Journal of Theoretical and Applied Information Technology
15th August 2024. Vol.102. No. 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL ADAPTIVE NETWORK INTRUSION PREVENTION SYSTEM FOR INTERNET OF THINGS

[1]PARTHIBAN ARAVAMUDHAN, [2]DR.T. KANIMOZHI

[1]Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India.
[2]Electronics and Communication Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India.

ABSTRACT

Data traffic has significantly increased as a result of recent advancements in computer networks and other related services. At the same time, the negative effects of cyber-attacks have become far more prevalent. Variety of network threats which are increasing posing more difficulties and challenges than ever before. To overcome this issue, anomaly prevention and signature-based prevention are two key effective approaches for preventing the cyber attacks. Even though these two key approaches have a potential for success, but still it is like a "Cloud on the Horizon". Anomaly-based prevention has the potential for high false positives, while Signature-based prevention is more vulnerable to zero-day attacks. To address this concern, this paper explores the importance in developing an Intrusion Prevention System (IPS) model using Deep Learning (DL) algorithm. To evaluate the performance, NIDS dataset V.10 2017 is utilized. The Z-score normalization technique is used to clean the dataset, in which the scaling approach is applied to remove outlier's. Adaptive Principal Component Analysis (A-PCA) and Linear Discriminant Analysis (LDA) are deployed for dimensionality reduction and class separation optimization, respectively, resulting in a standardized dataset that accelerates prevention efforts by reducing processing latency and enhancing filtering effectiveness. The Whale Optimization Algorithm (WOA) is implemented to find the best optimal values using the "Rule of Convergence" method. The converged data are identified and trained for feature extraction to build an efficient optimized Convolutional Neural Network (CNN) model. "Re-Routing" concept is introduced, which identifies the threshold values during the hyper-parameter tuning. The primary cause for implementing this idea is to reduce the training cost, time consumption and also to enhance the prevention mechanism. The proposed NIPS model is tested and compared with other existing ML and DL methods. The simulation results affirm that the Optimized Rerouted Convolutional Neural Network (OR-CNN) model outperforms the other avant techniques in terms of time consumption, performance and accuracy.

Keywords: *Network Intrusion Prevention System (NIPS), Convolutional Neural Network (CNN), A-PCA, LDA,WOA, Re-Routing.*

1. INTRODUCTION

Recent years have seen a sharp rise in network efficiency and overall data traffic due to the quick development of Information Technology. Regrettably, the increase in cyber-attacks has resulted in increased harm, not only because of the rise in occurrences but also because of the advanced and commencement of multiple variations of these attacks. In today's landscape, fully safeguarding a network against malicious hackers is exceedingly challenging [1].One of the most dangerous types of cybercrimes is a zero-day attack, which takes use of a freshly found vulnerability in network systems before a solution is created. Network defenselessness is inevitable until a solution is created and implemented into the system [2]. A zero-day attack can last more than 2 years, although according to previous research, it can prolong up to more than 300 days. The occurrence of zero day attacks could increase furthermore, if any new vulnerability is revealed [3].

To discern specific patterns within incoming packets, contemporary network security systems primarily utilize signature-based techniques for real-time attack prevention. These methods bear resemblance to virus scanners [4] – [6].Achieving real-time detection necessitates in-line processing algorithms capable of identifying attacks at the line rate. While distributed systems can enhance detection speed, they often face costly synchronization overhead, making them less of a fundamental solution. Hence, the implementation of meta-heuristic (or) bio inspired algorithms are essential for quick detection for a real-time detection systems.

In terms of identifying familiar attacks, the signature-based approach excels in both speed and accuracy. Nonetheless, it encounters challenges in discerning unknown threats like zero-day attacks and remains susceptible to variant attacks that evade detection through techniques such as encryption or code obfuscation. The maintenance of an up-to-date signature database also imposes a substantial burden on the system.

Unlike the signature-based approach, anomaly detection relies on analyzing the statistical attributes of individual network traffic rows. It identifies potential attacks by flagging deviations from typical behavior, such as exceeding the established range of statistical norms. One notable advantage of this method is its independence from a signature database, eliminating the need for ongoing maintenance and rendering it highly resilient against novel or variant attacks. Modern network intrusion detection systems (NIDSs) often leverage diverse machine learning techniques for distinguishing between normal and attack-related network rows within the framework of anomaly detection.

Achieving both speed and accuracy with machine learning algorithms for processing real-time network traffic poses a considerable challenge. Moreover, anomaly detection algorithms typically categorize traffic by analyzing complete rows, considering attributes like the number of transmitted and received packets, packet loss, and more. This approach contrasts with classifying based on each individual packet. Consequently, the determination of whether a row is malicious only begins once the row has concluded. This per-row classification method has the limitation of not being able to detect attacks until the flow is complete, offering limited defense for the current network and users during an ongoing attack [7].

