Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

WOA-SA: OPTIMIZING NIDS WITH ENHANCED DEEP LEARNING FOR ZERO-DAY ATTACK DETECTION

MOHAMMED SAYEEDUDDIN HABEEB*

Research Scholar, Department of Electronics and Communication Engineering, University College of Engineering, Acharya Nagarjuna University, Andhra Pradesh, India.
*Corresponding Author Email: msayeeduddinhabeeb@gmail.com

TUMMALA RANGA BABU

Department of Electronics & Communication Engineering, R.V.R. & J.C.College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India.

Abstract

Network Intrusion Detection Systems (NIDS) are vital in safeguarding computer networks from cyber threats. However, designing an effective NIDS configuration involves optimizing multiple objectives, often leading to suboptimal solutions. This paper presents an innovative approach combining two powerful optimization methods, the Whale Optimization Algorithm (WOA) and Simulated Annealing (SA), for feature selection for NIDS. Our proposed WOA-SA methodology aims to achieve superior results by balancing global exploration and local improvement capabilities. Additionally, Deep Learning (DL) techniques are integrated to enhance feature extraction and classification accuracy for zero-day and new types of attacks with optimal DL models. This paper provides a detailed exposition of WOA-SA for feature selection and its practical application to NIDS optimization. This paper aims to achieve the maximum detection rate for zero-day attacks while reducing the false alarm rate (FAR) and reducing computational complexity. The comprehensive analysis of DL different approaches such as Long Short-term Memory, Convolutional Neural Networks, Recurrent Neural Networks, and Deep Neural Networks was carried out on the original and optimal feature set of the BOT-IOT 2020 dataset. From WOA-SA the feature set was reduced to 13 from 79, these 13 selected feature set performances were tested using the DL approach. Experimental results show that model accuracy improved with optimal features, it was improved to 2.2% and also reduced in FAR of the model to 10%, to show how well the optimal feature-based NIDS model performs in comparison to other well-known DL approaches. The proposed method also shows reduced computational complexity due to a reduced number of features. On the whole, our proposed design outperforms the current approach in terms of computational complexity, zero-day attack detection, accuracy, and FAR.

Keywords: Network Intrusion Detection system (NIDS), Whale Optimization Algorithm (WOA), Simulated Annealing (SA), Deep Learning (DL), Deep Neural Network (DNN), Attack.

1. INTRODUCTION

The rapid growth of the internet and communication technologies results in larger network sizes and increased complexity. This increase in network size results in an exchange of massive amounts of data between the different parts of the network, we must ensure the security of this data from attackers [1]. To ensure the security of this massive data, different securing techniques like anti-virus, firewall, authentical, etc. can be used to safeguard as a first security cover. However, these security measures can only handle known attacks, it is not beneficial for new patterns or updated versions of attacks. So, to overcome this and increase security, we need to add a second security cover, which is an intrusion detection system (IDS) [2]. Based upon the deployment IDS can be placed

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

at the host or network site. In this paper, we are going to discuss the network intrusion detection system (NIDS) which is deployed at the network site at the entry point [3]

NIDS plays an important role in safeguarding and protecting the network from attackers by continuously monitoring the network traffic. It will identify and stop any unusual and malicious activity in the network that may compromise the safety and security of the network, this is monitored by the network administrator. The efficient NIDS must differentiate between anomaly and normal traffic quickly and respond accurately, but this traditional NIDS cannot handle new attack patterns in the network efficiently and more false alarm rate (FAR). Many researchers have come up with ideas beyond the conventional method to achieve quick and better detection accuracy with a reduced false alarm rate (FAR). As attackers are becoming smarter and the attacks become more sophisticated and difficult to detect, the conventional approach of NIDS is no longer effective against new and updated types of attacks. So, we need to update the IDS, this updated NIDS should learn and adapt attack patterns intelligently on its own for better detection accuracy[4].

This intelligent system requires more computation complexity and resources, resulting in decreased efficacy. By introducing feature selection in NIDS, we can minimize the resources, and reduce the computation complexity to process the big data. Feature selection and model classification are the two fundamental steps that are commonly involved in NIDS. The feature selection method is a technique for selecting the most relevant features and removing unnecessary and redundant information from the dataset, increasing the efficiency of NIDS. An increase in performance and reduction in complexity is the main aim of the feature engineering on the dataset. While removing the irrelevant features from the dataset, additional care has to be taken with the key features that contain information about the behavior of the dataset, so these key features cannot be deleted [5]. The optimization technique is introduced to increase the efficiency of NIDS by selecting optimal features from the dataset, this results in minimization of resources and computational complexity. These optimization techniques can significantly increase the accuracy of cybersecurity by enhancing the NIDS to identify threats and lowering false alarms [6].

In order to achieve effective NIDS, in this paper we introduced a novel approach that improves NIDS by utilizing optimization methods and a deep learning approach. In particular, we optimize feature selection using the Whale Optimization Algorithm (WOA) and Simulated Annealing to enhance the performance of NIDS [7]. WOA is naturally inspired by the collective hunting strategy of humpback whales. Simulated annealing is similar to a smart problem-solving technique that investigates multiple options, sometimes making poor decisions, to find the optimum answer to a challenging issue. This approach is used to find the best features from the dataset. Combining these two approaches, Whale Optimization with a Simulated Annealing (WO-SA) framework offers a dynamic and complete strategy for NIDS optimization [8].

The effective NIDS is proposed by researchers based on machine learning (ML) and deep learning (DL). In this paper, we are using DL to train and validate our dataset, as we know

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

deep learning is deployed in deep architecture, and it has powerful learning capabilities that result in a good detection rate. The model is trained and validated using the selected optimized features from the dataset.

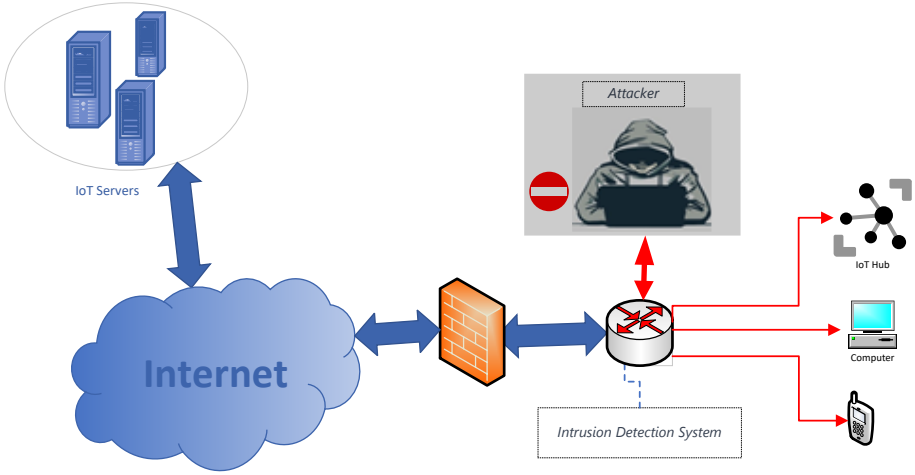
- Feature Selection Optimization Using WOA, WOA improves NIDS in choosing the most relevant features from the dataset.
- We use the K-Nearest Neighbours (KNN) technique to assess the fitness of each feature subset (represented by a whale's location). Choose the feature subset that exhibits the best performance in terms of classification precision and intrusion detection. The best search agent among whales is thought to be this top-performing feature subset.
- We employ Simulated Annealing for fine-tuning deep-learning model parameters to enhance NIDS adaptability and robustness.
- A simple but advanced technique for NIDS is presented through the integration of optimized features into deep learning models. This strategy strengthens NIDS overall security in network environments by increasing the efficiency of the systems as well as flexibility, and accuracy.

This paper is organized as Section 2 discusses current work trends in the field of NIDS IoT security. Section 3 describes the Proposed Hybrid Simulated Whale Optimization Algorithm Feature Selection Model. Section 4 describes the experimental setup, dataset description, and performance criteria employed to assess the suggested technique. Discusses the findings and gives a comparison to existing methodologies. Finally, Section 5 summarizes the contributions and highlights potential future study paths to bring the work to a close.

2. RELATED WORK

Artificial intelligence-enabled NIDS has attracted many researchers' attention in recent years due to its effective zero-day attack detection Figure 1 shows the IoT NIDS scenario. Both ML and DL-enabled NIDS are effective, but DL-based NIDS are preferred by many researchers due to their deep learning capability and fast computation [16]. One very important factor is the increased availability of online and standalone GPUs. By using the deep framework DL can learn the complex pattern and can identify the attack or normal activity in the network. Due to these reasons, many researchers have preferred DL-based NIDS in IoT networks in recent times [9], [10]

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842



Figures 1: NIDS IoT scenario

Different traditional ML algorithms with hybrid and ensemble methods are implemented and their performance matrix is calculated by the authors in the article. ML can classify unknown attacks perfectly but still struggles to handle enormous amounts of data. to solve problems related to datasets for intrusion detection, some of the studies came up with novel methods to address effectiveness problems related to machine learning methodologies. Some of the authors also came up with other approaches with Gradient boost, ensemble classifiers, and random forest (RF) which show batter detection accuracy but increased computer complexity and time taken for model training. Time constraints and computational complexity are also important elements in NIDS to look into. Accuracy and prediction time were used as criteria for evaluation when the author suggested a DL-based network intrusion detection system and tested it against various datasets [11].

