m + n$.

$$y_i(t) = f(a_i^o(t) = \frac{1}{1+\exp\left(-a_i^o(t)\right)}, i = 1,2, \dots, n \quad (16)$$

$$a_i^o(t) = \sum_{j=1}^{m} W_{ji}^{o,h}(t) \times h_j(t), i = 1,2, \dots, n \quad (17)$$

The relationship between the input, the context, and the HL weight matrices are defined by:

$$W^h(t) = \left|W^{h,i}(t)W^{h,c}(t)\right| \quad (18)$$

The output of input vector $x(t)$ is:

$$h_j(t) = f\left(a_j^h(t)\right) = \frac{1}{1+\exp\left(-a_j^h(t)\right)}, j = 1,2, \dots, m \quad (19)$$

$$a_j^h(t) = \sum_{l=1}^{k} W_{jl}^h(t) \times x_l(t), j = 1,2, \dots, m \quad (20)$$

The training of ELMAN aims to reduce the MSE:

$$E(t) = \frac{\|e(t)\|^2}{2} \quad (21)$$

$$e(t) = d(t) - y(t) \quad (22)$$

Here, $d(t)$ is the desired output. The training model reduces $E(t)$ by measuring the weight:

$$W^{o,h}(t+1) = W^{o,h}(t) - \mu\frac{\partial E(t)}{\partial W^{o,h}(t)} = W^{o,h}(t) - \mu y'(t)e(t)h^T(t) \quad (23)$$

$$W^h(t+1) = W^h(t) - \mu\frac{\partial E(t)}{\partial W^h(t)} = W^h + \mu h'(t)[W^{o,h}(t)]^T y'(t)e(t)x^T(t) \quad (24)$$

$$y'(t) = diag\left[f'\left(a_1^o(t)\right), f'\left(a_2^o(t)\right), \dots, f'\left(a_n^o(t)\right)\right] \quad (25)$$

$$h'(t) = diag\left[f'\left(a_1^h(t)\right), f'\left(a_2^h(t)\right), \dots, f'\left(a_m^h(t)\right)\right] \quad (26)$$

Where μ denotes the learning rate of ELMAN. The input, context, and output weight matrices are $W^{h,i}(t)$, $W^{h,c}(t)$ and $W^{o,h}(t)$.

3.4. Hyperparameter Tuning utilizing OS-OA

Eventually, the OS-OA method is utilized to adjust the hyperparameter values of these models [29]. The OS-OA model is particularly well-suited for hyperparameter optimization due to its ability to balance exploration and exploitation effectively. Its unique structure allows for

adaptive adjustments during the search process, which enhances convergence speed and accuracy. Unlike traditional optimization techniques, OS-OA incorporates chaotic elements that improve randomness, reducing the likelihood of getting trapped in local optima. This is critical in high-dimensional parameter spaces often encountered in ML models. Furthermore, its simplicity and efficiency make it computationally feasible, even in resource-constrained environments. By applying OS-OA, we can achieve better performance in tuning hyperparameters, ultimately leading to more effective and reliable IDSs. In recent times, various optimization models have been established in prior research to finetune the parameter with minimal iteration. However, the classical optimizer fails to generate a global optimum solution and quickly falls into early convergence. Therefore, the study presents a new OS-OA technique for hyperparameter tuning of the presented method. The Osprey is generally called a sea hawk that eats prey in a wide range. The mathematical modelling of the presented technique is discussed below.

3.4.1. Initialization Stage

Here, the ospreys are initialized at random in the search range, and the mathematical expression is represented by,

$$z_{a,b} = LB_b + rand_{a,b} \times (UB_b - LB_b), a = 1,2, \cdots \cdots M, b = 1_t 2 \ldots \ldots x \quad (27)$$

In Equation (27), x denotes the overall problem variables; $z_{a,b}$ represents the problem variable of b^{th} dimensions; UB_b and LB_b are the upper and lower boundaries of b^{th} dimensions; M represents the number of ospreys, and $rand_{\mathrm{a},b}$ specifies the random number within [0,1].