Developing a proficient Intrusion Prevention System (IPS) encounters significant hurdles centered around the delicate balance between precision and computational efficiency. The utilization of datasets characterized by higher dimensions poses a noteworthy challenge, as it can deterioratethe model's classification accuracy and time consumption [8] – [9]. To address this issue and to improve classification accuracy, a dimensionality reduction technique LDA and A-PCA is employed. The study utilizes the publicly accessible NIDS V.10 2017 dataset obtained from the Kaggle repository. Normalization of the dataset is carried out using the Z-Score process, and the scaled data is then utilized as input. Employing CNN, in this study focuses on classifying the prominent attributes within the data. The WOA is applied to seek optimal solutions. The activity function is optimized, and the dataset with reduced dimensions is subsequently classified using a tuned CNN that ensures hyper-parameter optimization. Lastly, the outcomes are evaluated and compared against models employing LDA+A-PCA, WOA and CNN.

1.1 Motivation

Developing an Intrusion Prevention System (IPS) model using DL algorithms is imperative in contemporary Cyber-Security landscapes due to the ever-evolving nature of cyber threats. Conventional methods often fall short in effectively identifying and mitigating sophisticated attacks, leading to increased vulnerability. Cyberthreats, with their inherent capacity to exploit vulnerabilities and evade traditional defenses, pose a substantial risk to critical systems and sensitive data. The drawbacks of current Cyber-Security approaches lie in their limited adaptability to rapidly changing attack methodologies, making them susceptible to evasion tactics and false negatives. To address these limitations, the authors opted

for advanced techniques such as LDA and A-PCA for dimensionality reduction, ensuring a streamlined and informative dataset. The inclusion of the WOA enhances the model's ability to find optimal solutions, thereby improving its overall efficiency. Finally, the authors developed an optimized CNN model, leveraging its DL capabilities for efficient feature extraction and classification, enhancing the IPS's accuracy and responsiveness in identifying and thwarting malicious activities. This comprehensive approach not only addresses the shortcomings of existing methods but also establishes a robust and adaptive framework for Cyber Security in the face of evolving cyber threats.

The contributions of the article are as follows:

- To reduce the processing latency, to increase the effectiveness of the filtering part and to form a standardized dataset, A-PCA and LDA are deployed, which prompt the acceleration of prevention scope.
- Feature extraction is done by implementing WOA, in which the best optimal values are found using the "Rule of Convergence" method.
- We have also introduced "Re-Routing" concept, in-order to enhance the classification accuracy.
- The OR-CNN adopts the nature of pattern analysis with respect to the complexity of the input data(unknown attack pattern) under test.

2. RELATED WORKS

The Intrusion Prevention System (IPS) actively keeps tracking the activities of network in-order to detect anomaly behavior using traditional techniques. To prevent similar attacks in the future, the IPS takes measures such as closing access points, terminating TCP sessions, reprogramming firewalls, and eliminating traces of attacks from infected files, headers and payloads. In identifying intrusions, the IPS employs stateful protocol based, anomaly and signature based analysis for both host and network based Intrusion Detection System (IDS). When it comes to setting them up, IPS and IDS are usually set up together, working together to make an Intrusion Detection and Prevention System (IDPS). But the IDPS methods we have now have trouble finding and stopping attacks in networks with a lot of moving parts [10].

Nikhil et al. [11] developed an integrated prevention system utilizing Support Vector Classification (SVC) and CNN. Their experimentation involved a real-time agriculture dataset, where images of animal entries were used as input for model training. The system demonstrated high accuracy in preventing the intrusion of three types of animals into the agriculture field.

Raghavendra et al. [12] introduce a prevention technique based on a SVM with Least Square Bolster, employing a two-segment approach. A hybrid mechanism is incorporated to eliminate superfluous data at the upper level. The wrapper method is employed for the selection of pertinent features during classification at the lower level. Following the attack classification, features with significant influence on the classification outcome are identified, and corresponding entries are obstructed to prevent intrusions.

In their paper [13], Akhil et al. describe a method that combines Multi-Layer Perceptron (MLP) and SVM to find Probe, U2R (User to Root), DoS, and R2L (Remote to Local) threats. One way to stop these kinds of attacks is to use an internal script that checks things like IP addresses and port numbers. The research work described in [14] introduces a detection and prevention technique utilizing a Discriminate Deep Belief Network (DDBN) with both local and non-local regularization. This model undergoes testing on two widely used datasets employing classifiers such as Hopfield, Deep Belief Network-Random Forest (DBN-RFS), SVM and Generative Adversarial Network (GAN).

In the development of prevention techniques, diverse parameters are altered to minimize the time required for detecting the attack category. They have noticed that the running time goes down as the number of secret layers in the model goes up.

Balamurugan et al. [15] present a dual-phase approach for the real-time detection and prevention of cloud dataset security issues. The technique involves three key components: Virtual Machine Management (VMM), Trust Authority (TA) and Cloud Controller (CC). The CC oversees packet monitoring and migration to idle cloudlets in the presence of heavy traffic. It rates packets by looking closely at the header information to see how confident it is in the arrival time and the number of packets. The Normalized K-means (NK) Recurring Neural Network model (NK-RNN) is used in the VMM to sort attacker packets as part of the breach detection. A method called "Queue modeling" is used to drop malicious packets that have been recognized, stopping them from the network to stop future attacks [15].

As part of their plan, Amir Ali et al. [16] talk about a Software Defined Network (SDN)-based IDPS that is specifically made for IoT networks. A three-tier structure is used in this method. The first tier is in charge of validating users at the IoT layer. The second tier is in charge of validating packets at the data plane layer and using fuzzy filters to sort attack records into groups. In the third level, confirmation takes place in the control plane layer to find problems and stop them. Deep Packet Inspection (DPI) and CNN work together with the control layer to find and predict attack values. There is only a 1% chance of getting a false result when you compare Fuzzy, ANN, SVM, and other machine learning methods.