The size of the dataset increases model learning time also increases for training. To overcome this issue, some of the authors proposed dimensionality reduction without affecting the information in the dataset. it can be achieved by removing redundant features from the original dataset, this is called feature engineering or feature selection process. It is important for building a powerful machine-learning model. Feature selection is classified as filter, wrapper, and raking-based, based upon this criterion features are selected and other features found to be redundant are eliminated from the dataset. These selected features are used to train the ML and DL models [12].

Some of the authors also introduced optimization techniques for feature selection, The particle swarm technique and a fast-learning network were used for feature selection [26]by the author to suggest an IDS. By employing the BAT optimization algorithm during the ensemble clipping phase, the author offered their methodology for adopting the ensemble approach while taking into account numerous extreme learning machines [13]. This model works well in normal conditions with compromised accuracy. Some of the authors used metaheuristic algorithms like the whale optimization algorithm (WOA) with genetic algorithm (GA) for feature selection for sample-based classification [14], in this

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

proposed FAR was a major concern. The WILS framework deep learning (DL) has been proposed by the author. The proposed design employs deep learning models that are whale-optimized for the prediction of attacks in an IoT network.

The optimization techniques have improved the performance of NIDS, but still achieving effective NIDS with a high detection rate and low false alarm rates (FAR)is a challenging task. Traditionally optimization techniques focus on specific attack patterns while ignoring other traffic in the network, this results in less effective NIDS which will affect the accuracy of NIDS and increase the FAR [15]. In this paper, we propose an approach that integrates the efficiency of two optimization algorithms that is Simulated Annealing (SA) and Whale Optimization algorithm (WOA). In this proposed approach we use SA to improve the local optima problem which is a major problem with WOA, this results in better performance of NIDS in one frame. We use KNN to calculate the fitness of each selected feature. In this paper we also use deep learning techniques, to evaluate the accuracy of feature extraction and classification of anomaly and normal traffic. The proposed model is called hybridization of Whale Optimization Algorithm with Simulated Annealing (HWSA) [16]

3. PROPOSED FEATURE SELECTION TECHNIQUE USING HYBRIDIZATION OF WHALE OPTIMIZATION ALGORITHM WITH SIMULATED ANNEALING (HWSA)

In our proposed Whale Optimization Algorithm with Simulated Annealing (HWSA) model feature selection method, we combine the two effective optimization techniques Whale Optimization algorithm (WOA) and Simulated Annealing (SA) by using the advantages of both techniques. WOA gives good performance in the global exploration phase and whereas the SA performs optimal local optima problem, it can be used as fine-tuning. In this proposed approach we aim to reduce the number of redundant features from the dataset by maximizing the detection rate (accuracy) and minimizing FAR.

This WOA, which falls within the metaheuristic approach and was inspired by a humpback whale's collaborative hunting method, was mentioned by the author in 2016 [17]. The social behavior and movement patterns of these water creatures are imitated in this approach, to solve challenging optimization challenges. In WOA, the algorithm constantly improves possible solutions to identify the best or nearly the best outcome by representing solutions as potential solutions in the search space. A population of virtual whales that each represent a potential resolution inside the search space of an optimization problem makes up the basis of the WOA's operation. Utilizing the pattern of humpback whale behaviors encircling prey and bubble-net feeding the system repeatedly improves the solutions [18].

In the proposed work, to evaluate the fitness of these selected features K-Nearest Neighbors (KNN) function is used to solve the optimization problem. KNN is a simple yet powerful machine learning (ML) algorithm for classification purposes. It works on the concept that the data points with similar features are likely to belong to the same class. [19]. In KNN, the algorithm finds the "k" closest data points (neighbors) in the training dataset and gives the class label that is most common among those neighbors for a particular data point to be classified. [20]. The KNN classifier doesn't require any explicit

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

training phase and can handle binary or multi-class classification problems [21]. It's a non-parametric and instance-based algorithm, meaning it makes predictions based on the instances themselves rather than building a model. We use an objective function which is formulated using the KNN technique by using the data set chosen from $\boldsymbol{S}$ called $\overline{\boldsymbol{S}}$ in each iteration of the WOA. The resultant function obtained by using KNN is represented as $f_{knn}(\bar{s})$, where $\bar{s}$ is the row vector from the chosen features $\overline{\boldsymbol{S}} \subset \boldsymbol{S}$. The $\overline{\boldsymbol{S}}$ contains the selected columns of $\boldsymbol{S}$, according to the selection procedure using WOA. To initiate the selection process using WOA, we first define the matrix of a random variable with size $(w \times m)$ as $\boldsymbol{Z}$, where $\boldsymbol{Z}_{ij} \in \mathcal{U}(0,1)$. Here $\mathcal{U}(0,1)$ is the uniformly distributed random variable between 0 and 1. The various steps involved in WOA are given below.

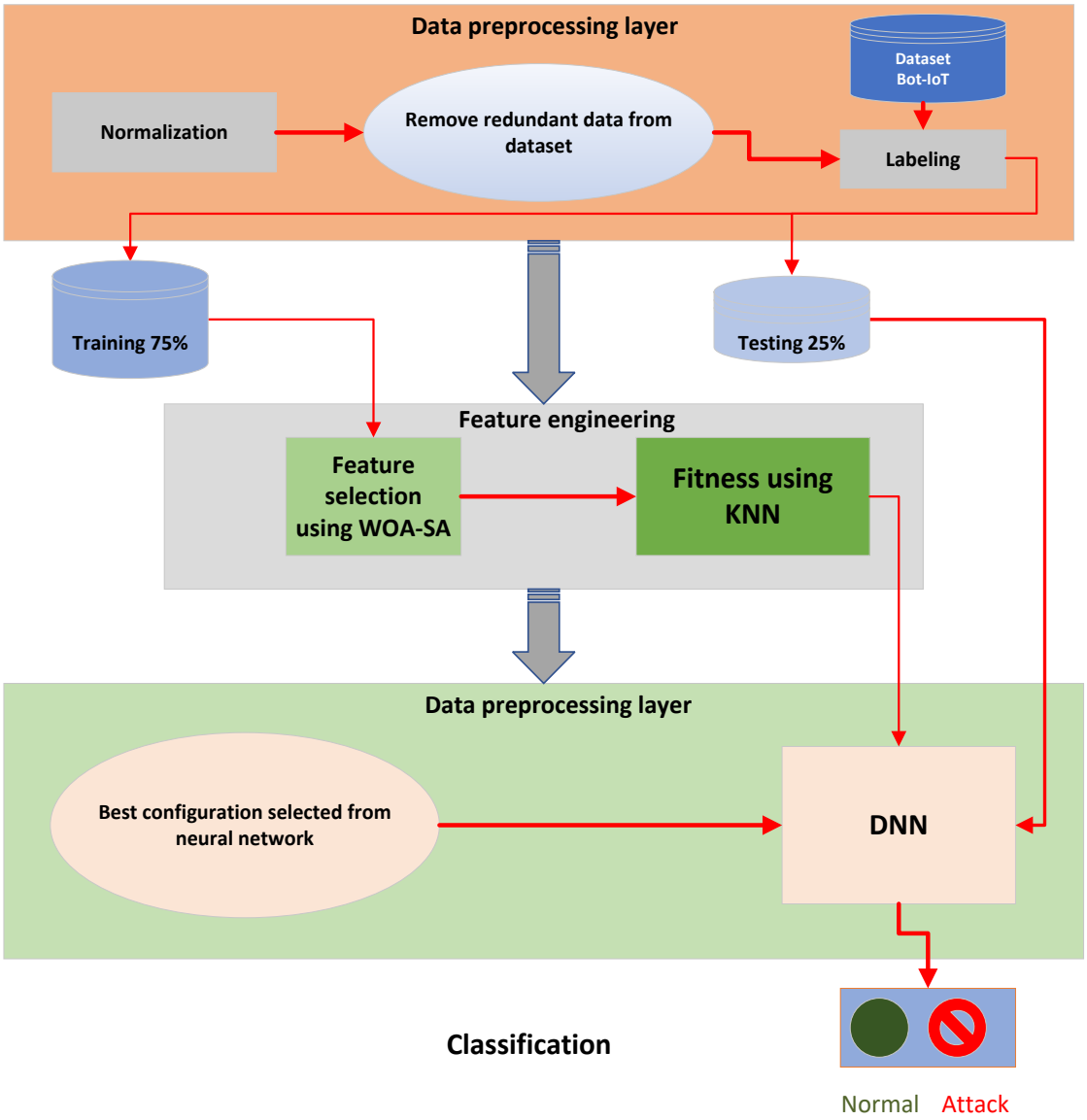


Figure 2: Proposed NIDS framework

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

Encircling prey: Humpback whales initially identify the location of the group of targets (pray) and then encircle them. The direction of the ideal hunting agent is given by the following procedure. The $(i,j)^{th}$ entry of $\boldsymbol{Z}$ is updated as follows:

$$\boldsymbol{Z}_{ij}(t+1) = \boldsymbol{z}_{sel}^{j}(t) - A * D \tag{4}$$

$$D = \left|C\boldsymbol{Z}_{n}^{j}(t) - \boldsymbol{Z}_{ij}(t+1)\right| \tag{5}$$

$\boldsymbol{z}_{sel}$best solution, $\boldsymbol{z}_{sel}^{j}(t)$ represents the j^{th}entry of $\boldsymbol{z}_{sel}(t)$ in the t^{th} iteration (or time step). The matrix $\boldsymbol{Z}$ represents the position for a single search space. *A and C* is coefficient Vectors that may be represented using the formula

$$A = 2ar - a \tag{6}$$

$$C = 2r \tag{7}$$

Whereas *a* is the random parameter that is decreasing in the range [2, 0] and $C \in \mathcal{U}(0,1)$.