3.4.2. FF

On every iteration, the accuracy of the technique is evaluated for the FF, and the mathematical expression is given below:

$$f = \max(accuracy) \quad (28)$$

3.4.3. Exploration Phase

In the exploration phase, the Osprey identifies an arbitrary place to attack the fish (prey). The updated osprey position is mathematically expressed depending on the osprey movement towards the fish.

$$z_{a,b}^{l_1} = z_{a,b} + rand_{a,b}\left(\beta_{a,b} - y_{a,b} \times z_{a,b}\right) \quad (29)$$

$$z_{a,b}^{l_1} = \begin{cases} z_{a,b}^{l_1}, LB_b \le z_{a,b}^{l_1} \le UB_b \\ LB_b, z_{a,b}^{l_1} < LB_b \\ UB_b, z_{a,b}^{l_1} > UB_b \end{cases} \quad (30)$$

Now, $z_{a,b}^{l_1}$ indicates the updated location of a^{th} ospreys in b^{th} dimensions, $\beta_{a,b}$ shows that the fish have been carefully chosen by a^{th} ospreys, $rand_{\mathrm{a},b}$ specifies the arbitrary value in [0,1], $y_{a,b}$ depicts the random number in the range of [1,2].

3.4.4. Exploitation Phase

In the exploitation phase, the Osprey selects the best, safest place to eat the fish, and the mathematical formula for updating the osprey location is given below,

$$z_{a,b}^{l_2} = z_{a,b} + \frac{LB_b + rana_{a.b} \times (UB_b - LB_b)}{f},$$

$$a = 1,2, \ldots . M, b = 1,2, \cdots x, t = 1,2 \ldots . T \quad (31)$$

$$z_{a,b}^{l_2} = \begin{cases} z_{a,b}^{l_2}, LB_b \le zz_{a,b}^{l_2} \le UB_b \\ LB_b, z_{a,b}^{l_2} < LB_b \\ UB_b, z_{a,b}^{l_2} > UB_b \end{cases} \quad (32)$$

In Equation (32), $z_{a,b}^{l_2}$ is the updated location of a^{th}osprey in b^{th} dimension, T shows the overall iterations and $rand_{\mathrm{a},b}$ refers to the random integers ranging from zero to one. Lastly, the optimum global solution is attained from the proposed OS-OA method for accurately tuning the parameter.

The first step is initializing the population of the metaheuristic approach to enhance DSAtt-CMNetV3's hyperparameter; the hyperparameter count indicates the number of dimensions to improve. Next, update the location of OS-OA. The optimum performance has been selected from the novel result, and the procedure still attains the novel performance.

This proposed OS-OA is utilized to enhance classification accuracy. Fitness selection is crucial for determining the outcome of the OS-OA model. It involves assessing the encoded process's performance to evaluate the effectiveness of candidate outputs. In this context, OS-OA prioritizes accuracy as the key criterion for designing the fitness function, defined as:

$$Fitness = \max(P) \quad (33)$$

$$P = \frac{TP}{TP + FP} \quad (34)$$

TP and FP demonstrate the true and false positive rates.

4. Performance Validation

The experimentation of the EIDS-OSOAEL methodology is performed by using a standard dataset [30]. The dataset encompasses 125973 instances under five classes, as stated in Table 1. The simulation is performed by utilizing the Python 3.6.5 tool on PC i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings are the rate of learning: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5. Figure 3 depicts the confusion matrices of the EIDS-OSOAEL approach under 60:40 and 70:30 of TRAPH/TESPH. The simulation value implied that the EIDS-OSOAEL method effectively recognizes and classifies all five classes.

Table 1. Dataset specification

Classes	Instance Numbers
DoS	45927
R2l	995
Probe	11656
U2r	52
Normal	67343
Overall samples	**125973**