This idea from Islabudden et al. [17] is a mix of three different models: the Bootstrapped Optimistic Algorithm for Tree Construction (BOAT), the Artificial Neural Network for classification, and the One Way Hash Chain (SHA-256) for security in Mobile Ad Hoc Networks (MANET). The main parts of this model are a fuzzy controller for the packet analyzer, data pre-processing with logarithmic and linear normalization, feature extraction with the Mutual Information function to find the best set of features, and classification with the Association Rule Tree (ART) [17].

James et al. [18] have developed a risk analysis model that focuses on breaches related to confidentiality, authentication, and access control, as outlined in three specific test cases. The proposed model operates at multiple levels: firstly, it identifies risks by analyzing events and their defined relationships. Subsequently, it prioritizes, evaluates, and ranks the risk factors. The approach selects a mitigation plan according to the correlations among risks and typical sources of threats. Subsequently, it evaluates feasibility and executes an appropriate solution, all the while monitoring performance regularly to proactively avert attacks.

A CNN uses a deep learning method with a convolution layer to automatically pull out useful features from the original data feature plane. Previous study [19] has shown that choosing the best hyper-parameters is very important for how well deep learning models work.

To find the best hyper-parameters, there are two main ways: human search methods and computer search methods. With the manual setup method, it can be hard to find a good network topology by chance. It's hard to make hyper-parameters automatically fine-tuned for the best model structure because hyper-parameter tuning includes working with an uncertain goal function. Traditional optimization approaches like gradient descent or Newton's method are impractical in this context [20].

Biological heuristic techniques, including but not limited to the ACO, GA, A-BCA, PSO, WHO and BBO optimization algorithms, enhance the effectiveness of deep learning models through the optimization of hyper-parameters.

A hybrid optimization approach introduced by Caroline et al. [21] developed a hybrid

optimization approaches combining genetic optimization and particle swarm to identify the optimal hyper-parameters for fully connected layers. This study showed a significant improvement of about 0.90 F1-score across all three networks: VGG-16, DenseNet-201, and ResNet-50.

Gurukumar et al. [22] came up with the Fitness Sorted Rider Optimization Algorithm (FS-ROA) to find the best hyper-parameters. These include the number of neurons in the fully connected layer, the pooling layer, and the convolutional layer; as well as the number of neurons in the hidden layers. The study got a great response rate—more than 97% of people agreed with it.

Bayesian optimization introduced by Wu Chen et al. [23] to identify the optimal hyper-parameters for machine learning algorithms. The experiment revealed that this methodology effectively determines optimal hyper-parameters for widely used machine learning models, including Random Forest (RF) and Neural Networks (NN) algorithm.

In their paper [24], Elmasry et al. described a way to use Particle Swarm Optimization (PSO) to find hyper-parameters and choose feature subsets for deep learning models, such as Deep Neural Networks (DNN), Long Short Term Memory - Recurrent Neural Networks (LSTM-RNN), and Deep Belief Networks (DBN).

This program was created by Xiu Kan et al. [25] so that it could spot different kinds of attacks on IoT networks.

In their paper [26], Alharbi et al. created a new model that uses the Bat Algorithm (BA) and the local-global best bat algorithm, Particle Swarm Optimization (PSO), to find the neural network hyper-parameters needed for IoT attack detection. Compared to both BA-NN and PSO-NN, the planned LGBA-NN was more accurate.

In their paper [27], Sajjad et al. described a way to find hyper-parameters in machine learning methods by using both the Grey Wolf Optimization (GWO) and the Genetic Algorithm (GA). The results of the experiment showed that performance got better during the training phases in all of the rounds. In addition, GWO showed better performance, which was backed by a p-value of 2.6E.

A 3D search space Whale Optimization Algorithm (WOA) was created by Andrzej et al. [28]. It uses hyper-parameter optimization, but the results showed that it wasn't very accurate.

Ali et al. [29] wrote an article about an evolutionary sparse convolution network Intrusion Detection System (IDS) that can find different types of IoT network threats. The model was able to lower the number of fake alarms and raise the number of real alarms. Still, it's important to keep in mind that this method involves steps that take a lot of time and require complex computer tools.

Abdullah et al. [30] introduced an adapted Grey Wolf Optimizer (GWO) for the purpose of hyper-parameter identification in the Extreme Learning Machine. This modified model was designed to significantly reduce the false positive rate to below thirty percent.