Bubble net attacking Humpback whales exhibit a unique feeding behavior called "bubble-net attacking," where they coordinate to create a bubble net that traps and concentrates their prey. This particular strategy is described through the following equation [22]

$$\boldsymbol{Z}_{ij}(t+1) = E.\exp(bt).\cos(2\pi l) + \boldsymbol{z}_{sel}^{j}(t) \tag{8}$$

$$E = \boldsymbol{z}_{sel}^{j}(t) - \boldsymbol{Z}_{ij}(t) \tag{9}$$

E is a distinct vector component and the constant vector is given by b which is used to find the shape of spiral *l which* ranges from -1 to 1 which is some random vector, each with a probability of 50%, given as:

$$\boldsymbol{Z}_{ij}(t+1) = \begin{cases} \boldsymbol{z}_{sel}^{j}(t) - A * D & P < 0.5 \\ E.\exp(bt) * \cos(2\pi l) + \boldsymbol{z}_{sel}^{j}(t) & P \geq 0.5 \end{cases} \tag{10}$$

Where $P \in \mathcal{U}(0,1)$

Prey Search Phase In WOA, the exploration phase is the hunt for prey. In other words, this approach and |A|>1 allow whales to be examined worldwide while searching. This mathematical model is provided by

$$\boldsymbol{Z}_{ij}(t+1) = \boldsymbol{Z}_{hj} - A * D \tag{11}$$

$$D = \left|\boldsymbol{Z}_{hj} - \boldsymbol{Z}_{ij}\right| \tag{12}$$

$\boldsymbol{Z}_{hj}$ represents the position of a humpback whale, where h is the random integer chosen in the range $[1, w]$. In each iteration in WOA algorithm it is ensured that the values of $\widetilde{\boldsymbol{Z}}$ are ensured to be confined in the interval 0 and 1 as follows

$$if\ \boldsymbol{Z}_{ij}(t+1) > 1, \quad then\ \boldsymbol{Z}_{ij}(t+1) = 1 \tag{16}$$

$$if\ \boldsymbol{Z}_{ij}(t+1) < 0, \quad then\ \boldsymbol{Z}_{ij}(t+1) = 0 \tag{17}$$

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

At the end of each time step, we define a new matrix $\boldsymbol{V}$ **of** dimension $(w \times m)$ using the condition given as

$$if\ \boldsymbol{Z}_{ij}(t+1) > 0.5, \qquad then\ \boldsymbol{V}_{ij}(t+1) = 1 \tag{18}$$

$$if\ \boldsymbol{Z}_{ij}(t+1) < 0.5, \qquad then\ \boldsymbol{V}_{ij}(t+1) = 0 \tag{19}$$

Note that $\boldsymbol{V}$ now contains the entries either 0 or 1. Further an iterative procedure is carried out to evaluate the fitness function using KNN. In the l-th iteration, we consider the l-th row of the matrix $\boldsymbol{V}(t+1)$ denoted as $\boldsymbol{v}_l(t+1)$. Note that l varies from 1 to w. An index set defined as $\mathcal{J}(t+1)$ is created such that

$$if\ \boldsymbol{v}_{lj}(t+1) = 1, \qquad then\ j \in \mathcal{J}(t+1) \tag{20}$$

Here $\boldsymbol{v}_{lj}(t+1)$ is the j-th value in the l-th row of $\boldsymbol{V}$. According to the set $\mathcal{J}(t+1)$, the features from the $\boldsymbol{S}$ are selected to obtain $\overline{\boldsymbol{S}}_l(t+1)$, where $\overline{\boldsymbol{S}}_l(t+1)$ is the $\overline{\boldsymbol{S}}$ obtained in the l-th iteration of the $(t+1)$-th time step. The corresponding fitness value is evaluated for $f_{knn}(\overline{\boldsymbol{S}}_l(t+1))$ defined as fv_{knn}^{l}. Once all the fitness values are obtained, we calculate l_{sel} as

$$l_{sel} = index\left(\min\left(fv_{knn}^{l}\right)\right) \tag{21}$$

Where $index(\cdot)$ is the index of the fitness value and $\min(\cdot)$ is the minimum out of the all-fitness values. The l_{sel} is further used for calculating best solution $\boldsymbol{z}_{sel}(t+1)$ as

$$\tilde{\boldsymbol{z}}_{sel}(t+1) = \boldsymbol{Z}_{l_{sel}:}(t+1) \tag{22}$$

Note that $\boldsymbol{Z}_{l_{sel}:}$ is the l_{sel}-th row of $\boldsymbol{Z}(t+1)$. The procedure is repeated until the maximum number of time step is reached to obtain final selected index set $\mathcal{J}_{fin}$

Once $\tilde{\boldsymbol{z}}_{sel}(t+1)$ is obtained, then the $\tilde{\boldsymbol{z}}_{sel}(t+1)$ further refined using simulated annealing algorithm. This may help to further optimize the solution obtained in the time step $(t+1)$ and reduce the probability of trapping at local minima points.

In order to proceed, we first initialize the maximum temperature T_m. The initial candidate solution is given as $\boldsymbol{y}^0 = \tilde{\boldsymbol{z}}_{sel}(t+1)$. We first define $E^r = E_{t+1}(\boldsymbol{y}^r) = f_{knn}(\boldsymbol{y}^r)$. Here $\boldsymbol{y}^r$ is the candidate solution in r-th iteration in simulated annealing process. To start, the initial energy E^0 is calculated by using the aforementioned method. Also, the maximum and minimum temperature are defined as T_{max} and T_{min}, respectively. Similarly, the threshold value of E is also defined as E_{th}. Also by defining $\mu \in [0,1]$, the process is summarized in below pseudo code.

Pseudo code: Simulated Annealing Enhancement for WoA

***Input to SA**: $T \leftarrow T_{max}$ and $\boldsymbol{y}^0$*

***While** $(T > T_{max})$ and $(E^r > E_{th})$ **do** (*in r-th iteration*)*

$\boldsymbol{y}^r = \boldsymbol{y}^{r-1} + \mu\, rand(1,m)$

if $\boldsymbol{y}^r[:,:] > 1$, ***then*** $\boldsymbol{y}^r[:,:] = 1$ **; *if*** $\boldsymbol{y}^r[:,:] < 0$, ***then*** $\boldsymbol{y}^r[:,:] = 0$

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

Calculate: E^r

$$\Delta E^r = E^r - E^{r-1}$$

$$P(\Delta E^r, T) = exp(-\Delta E^r / T)$$

if $P(\Delta E^r, T) > \varepsilon \in \mathcal{U}(0,1)$ ***then*** accept $\boldsymbol{y}^r$ and E^r

else $\boldsymbol{y}^r = \boldsymbol{y}^{r-1}$ and $E^r = E^{r-1}$

$T \leftarrow T / \mu$

end

Output: $\boldsymbol{z}_{sel}(t+1) \leftarrow \boldsymbol{y}^r$

Once $\boldsymbol{z}_{sel}(t+1)$ is obtained, it is used in the further iteration in WOA and the process is continued.

3.1. Feature Selection Process for S-WOA

In the initial phase we preprocessed the data, then in the feature selection stage we created a population of "whales," where each whale corresponds to a feature subset. In order to find viable pairings, the whales scour the enormous feature space. In order to identify possible combinations, the whales explore the huge feature space. The S-WOA architecture uses WOA and Simulated Annealing (SA) to help in exploration. Due to the addition of temperature-controlled randomization in SA, the algorithm has the ability to avoid local optima and explore various feature subsets. We use the K-Nearest Neighbors (KNN) approach to evaluate the fitness of each feature subset. Using these chosen features, KNN evaluates NIDS performance and assigns a fitness score to each whale (feature subset), figure 2 shows the proposed NIDS framework.

3.2. Deep Nurel Network implementation

In the final step of our proposed model, the selected features are subjected to deep learning for testing, training, and validation. In our proposed model we evaluated the performance using stranded DL models like Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN) [40]. The DL model is tested with full features set and the features selected from S-WOA and good performance DL is called an optimized neural network (OptiNet) which is DNN with optimal features [41]. DNN model is trained in multiple layers which comes under the supervised learning approach [23]. The DNN used in this paper is based on the concept of a FFAN with multiple hidden layers to improve the extracted features to better capacity [24].