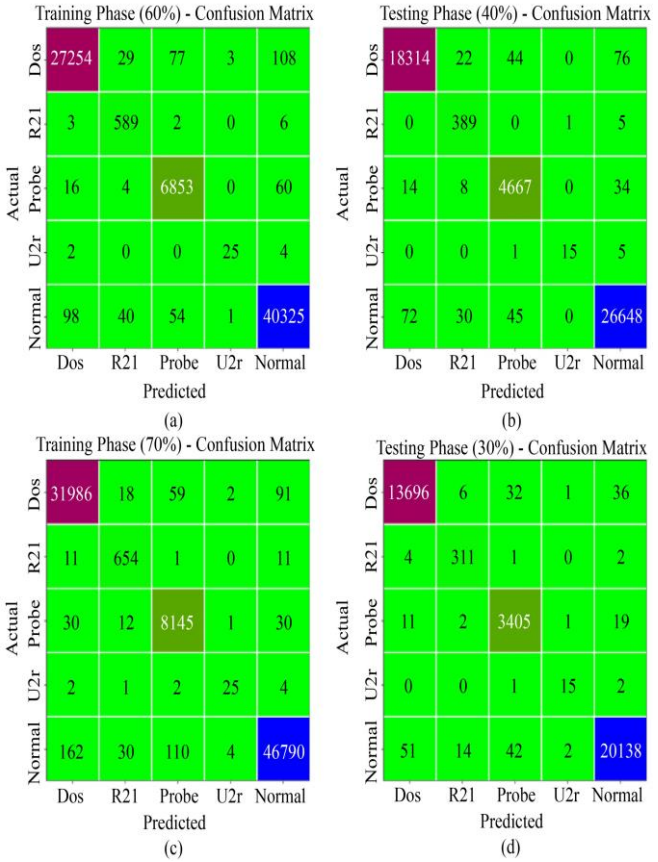

Fig. 3 Confusion matrices of (a-c) 60:70 of TRAPH and (b-d) 40:30 of TESPH

In Table 2 and Figures 4-5, the IDS outcome of the EIDS-OSOAEL methodology is exposed under 60% of TRAPH and 40% of TESPH. The outputs ensure the enhanced capability of the EIDS-OSOAEL methodology to detect various five classes. With 60% of TRAPH, the EIDS-OSOAEL methodology offers $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.72%, 94.40%, 95.26%, 94.77%, and 94.57%, respectively.

Also, with 40% of TESPH, the EIDS-OSOAEL approach offers $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.72%, 95.52%, 93.48%, 94.12%, and 94.08%, correspondingly. In Table 3 and Figures 6-7, the IDS performances of the EIDS-OSOAEL methodology are depicted under 70% of TRAPH and 30% of TESPH. The simulation value ensures the EIDS-OSOAEL technique's improved ability to recognize multiple classes.

Table 2. IDS output of EIDS-OSOAEL approach under 60:40 of TRAPH/TESPH

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRAPH (60%)					
Dos	99.56	99.57	99.21	99.39	99.04
R2l	99.89	88.97	98.17	93.34	93.40
Probe	99.68	97.68	98.85	98.26	98.08
U2r	99.99	86.21	80.65	83.33	83.37
Normal	99.47	99.56	99.45	99.51	98.93
Average	**99.72**	**94.40**	**95.26**	**94.77**	**94.57**
TESPH (40%)					
Dos	99.55	99.53	99.23	99.38	99.02
R2l	99.87	86.64	98.48	92.18	92.31
Probe	99.71	98.11	98.81	98.46	98.30
U2r	99.99	93.75	71.43	81.08	81.83
Normal	99.47	99.55	99.45	99.50	98.94
Average	**99.72**	**95.52**	**93.48**	**94.12**	**94.08**