Journal of Theoretical and Applied Information Technology
15th August 2024. Vol.102. No. 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED METHODOLOGY

3.1. Work flow of Proposed Model

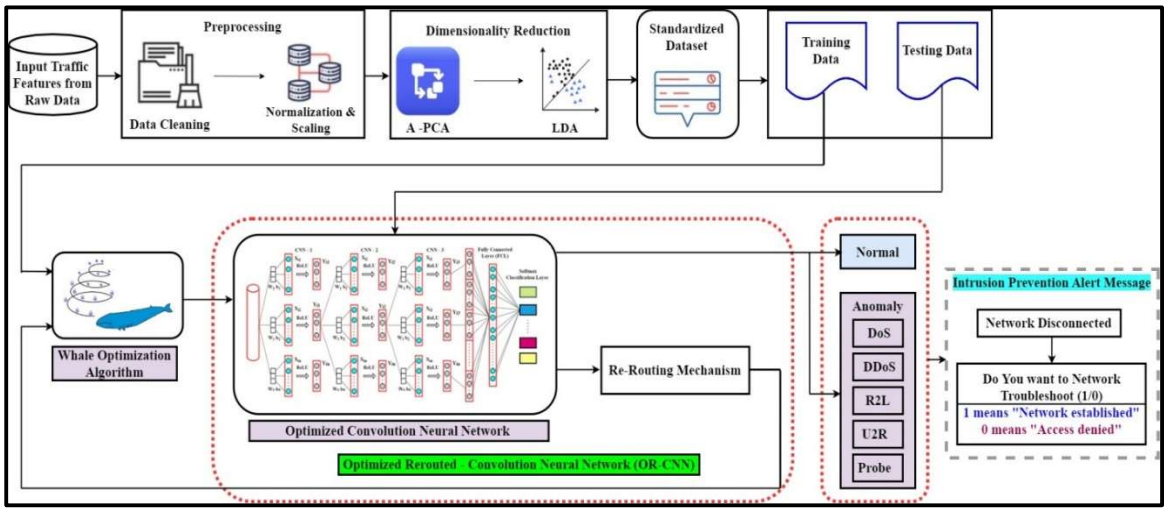


Figure 1: Proposed Work Flow Of NIPS Model

3.1.1 Dataset Description

One part of the NIDS V.10 2017 dataset has 22,544 rows and 41 columns are utilized for training. The other part has 25,192 rows and 42 columns are called the Train Dataset. These attributes play a pivotal role in providing ground truth labels that accurately represent various network activities and their characteristics. It encompasses a wide spectrum of network activities, encompassing both "Normal" and various forms of "Anomaly Traffics like DoS, DDoS, R2L, U2R and Probe". This dataset is available as an open source document and can be directly accessible from [31]. Table 1 shows the dataset repository of NIDS V.10 2017 dataset.

Table 1: Data Repository For Feature Extraction

Category			**Category**		
Basic Features	No.	**Attribute Name**	Content Related Features	No.	**Attribute Name**
	1	duration		10	num_failed_logins
	2	protocol		11	logged_in
	3	service		12	num_compromised
	4	flag		13	root_shell
	5	src_byte		14	su_attempted
	6	dst_byte		15	num_root
	7	land		16	num_file_creations
	8	wrong_fragment		17	num_shells
	9	urgent		18	hot

Category			**Category**		
Content Related Features	No.	**Attribute Name**	Time Related Features	No.	**Attribute Name**
	19	num_access_files		23	count
	20	num_outbound_commands		24	srv_count
	21	is_host_login		25	25 serror_rate
	22	is_guest_login		26	srv_error_rate
				27	rerror_rate
				28	srv_rerror_rate

				29	same_srv_rate
				30	diff_srv_rate
				31	srv_diff_host_rate

Category			**Category**		
Host Based Traffic Features	No.	**Attribute Name**	Host Based Traffic Features	No.	**Attribute Name**
	32	dst_host_count		39	dst_host_srv_serror_rate
	33	dst_host_srv_count		40	dst_host_rerror_rate
	34	dst_host_same_srv_rate		41	dst_host_srv_error_rate
	35	dst_host_diff_srv_rate			
	36	dst_host_same_src_port_rate			
	37	dst_host_srv_diff_host_rate			
	38	dst_host_serror_rate			

3.1.2 Preprocessing

The network traffic data and associated features are gathered and stored in a CSV file named NIDS V.10 2017 dataset. To build a standardized dataset we implemented "Normalization" technique using Z Score standardization technique. This process rescales and centers the data so that it has mean (μ) values of 0 and standard deviation (σ) values of 1.The formula for Z-score normalization is as follows:

$$Z = \frac{(X - \mu)}{\sigma} \quad (1)$$

Where,

The standardized value is Z, the original value of the data point is X, the mean of the data set is μ, and the standard deviation of the data set is π.

The goal of this step is to make sure that all of the continuous initial variables have the same range so that they all add the same amount to the study. There is a general process model of the NIPS model shown in Figure 1.

3.1.3 Standardizing the Dataset

The dataset is read and addressed fully in-order to identify any missing data points and also to remove the in-complete samples from the raw dataset. Typically, datasets exhibit a higher concentration of similar data types, resulting in smaller variance within those specific data categories. Principal Component Analysis (PCA) leverages this characteristic by generating a set of orthogonal values. These numbers are used to project high-dimensional data onto a level set. This changes the data into a picture with fewer dimensions. The objective is to maximize the variance in the reduced dataset, ensuring the retention of a significant portion of the original information [32].

A data-centric method called as PCA, is deployed to dwindle the dataset dimension by keeping the original values up-to the maximum extent.