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

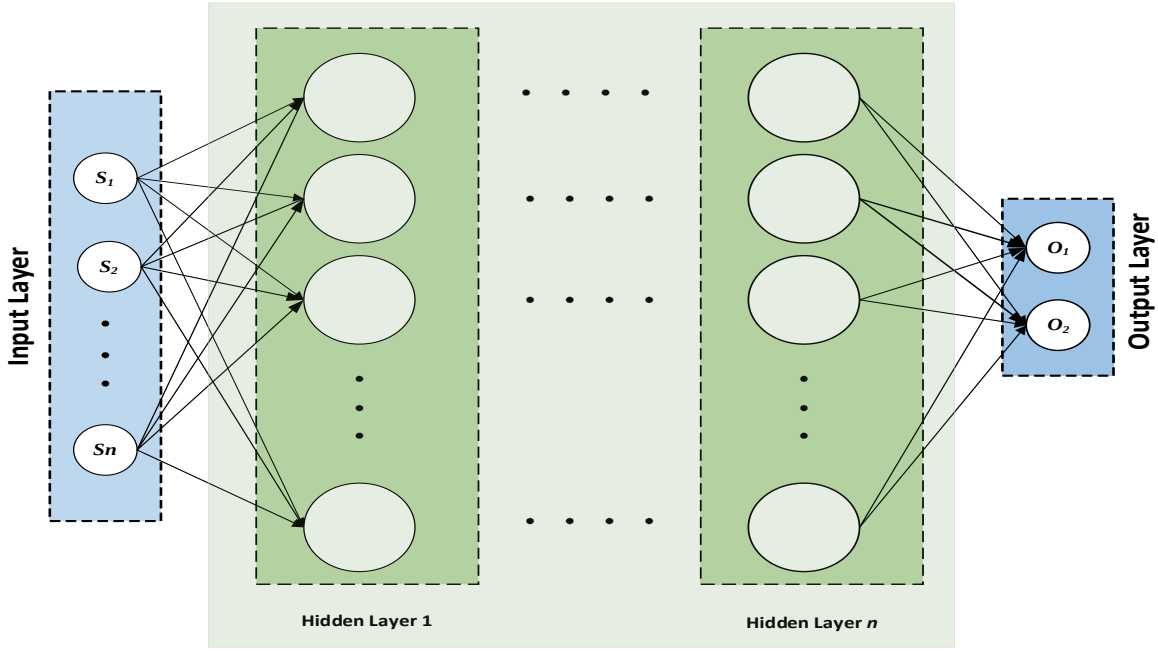


Figure 3: Deep learning framework

This proposed OptiNat (DNN) model consists of three dense layers which are the input layer, a hidden layer that includes the ReLU activation function, and use sigmoid activation as the output layer as shown in figure 3. $S = \{s_1, s_2, \ldots\ldots s_n\}$ is the selected features vector from the WOA-SA where n is number of optimized features from the whole dataset of 79, whereas $O = \{o_1, o_2\}$ is the output from the output layer, output vector consists of values 0 or 1 for classification of attack or normal. Mathematically, each hidden layer H_i output computation is represented as follows

$$H_i(S) = \sigma_i(W_i * S + B_i) \quad (5)$$

The weighted sum of the inputs from the layer before it S is represented as $W_i * S$ using the weight matrix $W_i.B_i$, which is added to the weighted total, represents the bias vector. $\sigma_i$is the activation function which is nonlinear.The weights, biases, and activation functions play important roles in showing the output of each hidden layer in a deep learning model, and above equation shows the mathematical representation which accurately represents the process. ReLU is an activation function that is commonly applied. It creates non-linearity by generating the input if it is positive and zero otherwise. It is represent as shown in equation below

$$f(S) = max(0, S) \quad (6)$$

Output of each neuron is given by $f(S)$ and s is the weighted sum of inputs matrix. Since it constricts the output within the range [0, 1], sigmoid is used in the output layer for binary classification because of its better-predicting probabilities [44]. According to mathematics, it is given as

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

$$f(s) = \frac{1}{1 + exp(-s)} \tag{7}$$

4. IMPLEMENTATION

The dataset used evaluation metrics, experimental setup, and results analysis are presented in this section. By applying the optical features to this model OptiNet, we evaluate the performance matrix parameters like accuracy, FAR, FNR and overall performance of NIDS

4.1. Dataset

We made use of the IoT-Botnet 2020 dataset, which is freely available to evaluate the performance of the deep learning approaches taken into consideration in this proposed work. This dataset is available in CSV format, which was created using PCAP files from the BoT-IoT dataset [8] . The focus of IoT-Botnet 2020 has been expanded to cover more network- and flow-based properties. The initial BoT-IoT dataset comprises samples that include different attacks, including DoS, DDoS, Reconnaissance, and information theft attacks. In this study we picked the normal entries from the dataset from the original dataset for usual network activity, in order to give an equal evaluation of our models, we also created the anomaly class by randomly selecting from each anomalous category. This dataset contains a total of 80 features, in this study, we use 79 characteristics, with the last one serving as a label for binary classification. However, as a binary class, we concentrate on finding abnormalities. features Table 1 provides the anomaly and benign distribution of the data set,

Table 1: Dataset Composition

Dataset	Benign Samples	Anomaly Samples	Total
This study	30000	30000	60000
Training (80%)	24090	23982	48072
Testing (20%)	5910	6018	11928

4.2. Data Preprocessing

To make the dataset ready for model training, data preparation is an important step. Following are some of the important parameters in data processing:

Data Loading: In this phase standard dataset is loaded in raw format in our study we are using the IoT-Botnet 2020 dataset this dataset is available in .CSV format.

Feature Standardization: We standardize the input features in order to keep consistency and minimize the effect of feature dimensions. The process of standardization involves modifying the data to have a mean of 0 and a standard deviation of 1. It improves model convergence by doing this.

Feature Extraction: Many features, such as network- and flow-based features, are present in the dataset. This phase is helpful in reducing dimensionality and focusing on those features that have the most influence on identifying incursions.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

4.3. Defining Hyperparameters

In order to properly train deep learning models, hyperparameters are crucial. Using the IoT-Botnet 2020 dataset, these hyperparameters are carefully chosen to guarantee efficient training and the best performance of our deep learning models in identifying network intrusions [25], [26]. Training is conducted for 10 epochs. Here, we explain the hyperparameters that we employed in our tests

Batch Size: A batch size of 27 has been selected. It specifies how many samples will be utilized in each training iteration. While a smaller batch size may result in faster convergence, more iterations can be required.

Learning Rate: We use the learning rate as 0.01, this regulates the optimization step size. For the model to converge successfully, a suitable learning rate is required.

Optimizer: We used Adam optimizer in this study, it is a well-known optimization technique that may be used for a range of deep-learning problems because it can adjust the learning rate while training.

Loss Function: Binary cross-entropy is used as the loss function in this case. For binary classification issues like intrusion detection, this is a common approach.

Activation Functions: We use the Rectified Linear Unit (ReLU) activation function for hidden layers and the Sigmoid activation function for the output layer and in our deep learning models.

4.4. Evaluation matrix

The accuracy, precision, recall, F1-Score, and false alarm rate metrics are used to assess the effectiveness of the examined ML classifiers. The various attributes of the confusion matrix shown in Table 2 constitute the foundation for these assessment criteria. The true positive (TP) and true negative (TN) examples in the confusion matrix show the correct attack prediction and normal circumstances, respectively. Like normal and assault situations, false negative (FN) and false positive (FP) occurrences are likewise inaccurate classifier predictions[26].

Table 2: Confusion Matrix

		Prediction	
		Attack	**Normal**
Attack	Attack	TP	FN
	Normal	FP	TN

5. RESULTS AND DISCUSSION

We use common deep learning models for benchmarking and comparative analysis, such as a traditional DNN, CNN, RNN, and an LSTM approach. While the CNN has a 1D convolutional layer followed by smoothing and dense layers, the DNN has two dense layers. The RNN and LSTM models have an output layer after the first hidden layer and are built for sequence data. For the purpose of model evaluation and validation, the dataset is divided into training and testing sets using an 80-20 ratio.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

5.1. Set up for Experiment

The experimental setting for our proposed research is created using a cloud-hosted version of Jupyter Notebook, namely Google Colab [27]. The specification of the experimental setup parameter is given table.