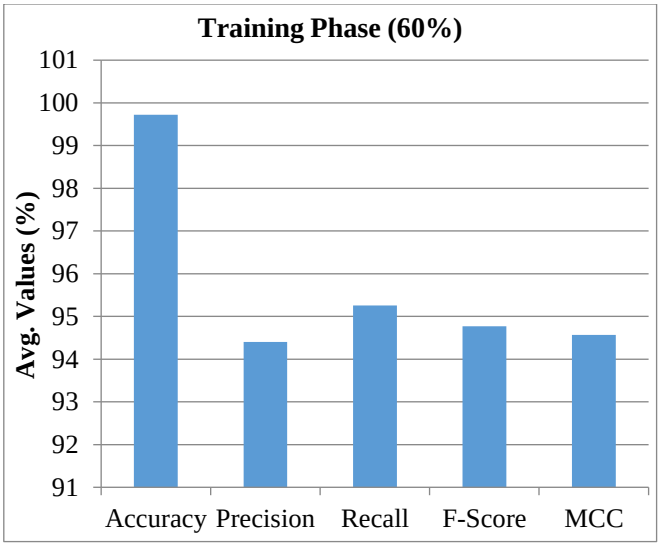

Fig. 4 Average of EIDS-OSOAEL method under 60% of TRAPH

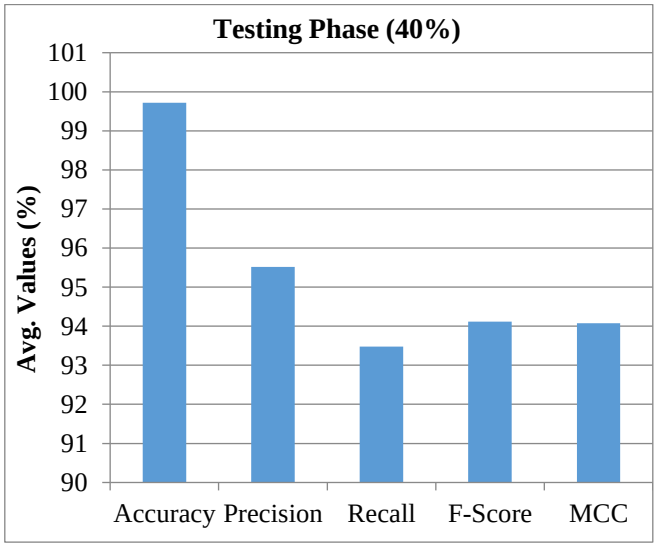

Fig. 5 Average of EIDS-OSOAEL method under 40% of TESPH

Table 3. IDS output of EIDS-OSOAEL approach under 70:30 of TRAPH/TESPH

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRAPH (70%)					
Dos	99.57	99.36	99.47	99.42	99.08
R2l	99.90	91.47	96.60	93.97	93.95
Probe	99.72	97.93	99.11	98.52	98.37
U2r	99.98	78.12	73.53	75.76	75.78
Normal	99.50	99.71	99.35	99.53	98.99
Average	**99.74**	**93.32**	**93.61**	**93.44**	**93.24**
TESPH (30%)					
Dos	99.63	99.52	99.46	99.49	99.19
R2l	99.92	93.39	97.80	95.55	95.53
Probe	99.71	97.82	99.04	98.42	98.27
U2r	99.98	78.95	83.33	81.08	81.10
Normal	99.56	99.71	99.46	99.58	99.11
Average	**99.76**	**93.88**	**95.82**	**94.82**	**94.64**