The principle of PCA is elucidated based on a dataset D = $[(p_j, q_j)|p_j \in S^x, q_j \in S^y, j=1, 2\cdots N]$ comprising N samples. Each sample, as indicated earlier, is characterized by a feature set denoted as $P_i = [p_{ij}, p_{2j}\ldots..p_{sj}]$, where,s represents the maximum number of features in a sample. Consequently, the entire dataset Z =$[Z_1, Z_2\ldots..Z_n]$ can be expressed as,

$$Z = \begin{bmatrix} z_{11} & z_{12} & z_{1t} \\ z_{21} & z_{22} & z_{2t} \\ z_{s1} & z_{s2} & z_{st} \end{bmatrix} \quad (2)$$

Initially, we establish the definitions for the mean observed value μ and the mean deviation of observed values δ, which can be expressed as,

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (3)$$

Subsequently, δ can be obtained through the decentralization of all samples. This can be calculated using the formula,

$$\delta_i = x_i - \mu = x_i - \frac{1}{n}\sum_{i=1}^{n} x_i \quad (4)$$

Finally, the dataset's covariance matrix is calculated by,

$$\text{Covariance} = \frac{1}{n}\sum_{i=1}^{n} (\delta_t)(\delta_t)^T = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_t - \mu)^T \quad (5)$$

Hence, employing Singular Value Decomposition (SVD) allows us to derive the eigenvalues $\{\alpha_1, \alpha_2, \ldots \alpha_p\}$ and their corresponding eigenvectors $\{\omega_1, \omega_2, \ldots \omega_p\}$ from the covariance matrix. Notably, $\{\omega_1, \omega_2, \ldots \omega_p\}$ represents the set of maximum linearly independent eigenvectors, where $1 \leq p \leq m$. Consequently, a new sample space can be reconstructed by selecting specific eigenvectors based on their corresponding eigenvalues.

The integration of PCA and adaptive control theory is known as Adaptive Principal Component Analysis (A-PCA). Using predetermined performance metrics as a guide, this technique selects features after PCA decomposition. The obtained dimension values from PCA are used to automatically modify the step size α of the compression process.

The final outcome of the A-PCA is represented as,

$$\text{Final dataset (Y)} = (\text{Feature Vector})^T \times (\text{Standardized original dataset})^T \quad (6)$$

To distinguish between the classes of data in the dataset (i.e. "Normal" and "Anomaly"), Linear Discriminant Analysis (LDA) is deployed in our proposed method. This statistical method's main goal is to find the linear mix of traits that makes the difference between predefined classes the biggest.

The multiple class LDA is derived from the formula,

$$\Sigma_b = \frac{1}{C}\sum_{i=1}^{c}(\mu_i - \mu)(\mu_i - \mu)^T \quad (7)$$

Where,

Σ_b is the Scatter matrix, C is the number of samples in class i, $\mu i$is the mean vector of class i.

In LDA, the within-class scatter matrix (SW) measures how spread out data are within each class. The value is found by adding up the covariance matrices for each class. Each covariance matrix is weighted by the number of samples in that class.

$$SW = \Sigma (N_i \Sigma_i) \quad (8)$$

Where,

Σi represents the covariance matrix associated with class i.

LDA seeks to discover a projection that optimally increases the gap between classes while simultaneously reducing the dispersion within each class. To do this, Fisher's criterion is used, which is found by dividing the between-class scatter matrix (SB) determinant by the within-class scatter matrix (SW) determinant [33].

$$J(w) = \frac{w^T SB_W}{w^T SW_w} \quad (9)$$

Where,

Projection vector is represented as w.

When using LDA, the job of finding the best projection vector w means answering a math problem called the generalized eigenvalue, which can be written as

$$SB_w = \lambda SW_w \quad (10)$$

Where,

λ represents the eigenvalue associated with the optimal projection vector w.

In the process of LDA, the selection of the top k eigenvectors, which correspond to the largest eigenvalues, serves as the basis for constructing projection vectors. These vectors collectively define a lower-dimensional subspace that retains the most crucial discriminative information among different classes.

For classification purposes, when dealing with a new observation, LDA performs a projection of the observation onto the subspace determined by the projection vectors. Subsequently, the

classification involves assigning the observation to the class whose mean is closest in the projected space.

It is crucial to standardize data before applying PCA because PCA is highly sensitive to variations in the initial variables' ranges. When substantial differences exist among the scales of the original variables, those with larger ranges exert a disproportionate influence on the principal components, potentially causing biased outcomes. For instance, a variable ranging from 0 to 100 may overshadow a variable with a range of 0 to 1. Standardizing the data mitigates this issue by transforming it into comparable scales, ensuring that PCA accurately identifies underlying patterns and relationships without being swayed by the magnitudes of individual variables. This approach fosters an unbiased representation of the data's intrinsic structure and facilitates the extraction of meaningful principal components.

Our dataset is chopped into 80% training and 20% testing to assess the "Performance" and "Accuracy" of proposed model.

3.1.4 Whale Optimization Algorithm

The WOA algorithm is an optimization approach inspired by herd intelligence, mimicking the predation strategies employed by humpback whales during their food hunting activities. The foraging approach of humpback whales involves a specific sequence: after identifying their prey, the whales initiate the creation of a bubble network, following a spiral path as they move upstream towards their target. The WOA operation can be categorized into three key stages: converging on prey, executing the bubble network attack, and finally, stalking down the identified prey [34].