Table 3: specification for experiment

Parameters	Specification
Programming language	MATLAB, Python
Processor Model	13th Gen Intel(R)
CPU	Core (TM) i7-1360P 2.20 GHz
RAM	16 gigabytes
Temporary memory (Cache)size	56320KB
No. of cores in CPU	1
Windows	11

5.2. Evaluation and Comparison

Results with full dataset*:* Here we train the DL with full features and the performance evaluation of each is given in Table 3. It provides an overview of the results of the performance evaluation of DL techniques using the full features of the original dataset. Accuracy, precision, recall, and f1 metrics scores will be determined as part of the evaluation. The accuracy with full features was found to be for DNN is 92.1%. These results show the way the models perform when all available features are used. Despite their fair performance, it is important to draw attention to the compromises made, particularly with respect to false alarm rates and computational complexity. Figure 4 shows the performance matric parameters like accuracy, precession recall and F1Scoer for different DL approaches such as LSTM, RNN, CNN and DNN it is tested and trained of full dataset. We noticed that the accuracy is compromised with high computational complexity to process all the features in the dataset and then we applied the optimization S-WOA or feature selection

Table 4: Results with the full dataset

DL Model	Accuracy	Precision	Recall	F1 Score
LSTM	0.5775	0.5655	0.7038	0.6271
RNN	0.9273	0.8899	0.9767	0.9313
CNN	0.9345	0.9376	0.9521	0.9248
DNN	0.9772	0.9726	0.9965	0.978

Results with optimal features: Here we provide the results from the proposed feature selection method, which were carefully chosen using the feature selection method known as Simulated Whale Optimization (S-WOA). It effectively identifies and selects the correct feature set that would improve the performance of our NIDS using an iterative and combinational optimization approach that was inspired by the hunting behavior of whales. The feature set's dimensionality was therefore substantially decreased. The feature selection based upon WOA-SA was performed on a full dataset with population size 10 and maximum of 100 iterations, convergence curve with S-WOA is shown in figure 4, it can be noticed from that the convergence rate is very sharp and still reduced after 10th

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

iteration. From this we can reduce the computation complexity by reducing the number of iterations from 100 to 25. Around 25% computation capacity be reduced with the proposed optimization approach

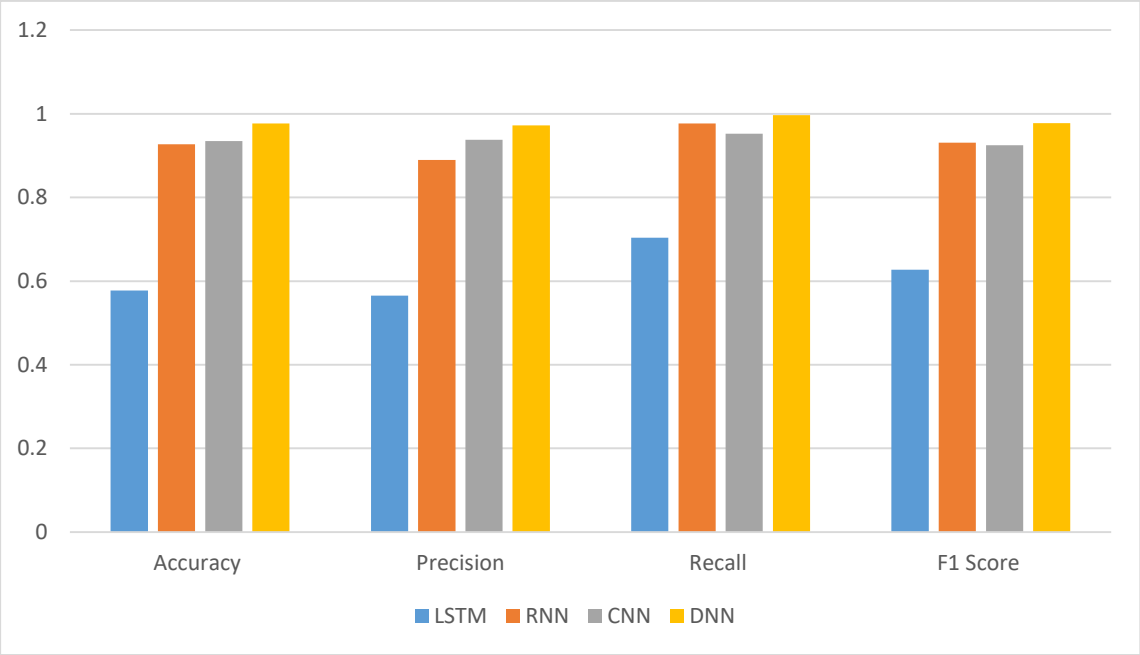
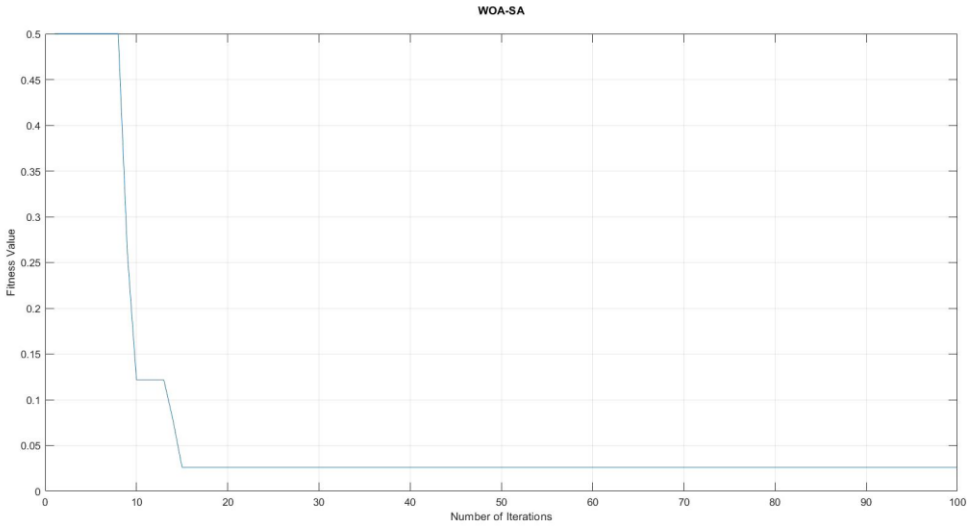


Figure 4: Performance matrix comparison for full future set



.

Figure 5: convergence curve for S-WOA

This curve gives the details of the proposed model and is searching for more informative features subset from the full dataset. As we can notice from the figure 5 the converging rate is very low that results in less computation complexity of the proposed model. The feature subset improved more gradually after an initial phase of fast convergence, which

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

was seen in the convergence curve. This behavior was similar to the collective hunting behavior of whales, which first gathers around prey before perfecting their encircling technique for better performance. The convergence curve explains the algorithm's effectiveness in searching different areas of the feature space and converging towards the ideal feature set. The selection of features with this proposed approach shows a good convergence rate.

Table 5: Selected features from S-WOA

Feature Name			
Src_Port	*Protocol*	*TotLen_Bwd_Pkts*	*Fwd_Pkt_Len_Mean*
Fwd_IAT_Tot	*Fwd_IAT_Mean*	*Bwd_IAT_Std*	*Bwd_IAT_Max*
Fwd_Header_Len	*RST_Flag_Cnt*	*Fwd_Act_Data_Pkts*	*Idle_Mean*
Idle_Std			

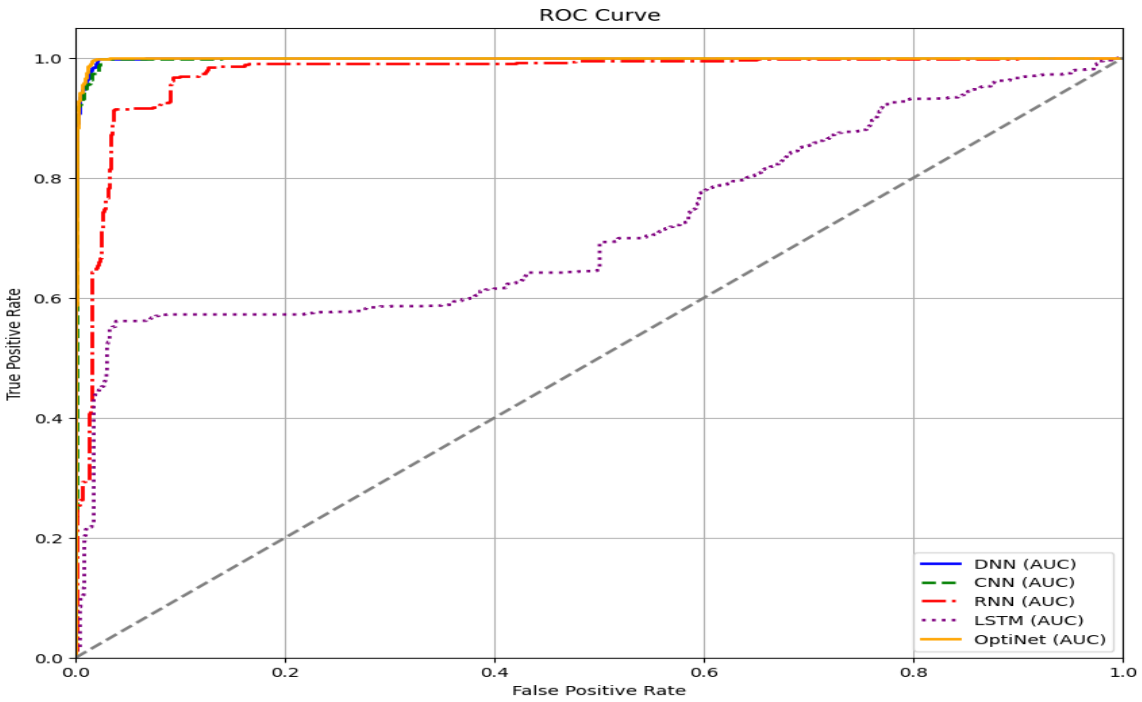


Figure 6: ROC curve for comparison

Table 6: Performance matric for the reduced feature set

DL Model	Accuracy	Precision	Recall	F1 Score
LSTM	0.6775	0.755	0.7038	0.782
RNN	0.9273	0.8899	0.9767	0.9313
CNN	0.9845	0.9776	0.9921	0.9848
DNN(OptiNet)	0.9978	0.9886	0.9975	0.988

The initial dataset, which had 79 features, was applied to the WOA-SA feature selection technique. By doing this, we were able to narrow down the feature set to 13 optimal features that were carefully selected to improve model performance. The mostly

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

correlated and redundant features were discarded and the resultant features are given in Table 5. By using this selected optimal feature, we train our deep learning model and we train our model for all traditional DL methods. We noticed from table 4 DNN outperform well when compared to other DL method, selected features are trained and tested with a DNN model named it as OptiNet.