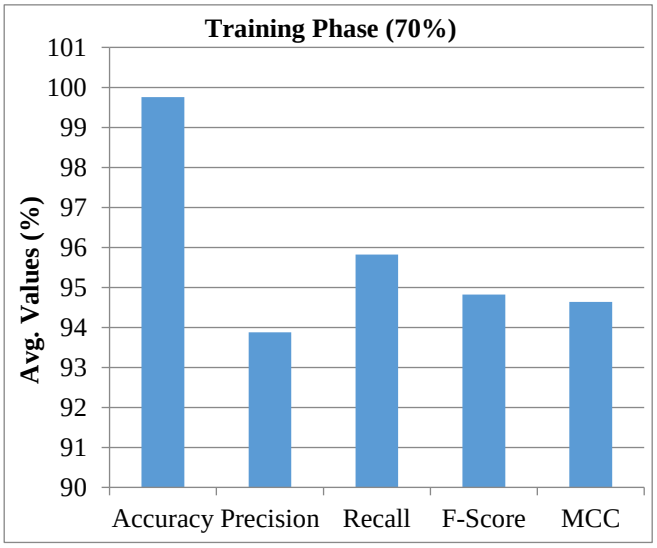

Fig. 6 Average of EIDS-OSOAEL approach under 70% of TRAPH

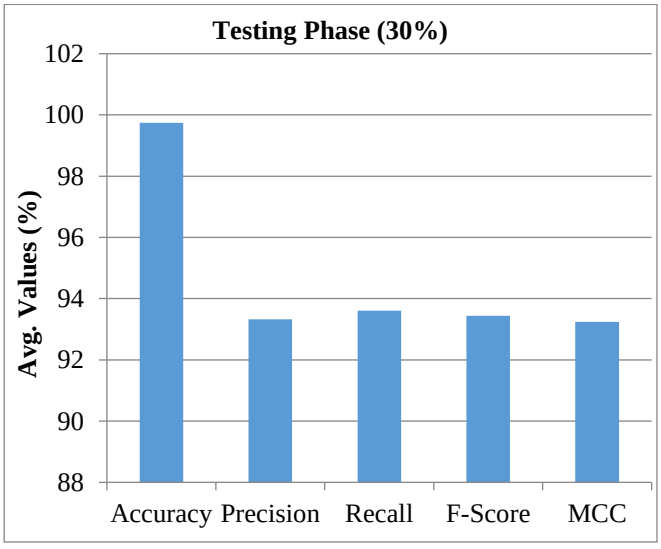

Fig. 7 Average of EIDS-OSOAEL approach under 30% of TESPH

With 70% of TRAPH, the EIDS-OSOAEL method provides $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.74%, 93.32%, 93.61%, 93.44%, and 93.24%, correspondingly. Moreover, with 30% of TESPH, the EIDS-OSOAEL method gains $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC of 99.76%, 93.88%, 95.82%, 94.82%, and 94.64%, respectively. The performance of the EIDS-OSOAEL model using a 70:30 TRAPH/TESPH ratio is shown in Figure 8, exhibiting the accuracy curves of training (TRAA) and validation (VALA). The figure illustrates the learning and generalization abilities of the model over diverse epochs, with a consistent increase in TRAA/VALA. This growth highlights the EIDS-OSOAEL method's adaptability in detecting patterns within TRA/TES data and its capability to classify unseen data, underscoring strong generalization performance precisely. Figures 9 and 10 depict the loss of training (TRLA) and validation (VALL) of the EIDS-OSOAEL approach using a 70:30 TRAPH/TESPH ratio over diverse epochs, emphasizing a consistent reduction in TRLA that represents efficient weight optimization and lessened classifier error. The figures accentuate the EIDS-OSOAEL approach's robust association with TRA data, accentuating its capability to understand patterns in both datasets. Furthermore, the PR curve in Figure 10 exhibits that the EIDS-OSOAEL methodology attains enhanced precision and recall across all classes, confirming its improved capabilities in detecting and recognizing diverse classes effectually.