In the initial stage, the whales engage in swarming around the prey. This is necessary because, initially, the whales lack precise information about the prey's specific location. When the current optimal position corresponds to the targeted prey, every individual whale within the group proceeds towards it.

$$Y(n+1) = Y^*(n) - P.S \quad (11)$$

$$S = | R \times Y^*(n) - Y(t) | \quad (12)$$

Where,

equations 11, 12, 13 and 14 represents the behavior of Whale.

Here, n denotes the current iteration. The current optimal solution or the prey's vector position is represented as Y*(n). The current optimal position is represented as Y(n). The step size is denoted as P and S. rand is random number ranging from 0 to 1 shown in equation 12, and the control parameter is denoted as x which is linearly decreases from 2 to 0 as iteration progress.

$$P = 2x \times rand\text{-}x \quad (13)$$

$$Q = 2 \times rand \quad (14)$$

$$X = 2 - \frac{2^n}{I_{max}} \quad (15)$$

The variable I_{max} represents the maximum number of iterations. In the second phase, the whale initiates the construction of a bubble network by navigating within a confined enclosure along a spiral path towards the prey. WOA classifies this behavior into two components: the shrinking and crowding processes, and the spiral update positions. The convergence factor x in equations 13, 14 and 15governs the shrinkage and crowding mechanism. The computation of the spiral update position involves determining the distance between individual whales and their current best location, followed by simulating the whale's capture of its prey in a spiral motion. This methodology can be expressed as,

$$Y(n+1) = S' \times e^{uv} \times \cos(2\Pi V) + Y^*(n) \quad (16)$$

$$S' = | Y^*(n) - Y(n) | \quad (17)$$

In equation 16, S' is the distance between the current best place and the whale in question. and u and v are fixed coefficients that define the logarithmic spiral and a random number from -1 to 1, respectively. The spiral envelope and contraction envelope are executed with equal

probability, creating this synchronization model. The present best solution is replaced with a whale at random if |P|≥ 1 in the third stage of prey hunting. By diverting the whale's attention from the present reference target and initiating the hunt for a more suitable prey, this procedure improves the algorithm's capacity for global exploration.

$$Y(n+1) = Y_{rand} - P \times S \quad (18)$$

$$S = | Q \times Y_{rand} - Y(n) | \quad (19)$$

Where,

Y_{rand} denotes the whale's vector location which is selected randomly.

This study will use WOA to find the best setting for different parts of a CNN, like the number of fully connected layers, ReLU layers, Softmax layers, neurons in each hidden layer, and the use of dropout in hidden layers, so that the results are as accurate as possible [35].

Algorithm 1

Step1: The population of whales are initialized
Step2: Every search agent fitness is calculated to determine which search agent is best
Step3: while(I_{max}< maximum number of iterations)
for each search agent:
Update
if(p<0.5):
if(|P|<1):
Update current agent by equation (12)
else:

Select a random agent
Update current agent by equation (18)
else:
Update search agent by equation (15)
end-for
Check if any search agent goes beyond the search space and amend it
Calculate fitness of each search agent
Update if there is a better solution
t = t+1
end-while
Step4: return

To begin, we initialize the whale population represented by a random position along with their corresponding values. After that, in order to determine the optimal search agent position, we assess the fitness solution values of the agents. Next, we set the iteration count to 1. Subsequent to applying equation 12 to locate prey, we employ Equation 19 to surround the identified prey. Once the prey is encircled, we proceed to update the positions of the search agents, employing the bubble-net strategy outlined in equations 16 and 17 to launch an attack on the prey. The best fitness values are then updated based on the new positions of the search agents. Following this, the equality and inequality constraints of WOA variable and |P| for each search agent's new position are verified. The iteration count is increased before repeating the entire process.

The algorithm's final step is to check if the maximum number of iterations (I_{max} \ max. no. of iterations) has been used. The procedure ends with the fitness value being saved as the optimal solution if this is true. If it doesn't work, iterating the process until it does [36].

3.1.5 Optimized Re-Routing Convolution Neural Network (OR-CNN) Design

To prevent the malicious patterns in the network, an efficient CNN is optimized and developed. The design includes ReLU, max-pooling,

convolutional, and Softmax layers to improve model performance. Feature learning is facilitated through a fully connected layer configuration in this experimental setup. The network's output is designed with 6 nodes, each dedicated to the prevention of distinct types of intrusions.

The proposed optimized CNN has an input layer with a size of 1000 x 1 x 1, indicating its compatibility with three-channel images having a resolution of 64 x 64. Typically, images obtained from the internet or online sources come with three channels, hence the input layer is configured to accommodate such images. Signal propagation from the input layer proceeds to the initial convolutional layer, which is equipped with 10 filters, each with a kernel size of 1 x 1.The operational concept of this layer is articulated through equation 20.

$$Z(p,q) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} K(p+i,\ q+j)C(i,j) \quad (20)$$

Where,

"C" represents Convolution filter of every layer, "Z" represents the feature map output, "n" represents the kernel size.