The ROC curve used our study for evaluating the efficiency of the proposed NIDS model. Higher accuracy and steeper curve climb results in considerable gains in model selection and performance, especially when using optimum features. Figure 5 gives the ROC curve for all DL and proposed DL OptiNet. From the figure it can be noticed that OptiNet outperforms the other traditional DL methodology. Table 6 gives the evaluation metrics scores for selected features from the dataset using S-WOA. From this table 6 it can be noticed that DNN has a full feature set and optimal features. Figure 7 shows the performance matric comparison for different DL models for selected feature set which is 13 after S-WOA.

Table 7 gives a detailed comparison of performance matrix parameters for full and selected features from the dataset. Table 8 displays the effectiveness of the various DL-based NIDS approaches in terms of FAR and FNR. As was already said, the high FAR is the fundamental issue with the existing NIDS. In order to do this, it was found that the proposed OptiNet approach had a very low FAR and FNR in contrast to other techniques. The suggested method obtained a false alarm of 2.2% while also obtaining extremely high detection accuracy. The LSTM approach is seen to be having a high false alarm of 55.12%, demonstrating its inability to successfully acquire good patterns to categorize network traffic. Figure 8 shows the comparison of FAR and FNR for different DL models and compared to our proposed approach.

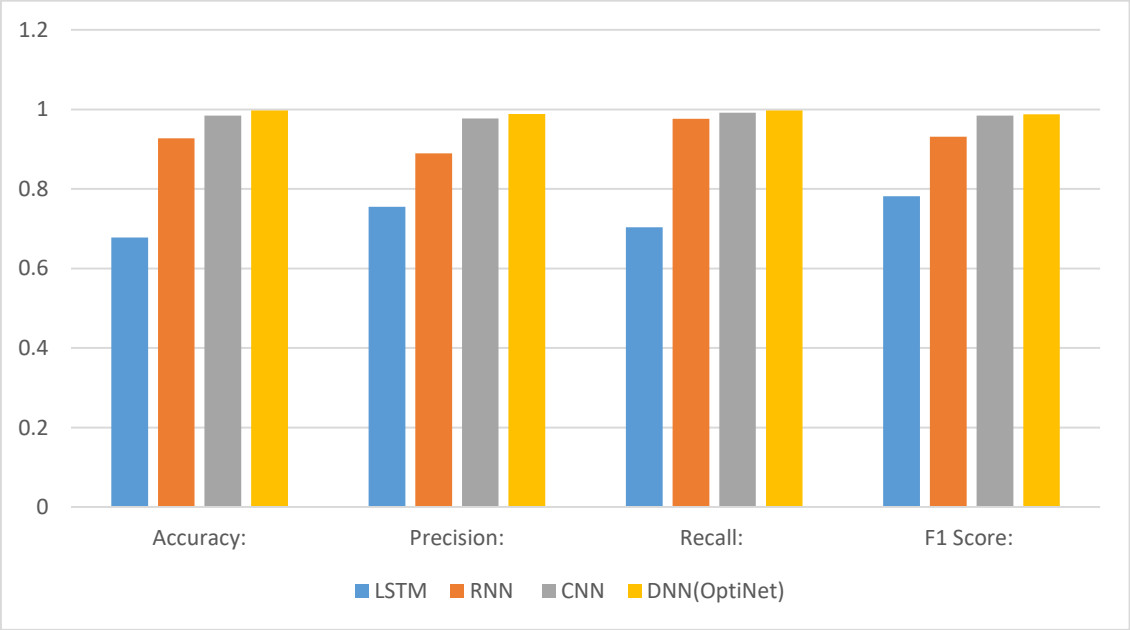


Figure 7: Performance matrix comparison for full future set

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

Table 7: Comparison DNN model

DL Model	Accuracy	Precision	Recall	F1 Score
DNN with full feature set	0.9772	0.9726	0.9965	0.978
DNN with optimal feature (OptiNet)	0.9978	0.9886	0.9975	0.988

Table 8: FAR and FNR for all model

Model	FAR	FNR
LSTM	0.5512	0.3056
RNN	0.1232	0.0069
CNN	0.0232	0.005
DNN	0.0222	0.004
OptiNet	0.022	0.0035

The feature selection procedure considerably increased model performance in terms of accuracy, precision, recall, and F1 score, among other metrics as shown in Figure 7. Reduced feature dimensionality improved model implementation, which in turn reduced overfitting. When working with high-dimensional datasets, this is very beneficial. We notice from this figure 9 that model accuracy for zero-day attack, with selected features that is 13 features it was increased to 2% and also decreased in FAR and FNR by about 10%.

Figure 10 shows the confusion matrix for all DL model, here we observed that there is miss classification problem with full features set of 79 features. Where Figure 11 gives the confusion matrix for optimal 13 features from the dataset which get selected in feature selection approach OptiNet. As we noticed from both figure 10 anomalies are detected with more accuracy for full feature set compared to optimal features with DNN and LSTM performance was found to be worse when compared to other methods and also more miss classification. When comparing optimal features of the OptiNet matrix there is negatable difference when compared to DNN.

For an IoT network, this study offers a thorough comparative analysis of several DL-based NIDS. The WOA-SA is used for feature selection to optimize the NIDS. We first perform the experiment on a full feature set with all 79 features and check the perform with traditional DL models. We noticed DNN outperformed well compared to other LSTM, RNN and CNN techniques, whereas LSTM was the worst among them accuracy about 97.7% was achieved with LSTM. FAR and FNR was also considerably less compared to other techniques to achieve this we need more computation resources.

The dataset results in a reduced feature set. With this feature selection method, we achieved Maximizing the detection of network intrusions while minimizing false alarms. From this feature selection method, we extracted the most relevant features that contribute significantly to detection accuracy and also result in enhancing the overall efficiency and effectiveness of NIDS configuration. With WOA-SA we achieved a good convergence curve and features were reduced to 13. Now we apply DNN on this reduced feature subset. With the experiment, we found the DNN achieved 100% outcomes, and the trial findings demonstrated a considerable improvement in detection accuracy of

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

about 99.78%. However, this study focuses only on binary classification for the BoT-IoT 2020 dataset, in future work, we will design the multiclass classification model for different types of datasets