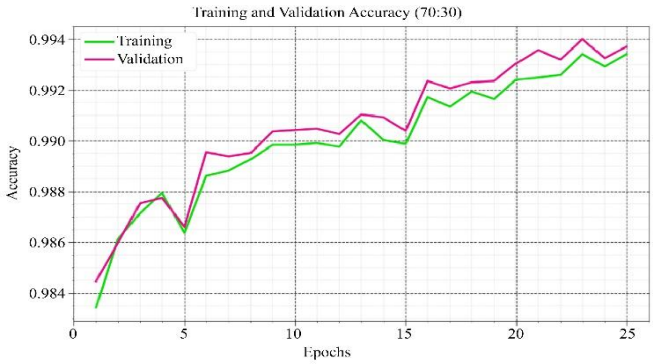

Fig. 8 $Accu_y$ curve of EIDS-OSOAEL method sunder 70:30 of TRAPH/TESPH

Fig. 9 Loss curve of EIDS-OSOAEL model under 70:30 of TRAPH/TESPH

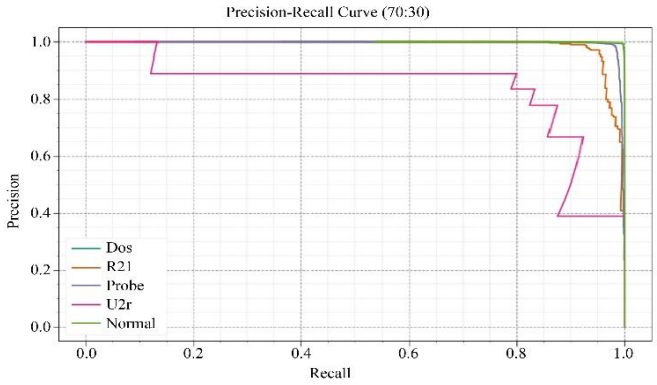

Fig. 10 PR curve of EIDS-OSOAEL model under 70:30 of TRAPH/TESPH

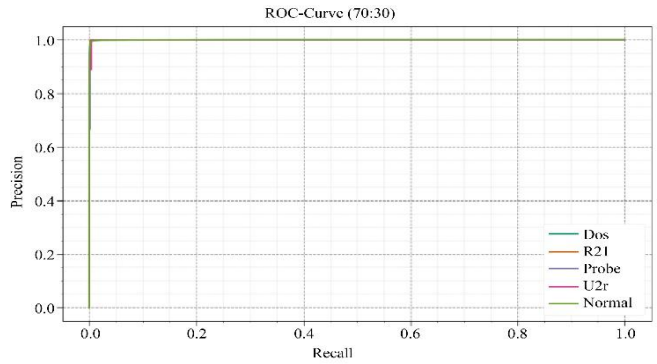

Fig. 11 ROC curve of EIDS-OSOAEL model under 70:30 of TRAPH/TESPH

In Figure 11, ROC curves from the EIDS-OSOAEL approach with a 70:30 TRAPH/TESPH ratio excelled in classifying various classes. It presents an elaborated depiction of the TPR/FRP trade-offs over diverse threshold values and epochs. The figure illustrates that the EIDS-OSOAEL model outperformed across all classes, underscoring its efficiency. A detailed relational study of the EIDS-OSOAEL approach is performed with existing methods on IDS in Table 4 and Figure 12 [31]. The investigational value implies that the NB and KNN techniques exhibited worse performance with $accu_y$ of 89.57% and 94.22%. In addition, the LR and SVM models have somewhat higher outcomes, with $an\ accu_y$ of 94.73% and 96.03%. Meanwhile, the IntruDTree and FSHDBN-CID approaches reached closer outcomes with $accu_y$ of 97.60% and 99.36%. Nevertheless, the EIDS-OSOAEL technique gains maximal performance with an $accu_y$ of 99.76%. At last, a comprehensive Computational Time (CT) output of the EIDS-OSOAEL methodology is performed with existing methods on IDS in Table 5 and Figure 13. The value indicates that the SVM and IntruDTree methodologies have resulted in poor accomplishment with CT of 11.38s and 11.05s. The NB and KNN approaches have reasonable outcomes, with CTs of 10.31s and 9.68s. In the meantime, the LR and FSHDBN-CID methodologies reached closer performances with CT of 9.30s and 8.89s. Nevertheless, the EIDS-OSOAEL technique gains a lesser solution with a CT of 5.04s.

Table 4. $Accu_y$ outcome of EIDS-OSOAEL technique with existing methods

Methods	Accuracy (%)
EIDS-OSOAEL	99.76
FSHDBN-CID	99.36
NB	89.57
LR	94.73
KNN	94.22
SVM	96.03
IntruDTree	97.60

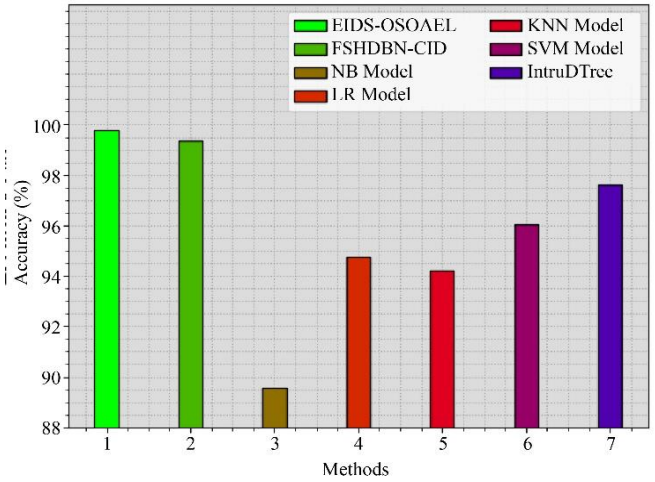

Fig. 12 $Accu_y$ outcome of EIDS-OSOAEL technique with existing methods

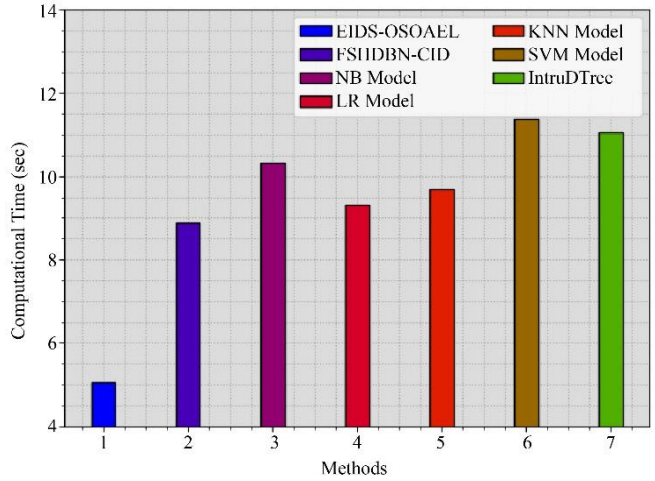

Fig. 13 CT outcome of EIDS-OSOAEL technique with existing methods

Table 5. CT output of EIDS-OSOAEL technique with existing models

Techniques	CT (sec)
EIDS-OSOAEL	5.04
FSHDBN-CID	8.89
NB	10.31
LR	9.30
KNN	9.68
SVM	11.38
IntruDTree	11.05