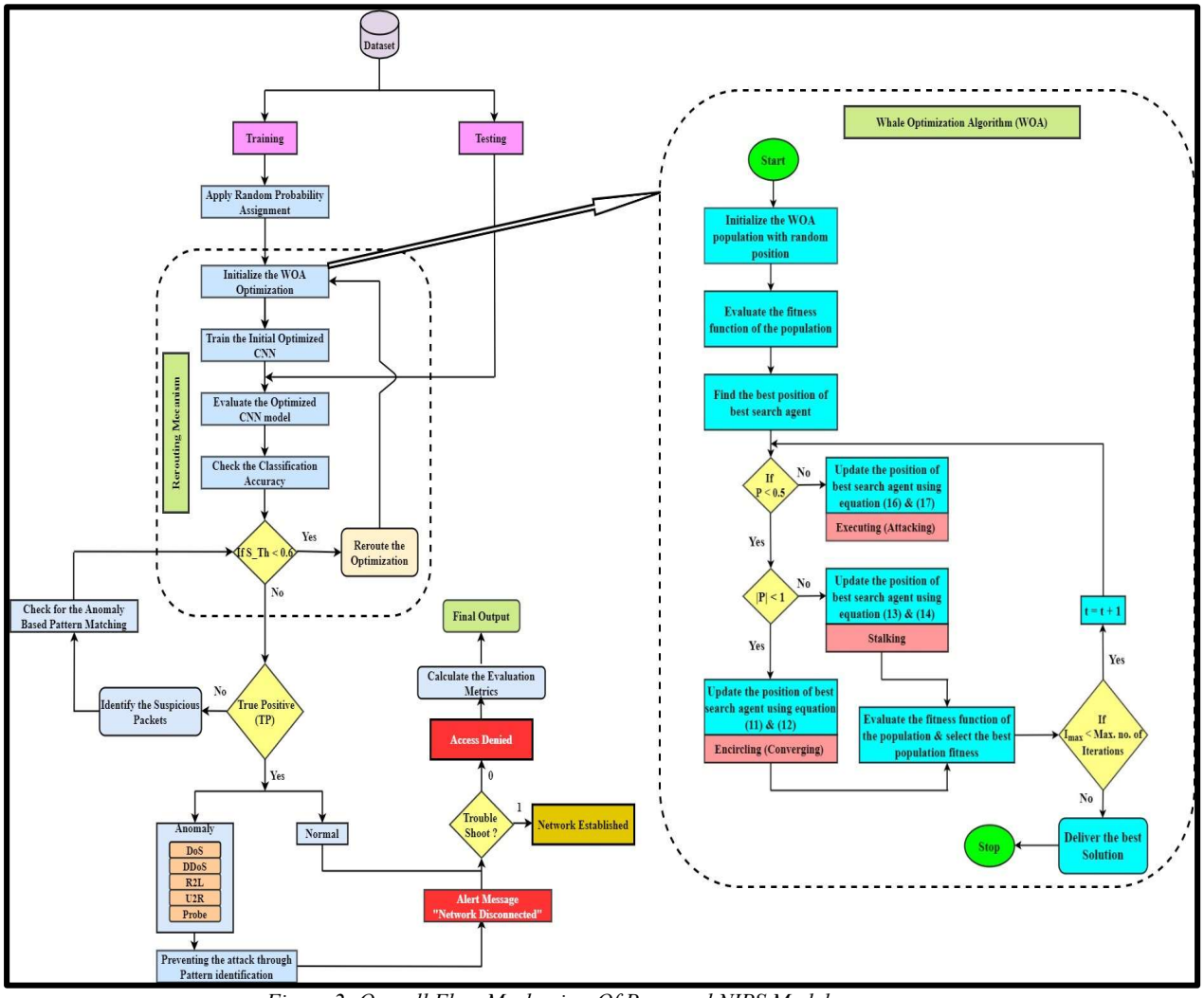


Figure 2: Overall Flow Mechanism Of Proposed NIPS Model

The activation function employed for the convolutional layers is the ReLU, as specified in equation 21. In the proposed network architecture, ReLU is applied element-wise to preserve non-linear characteristics.ReLU helps the network to learn complex patterns and

enables it to approximate any non-linear function.

ReLU (r) = max(0,r) (21)

A max pooling layer is added to the proposed optimized CNN architecture after the convolutional layers in order to perform downsampling. Among the most crucial points of feature extraction is the lowering of the spatial dimension in the input feature map, which is accomplished by applying the max pooling layer, which is defined in equation (22).

$$L(p,q) = \max_{i=0}^{n-1} max_{j=0}^{n-1} K(p \times n + i, q \times n + j) \quad (22)$$

Feature maps' spatial dimensions which are generated by the convolutional layers are reduced by pooling layers. They aggregate information in small regions of the input data. The feature maps were partitioned into distinct non-overlapping regions, followed by the extraction of maximum values within each region using equation 22. In this equation, L(p, q) represents the output obtained from the pooling layer, and the pool size is denoted as 'n'. The input feature map, corresponding to the input image, is denoted as 'K'.Pooling helps reduce the computational complexity and controls over-fitting.

In the course of the experiment, it was noted that an increased number of convolutional layers is essential. Consequently, there is a need to augment the network with additional ReLU activation functions and pooling layers. These supplementary layers are deemed crucial for enhancing the overall performance of the proposed network. The extra layer introduced comprises a convolutional layer featuring 10 filters, each with a kernel size of 1 x 1.After this convolutional layer's signals have been processed using the ReLU activation function, the next step is to use a max pooling layer with a 2 x 2 pool size. However, it has been observed that this architecture has a tendency to overfit. To address this issue, a dropout layer has been introduced which is governed by the equation 23.