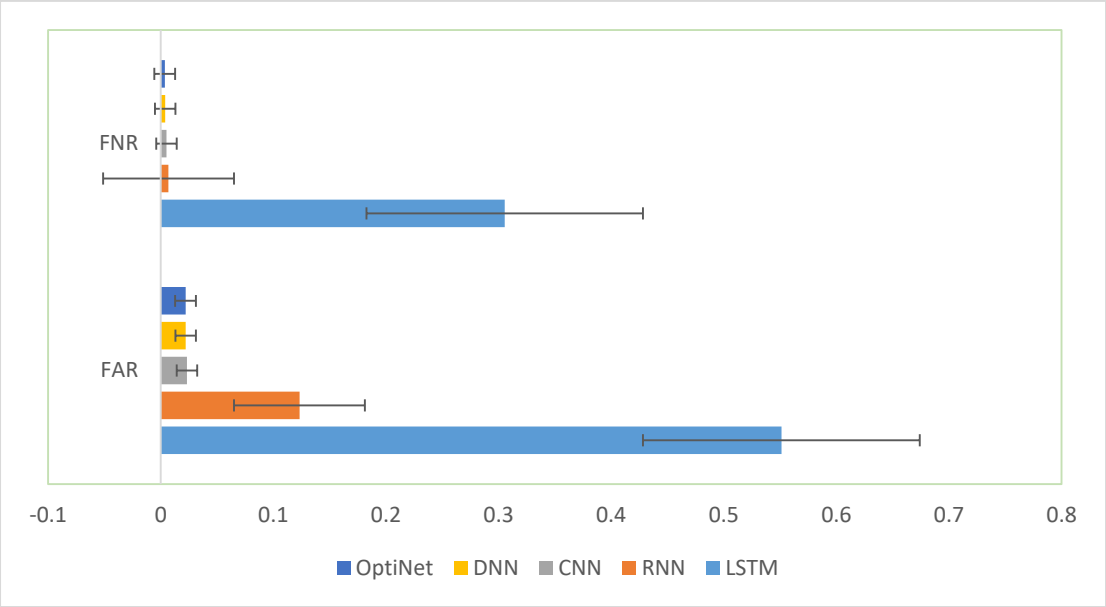


Figure 8: FAR and FNR

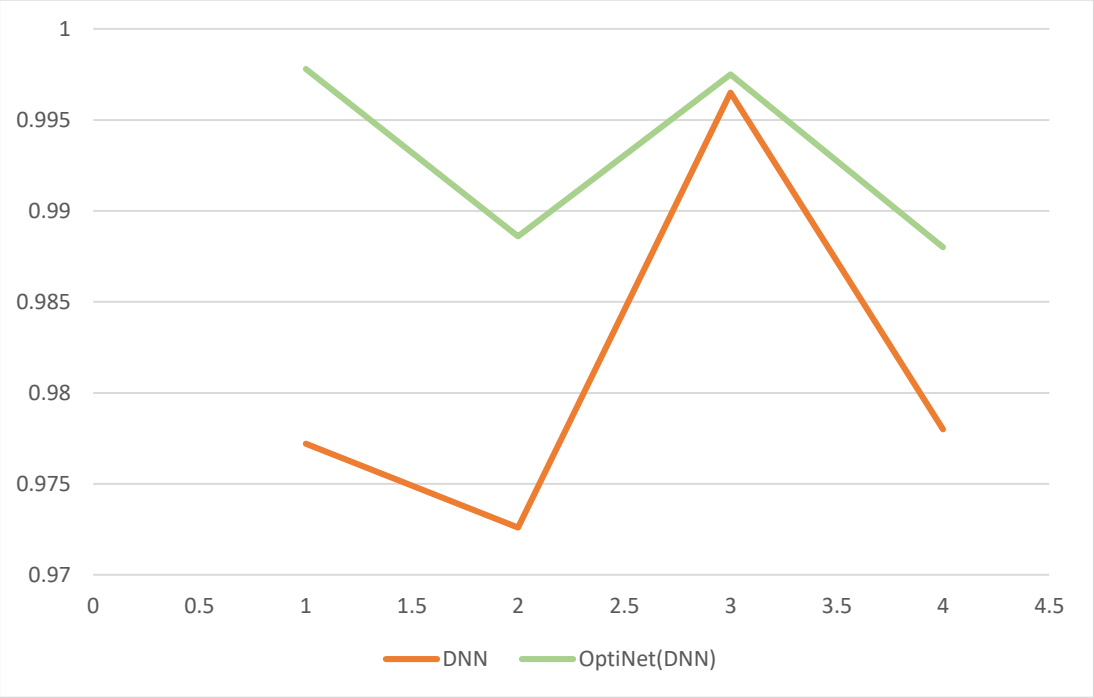


Figure 9: Comparison of proposed model with DNN

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

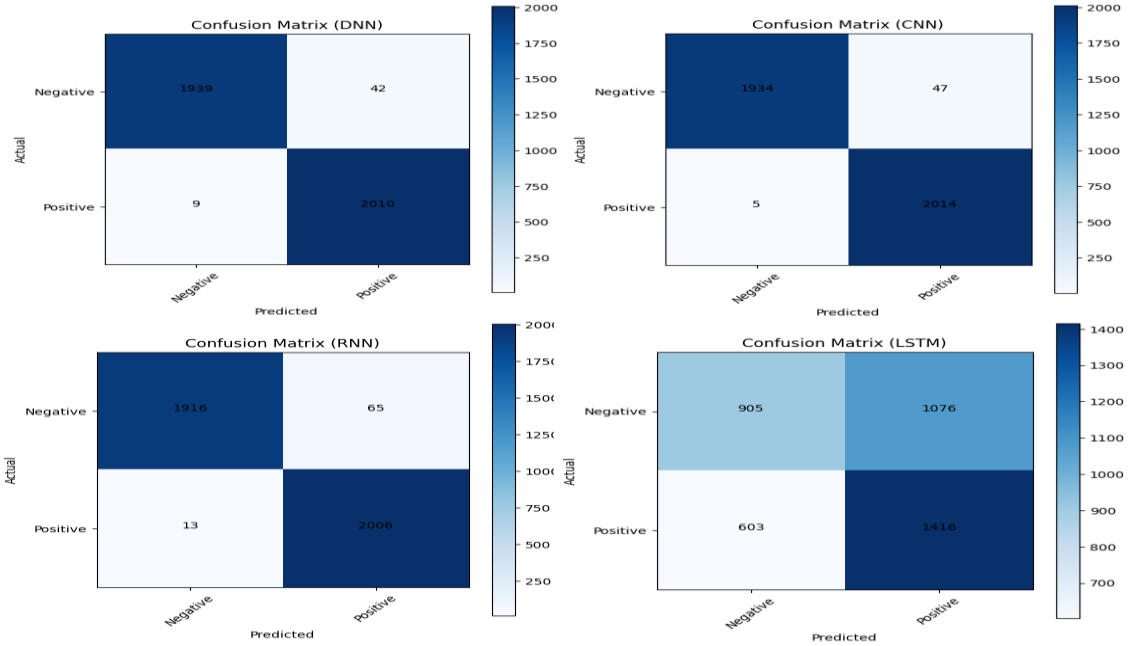


Figure 10: Confusion matrix for full feature set

5.3. Evaluation of OptiNet

We carried out a comprehensive comparison analysis against standard benchmark techniques and relevant earlier studies in the field of Network Intrusion Detection Systems (NIDS) in order to evaluate the performance of our suggested WOA-SA approach. Three benchmark paper used to evaluate the performance of our proposed method is, ML-PSO [28], RSA-CNN, HFS-ANN [29], Deep-RNN [23].

Table 9: Benchmark Techniques and Comparative Performance Metrics

Methodology	Performance evaluation metric					
	Accuracy	Precision	Recall	F1 Score	FAR	FNR
ML-PSO	99.32	99.26	99.37	99.31	0.62	0.74
AE-LSTM	98.726	98.874	98.814	98.827	0.327	1.186
HFS-ANN	98.8	98.9	98.9	98.8	0.013	1.2
Deep-RNN	96.208	96.689	95.161	95.788	0.976	4.839
OptiNet	99.78	9886	99.75	98.8	0.022	0.0035

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

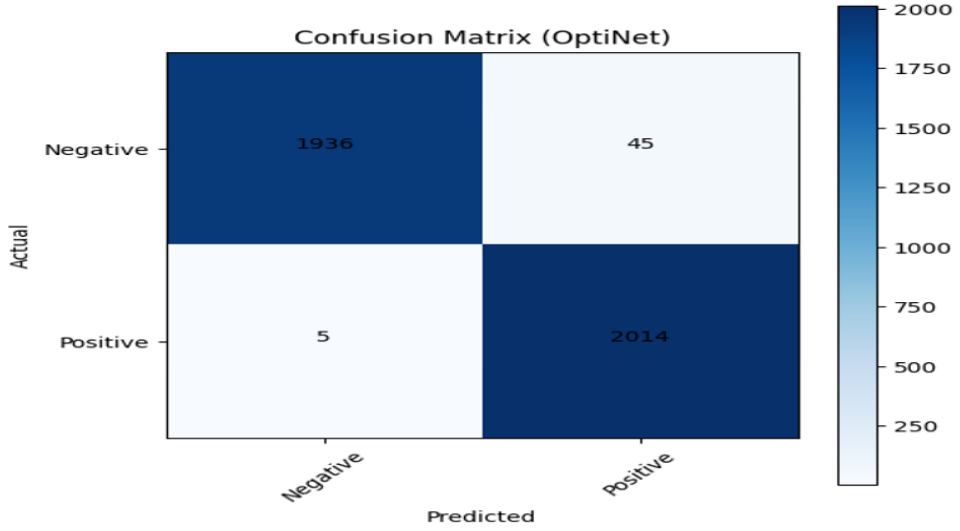


Figure 11: Confusion matrix for full feature set

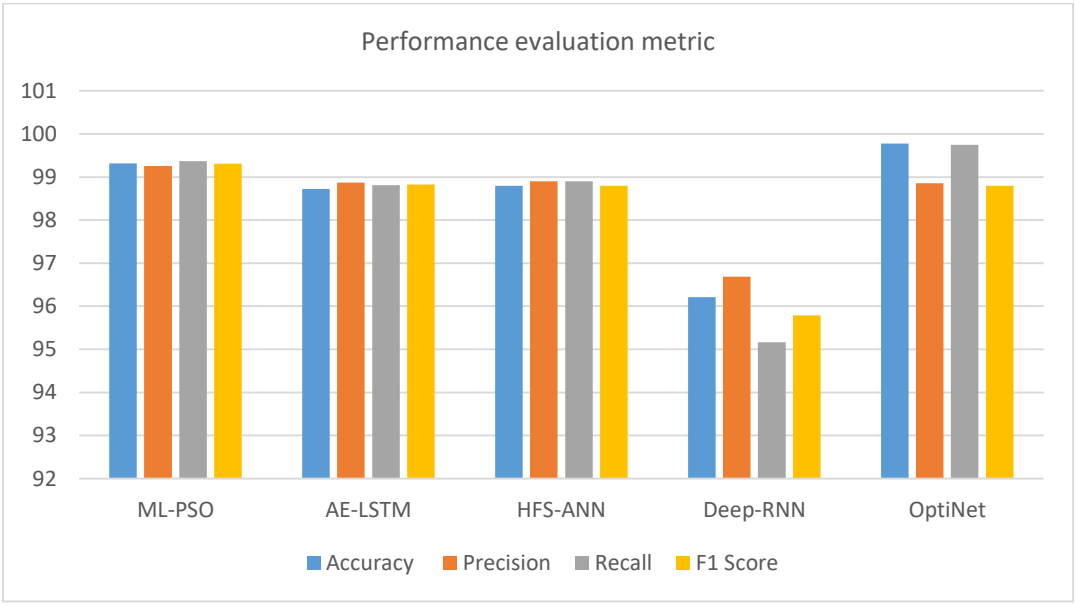


Figure 12: Comparative Performance Graph

Our OptiNet technique significantly perform batter then the selected baseline methods across numerous measures, as shown in Table 9 and Figure 12. The outcomes show how our approach reduced False Alarm Rates (FAR), enhanced feature extraction accuracy, enhanced detection rates. These results support the idea that, in comparison to current approaches, the suggested WOA-SA method offers significant progress in the field of NIDS. Our technique's ability to [list any unique benefits or strengths that the comparison highlights] shows how strong it is. This presents a strong case for

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

implementing the WOA-SA approach to increase the efficacy of network intrusion detection systems.