5. Conclusion

This paper proposes a novel EIDS-OSOAEL methodology. The EIDS-OSOAEL technique mainly focuses on the design of an ensemble classifier that integrates the output from multiple classes. It involves four stages of processes, namely Min-Max normalization, TT-MBFOA-based FS, ensemble learning, and OS-OA-based tuning. Primarily, the EIDS-OSOAEL technique consists of a min-max scalar for scaling the input data into a uniform format.

Besides, the TT-MBFOA approach is exploited for the optimum choice of features. For intrusion detection, the EIDS-OSOAEL technique undergoes an ensemble of three classifiers, namely AE, FFNN, and ENN. The OS-OA approach is utilized to adjust the hyperparameter values of these models. The experimental results of the EIDS-OSOAEL approach are evaluated under a standard dataset. The performance validation of the EIDS-OSOAEL model portrayed a superior 99.76% over recent approaches. The limitations of the EIDS-OSOAEL model encompass potential threats in computational effectualness due to the complexity of training multiple NNs concurrently. The model may also face difficulty with scalability when applied to very massive datasets, resulting in enhanced training times and resource demands. Furthermore, dependence on specific architectures could limit adaptability to growing attack patterns, as the model may need retraining to maintain performance. Future work may concentrate on optimizing the model for faster execution, exploring lightweight architectures, and integrating more adaptive learning methods. Moreover, improving the capability of the technique to generalize across diverse network environments and incorporating real-time data streams could additionally enhance its effectualness in intrusion detection. Finally, investigating hybrid models that integrate ensemble learning with other optimization methods may yield even improved outcomes.

References

[1] Wansoo Kim et al., "Vehicular Multilevel Data Arrangement-Based Intrusion Detection System for In-Vehicle CAN," *Security and Communication Networks*, vol. 2022, no. 1, pp. 1-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Evgenia Novikov, Elena Doynikova, and Sergey Golubev, "Federated Learning for Intrusion Detection in the Critical Infrastructures: Vertically Partitioned Data Use Case," *Algorithms*, vol. 15, no. 4, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jing Yu, Xiaojun Ye, and Hongbo Li, "A High Precision Intrusion Detection System for Network Security Communication Based on Multi-Scale Convolutional Neural Network," *Future Generation Computer Systems*, vol. 129, pp. 399-406, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Arun Kumar Bediya, and Rajendra Kumar, "A Novel Intrusion Detection System for Internet of Things Network Security," *Journal of Information Technology Research*, vol. 14, no. 3, pp. 1-18, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shaokang Cai et al., "A Hybrid Parallel Deep Learning Model for Efficient Intrusion Detection based on Metric Learning," *Connection Science*, vol. 34, no. 1, pp. 551-577, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Francesco Pascale et al., "Cybersecurity in Automotive: An Intrusion Detection System in Connected Vehicles," *Electronics*, vol. 10, no. 15, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Varun Prabhakaran, and Ashokkumar Kulandasamy, "Integration of Recurrent Convolutional Neural Network and Optimal Encryption Scheme for Intrusion Detection with Secure Data Storage in the Cloud," *Computational Intelligence*, vol. 37, no. 1, pp. 344-370, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Georgios Spathoulas, Georgios Theodoridis, and Georgios-Paraskevas Damiris, "Using Homomorphic Encryption for Privacy-Preserving Clustering of Intrusion Detection Alerts," *International Journal of Information Security*, vol. 20, pp. 347-370, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mahendra Prasad, Sachin Tripathi, and Keshav Dahal, "A Probability Estimation-Based Feature Reduction and Bayesian Rough Set Approach for Intrusion Detection in Mobile Ad-Hoc Network," *Applied Intelligence*, vol. 53, pp. 7169-7185, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Sébastien Canard, and Chaoyun Li, "Towards Practical Intrusion Detection System over Encrypted Traffic," *IET Information Security*, vol. 15, no. 3, pp. 205-266, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Hasanain Ali Al Essa, and Wesam S. Bhaya, "Ensemble Learning Classifiers Hybrid Feature Selection for Enhancing Performance of Intrusion Detection System," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 665-676, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Doaa N. Mhawi, and Sokeana H. Hashim, "Proposed Hybrid Ensemble Learning Algorithms for an Efficient Intrusion Detection System," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 22, no. 2, pp. 73-84, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Óscar Mogollón-Gutiérrez et al., "A Novel Ensemble Learning System for Cyberattack Classification," *Intelligent Automation & Soft Computing*, vol. 37, no. 2, pp. 1691-1709, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Zakaria Abou El Houda, Bouziane Brik, and Lyes Khoukhi, "Ensemble Learning for Intrusion Detection in SDN-based Zero Touch Smart Grid Systems," *2022 IEEE 47th Conference on Local Computer Networks*, Edmonton, AB, Canada, pp. 149-156, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yakub Kayode Saheed, and Sanjay Misra, "A Voting Gray Wolf Optimizer-Based Ensemble Learning Models for Intrusion Detection in the Internet of Things," *International Journal of Information Security*, vol. 23, pp. 1557-1581, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Thi-Thu-Huong Le et al., "Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method," *Sensors*, vol. 22, no. 3, pp. 1-28, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Wenbin Yao et al., "A Lightweight Intelligent Network Intrusion Detection System Using One-Class Autoencoder and Ensemble Learning for IoT," *Sensors*, vol. 23, no. 8, pp. 1-25, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Deepak Kumar et al., "ChOs_LSTM: Chebyshev Osprey Optimization-Based Model for Detecting Attacks," *2024 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT)*, Vellore, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Orieb Abu Alghanam et al., "An Improved PIO Feature Selection Algorithm for IoT Network Intrusion Detection System Based on Ensemble Learning," *Expert Systems with Applications*, vol. 213, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Bingying Yao et al., "Modified Osprey Algorithm for Optimizing Capsule Neural Network in Leukemia Image Recognition," *Scientific Reports*, vol. 14, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ranjeet B. Kagade, and N. Vijayaraj, "OCCOA for Clustering-Based Intrusion Detection System with MLP-RNN Architecture," *Multimedia Tools and Applications*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Rajendra Singh Kushwah, and Kamal Kishor Prasad, "A Novel Mobile Agent-Based Intrusion Detection Framework for Network Security Using Sl-Gat and Pp-FQCC," *Educational Administration: Theory and Practice*, vol. 30, no. 5, pp. 5051-5062, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Yongzhong Cao et al., "An Intrusion Detection System Based on Stacked Ensemble Learning for IoT Network," *Computers and Electrical Engineering*, vol. 110, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Fengqin Zuo et al., "GSOOA-1DDRSN: Network Traffic Anomaly Detection Based on Deep Residual Shrinkage Networks," *Heliyon*, vol. 10, no. 11, pp. 1-23, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Abdullah H. Al-Nefaie, and Theyazn H. H. Aldhyani, "Predicting CO_2 Emissions from Traffic Vehicles for Sustainable and Smart Environment Using a Deep Learning Model," *Sustainability*, vol. 15, no. 9, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] C. Kalamani et al., "An Efficient Reconfigurable FIR Filter Design with Coefficient Optimization Using a Modified Bacterial Foraging Optimization Algorithm," *Automatika*, vol. 65, no. 1, pp. 290-303, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Hanan Hindy et al., "Utilizing Deep Learning Techniques for Effective Zero-Day Attack Detection," *Electronics*, vol. 9, no. 10, pp. 1-16, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] David Opeoluwa Oyewola, Emmanuel Gbenga Dada, Sanjay Misraet, "Diagnosis of Cardiovascular Diseases by Ensemble Optimization Deep Learning Techniques," *International Journal of Healthcare Information Systems and Informatics*, vol. 19, no. 1, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[29] Shaik Salma Asiya Begum, and Hussain Syed, "GSATT-CMNetV3: Pepper Leaf Disease Classification Using Osprey Optimization," *IEEE Access*, vol. 12, pp. 32493-32505, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[30] Datasets, Canadian Institute for Cybersecurity. [Online]. Available: https://www.unb.ca/cic/datasets/

[31] Khalid A. Alissa et al., "Feature Subset Selection Hybrid Deep Belief Network Based Cybersecurity Intrusion Detection Model," *Electronics*, vol. 11, no. 19, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]