6. CONCLUSION

In this study, we introduced a novel deep learning model called OptiNet, which integrates feature selection techniques, specifically the Whale Optimization algorithm (WOA) and Simulated Annealing (SA), into the deep learning approach. Our approach aims to enhance the feature selection process and improve the overall performance of deep neural networks. We conducted a comprehensive evaluation of OptiNet with optimal features and compared it to a full feature set with four standard deep learning models, including DNN, CNN, RNN, and LSTM, on a binary classification task using the BoT-IoT dataset.

The combination of WOA-SA approaches for feature selection improves model performance while also allowing for the selection of more relevant features, hence reducing the dataset's dimensionality and increasing model efficacy, we reduced the number of features from 79 to 13. Our experimental results show that OptiNet with optimal features shows good accuracy that is 99.78% when compared to the traditional models. In order to fully evaluate the model performance, we also evaluate them using additional metrics such as accuracy, recall, F1 score, False Alarm Rate (FAR) and False Negative rate (FNR). The comparison analysis showed that OptiNet outperforms the other models in terms of precision, recall, and F1 score while maintaining good accuracy in detecting anomaly in IoT network.

References

1) M. S. Habeeb and T. R. Babu, “A Two-Phase Feature Selection Technique using Information Gain and XGBoost-RFE for NIDS,” International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 13s, pp. 278–287, Jan. 2024, Accessed: Feb. 02, 2024. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/4595

2) R. Kohavi and G. H. John, “Wrappers for feature subset selection,” Artif Intell, vol. 97, no. 1–2, pp. 273–324, Dec. 1997, doi: 10.1016/S0004-3702(97)00043-X.

3) A. S. Khan, Z. Ahmad, J. Abdullah, and F. Ahmad, “A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network,” IEEE Access, vol. 9, pp. 87079–87093, 2021, doi: 10.1109/ACCESS.2021.3088149.

4) Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, Jan. 2021, doi: 10.1002/ETT.4150.

5) M. S. Habeeb and | T Ranga Babu, “Network intrusion detection system: A survey on artificial intelligence-based techniques,” 2022, doi: 10.1111/exsy.13066.

6) Ullah and Q. H. Mahmoud, “A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks,” Conf Proc IEEE Int Conf Syst Man Cybern, vol. 2020-October, pp. 134–140, Oct. 2020, doi: 10.1109/SMC42975.2020.9283220.

ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

7) A. Rashid, M. J. Siddique, and S. M. Ahmed, “Machine and Deep Learning Based Comparative Analysis Using Hybrid Approaches for Intrusion Detection System,” 3rd International Conference on Advancements in Computational Sciences, ICACS 2020, Feb. 2020, doi: 10.1109/ICACS47775.2020.9055946.

8) N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,” Future Generation Computer Systems, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/J.FUTURE.2019.05.041.

9) Albulayhi, Q. A. Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, “IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method,” Applied Sciences 2022, Vol. 12, Page 5015, vol. 12, no. 10, p. 5015, May 2022, doi: 10.3390/APP12105015.

10) B. Xu, L. Sun, X. Mao, R. Ding, and C. Liu, “IoT Intrusion Detection System Based on Machine Learning,” Electronics 2023, Vol. 12, Page 4289, vol. 12, no. 20, p. 4289, Oct. 2023, doi: 10.3390/ELECTRONICS12204289.

11) Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review,” Data Classification: Algorithms and Applications, pp. 37–64, Jan. 2014, doi: 10.1201/B17320.

12) P. Bhale, S. Biswas, and S. Nandi, “ML for IEEE 802.15. 4e/TSCH: Energy Efficient Approach to Detect DDoS Attack Using Machine Learning,” 2021 International Wireless Communications and Mobile Computing, IWCMC 2021, pp. 1477–1482, 2021, doi: 10.1109/IWCMC51323.2021.9498637.

13) X. (Shane) Wang, J. H. (Joseph) Ryoo, N. Bendle, and P. K. Kopalle, “The role of machine learning analytics and metrics in retailing research,” Journal of Retailing, vol. 97, no. 4, pp. 658–675, Dec. 2021, doi: 10.1016/J.JRETAI.2020.12.001.

14) R. Vijayanand and D. Devaraj, “A Novel Feature Selection Method Using Whale Optimization Algorithm and Genetic Operators for Intrusion Detection System in Wireless Mesh Network,” IEEE Access, vol. 8, pp. 56847–56854, 2020, doi: 10.1109/ACCESS.2020.2978035.

15) J. Chen, S. Yuan, D. Lv, and Y. Xiang, “A novel self-learning feature selection approach based on feature attributions,” Expert Syst Appl, vol. 183, p. 115219, Nov. 2021, doi: 10.1016/J.ESWA.2021.115219.

16) H. Liu and H. Motoda, “Feature Selection for Knowledge Discovery and Data Mining,” Feature Selection for Knowledge Discovery and Data Mining, 1998, doi: 10.1007/978-1-4615-5689-3.

17) S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” Advances in Engineering Software, vol. 95, pp. 51–67, May 2016, doi: 10.1016/J.ADVENGSOFT.2016.01.008.

18) A. Kaveh and M. I. Ghazaan, “Enhanced whale optimization algorithm for sizing optimization of skeletal structures,” https://doi.org/10.1080/15397734.2016.1213639, vol. 45, no. 3, pp. 345–362, Jul. 2016, doi: 10.1080/15397734.2016.1213639.

19) G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN model-based approach in classification,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2888, pp. 986–996, 2003, doi: 10.1007/978-3-540-39964-3_62/COVER.

20) S. Habeeb and T. R. Babu, “Network intrusion detection system: A survey on artificial intelligence-based techniques,” Expert Syst, vol. 39, no. 9, p. e13066, Nov. 2022, doi: 10.1111/EXSY.13066.

21) S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, “Learning k for kNN Classification,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8, no. 3, Jan. 2017, doi: 10.1145/2990508.

22) S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” Advances in Engineering Software, vol. 95, pp. 51–67, May 2016, doi: 10.1016/J.ADVENGSOFT.2016.01.008.

ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 03-2024
DOI: 10.5281/zenodo.10851842

23) Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for IoT intrusion detection system,” Simul Model Pract Theory, vol. 101, p. 102031, May 2020, doi: 10.1016/J.SIMPAT.2019.102031.

24) Gheisari, G. Wang, and M. Z. A. Bhuiyan, “A Survey on Deep Learning in Big Data,” Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, vol. 2, pp. 173–180, Aug. 2017, doi: 10.1109/CSE-EUC.2017.215.

25) Bhale, S. Biswas, and S. Nandi, “Effective injection of adversarial botnet attacks in IoT ecosystem using evolutionary computing,” Internet Technology Letters, vol. 6, no. 4, p. e433, Jul. 2023, doi: 10.1002/ITL2.433.

26) Z. Ahmad et al., “S-ADS: Spectrogram Image-based Anomaly Detection System for IoT networks,” Proceedings - AiIC 2022: 2022 Applied Informatics International Conference: Digital Innovation in Applied Informatics during the Pandemic, pp. 105–110, 2022, doi: 10.1109/AIIC54368.2022.9914599.

27) E. Bisong, “Google Colaboratory,” Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64, 2019, doi: 10.1007/978-1-4842-4470-8_7.

28) N. Kunhare, R. Tiwari, and J. Dhar, “Particle swarm optimization and feature selection for intrusion detection system,” Sadhana - Academy Proceedings in Engineering Sciences, vol. 45, no. 1, pp. 1–14, Dec. 2020, doi: 10.1007/S12046-020-1308-5/FIGURES/12.

29) F. E. Ayo, S. O. Folorunso, A. A. Abayomi-Alli, A. O. Adekunle, and J. B. Awotunde, “Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection,” Information Security Journal: A Global Perspective, vol. 29, no. 6, pp. 267–283, Nov. 2020, doi: 10.1080/19393555.2020.1767240.