The model then advances with more convolutional layers that include ReLU activation, 10 filters, and a 1 x 1 kernel size. Then, a dropout layer is added after a maximum pooling layer has been created with a 2 × 2 pool size. To prevent over-fitting, dropout is a regularization technique that, while training, arbitrarily changes certain input units to 0.

$$R(p) = \begin{cases} p, \text{with probabilty } 1 - x \\ 0, \text{with probabilty } x \end{cases} \quad (23)$$

As outlined in equation 23, the dropout process involves the random selection of weights, which are then set to zero. Consequently, these specific weights do not contribute to the training, leading to increased regularization and a preventive measure against over-fitting in the network. In this equation, denoted as R(p), the dropout layer is characterized by the input parameter p. where x represents the dropout itself.

The fully connected layer comes into play following the dropout layer, effectively mitigating over-fitting concerns. Additionally, a flattened layer is employed to transform each signal into a 1-D array. Subsequently, this 1-D data is transmitted to the fully connected layers, often referred to as dense layers. The first completely linked layer has one hundred hidden nodes, and the Rectified Linear Unit (ReLU) activation function is used to activate each node.

To mitigate over-fitting in our proposed optimized CNN, a dropout rate of 0.5% was employed. Subsequently, an additional fully connected layer consisting of 100 hidden nodes was incorporated, each utilizing the same activation function. This entire procedure is succinctly represented by equation 24.

$$x_i = act(\sum_{k=1}^{n} g_{ij} u_k + b_i) \quad (24)$$

Where,

"x_i" represents the output and "i" represents the unit in the fully connected layer.

"b_i" represents the bias factor and "u_k" and "g_{ij}" represents the input and matrices weights.

All terms are indexed at i, with the exception of u, which is the index of the input. Equation 24 sums up the number of pixels in the input, indexed at j, and then iterates over those numbers. In this experimental setting, the ReLU activation function affects the entire expression.

The proposed NIPS model leverages the NIDS dataset V.10 2017, focusing on addressing 6 distinct types of intrusions identified as threats to the Internet of Things (IoT) in the cascaded model. Consequently, the output layer of the designed optimized CNN comprises six nodes, aligning with these specific intrusion types. The activation function chosen for the hidden nodes is Rectified Linear Unit (ReLU). On the other hand, the output nodes are activated using the Softmax algorithm. Using Softmax guarantees that the optimized CNN's outputs are normalized between 0 and 1, which is necessary for obtaining scaled cascaded answers. The scaled output can be calculated using the formula,

$$y_i = \frac{e(y_i)}{\sum_{l=1}^{n} e(y_j)} \qquad (25)$$

The dense layer, represented by the letter yi, provides the input for equation 25 and is processed using the Softmax classifier. The number of output nodes in this equation is represented by the parameter 'n'. Since there are 6 output classes in this experiment, 'n' is set to 6, which correspond to the six different intrusions.

At finally, the model is put together using the categorical cross-entropy loss function. The improved convolutional neural network (CNN) that was developed can sort data into several categories all at once. In the context of such classifiers, the cross-entropy loss function is preferred, as it tends to yield more effective results. This can be calculated using the equation,

$$S(y,y') = - \sum_{m=1}^{n} y_i \log (y'_i) \qquad (26)$$

A categorical output is represented by the output class label y', in equation 26. The One-Hot encoding system, which uses binary sequences to represent various classes, is used to manage categorical outputs. The model predicts in the form of a probability distribution, and its value is represented as y'. In this way, the maximum of 'n' classes is predicted by the proposed optimized CNN [37].

Stochastic Gradient Descent (SGD) [38] is a well-known repeated optimization method that is often used to train deep learning models. This method specializes in updating the hyper-parameters of the proposed model when we deal with big dataset. The fundamental feature of SGD is its capacity to update model parameters by incrementally moving tiny steps in the direction of the parameters' negative gradient in the loss function. This feature guarantees effective and flexible model modifications throughout the training procedure.Equation 27 encapsulates the update mechanism for each parameter δ_i.

$$\delta_i^{(n+1)} = \delta_i^{(n)} - \gamma \frac{\partial G}{\partial \delta_i} \qquad (27)$$

Where,

$\delta_i^{(n+1)}$ represents the current value of δ_i parameter at (n+1) iteration. The step size is controlled by the learning rate and is denoted as γ. The loss function's gradient is calculated by $\frac{\partial G}{\partial \delta_i}$ with respect to δ_i.

Gradient estimation often employs a randomly chosen subset of training data, referred to as a mini-batch, to lower computational expenses. SGDiteratively executes parameter updates using these mini-batches until reaching convergence or a predefined number of iterations.Despite its vulnerability to noisy updates, SGD demonstrates rapid convergence and efficient handling of large datasets, rendering it a widely favored option for training various ML models [39].

3.1.5.1 Re-Routing Mechanism

CNN is highly adaptable and it excels automatically in learning the relevant features

from raw data. It can handle a variety of network traffic patterns, which learn to recognize anomalies and extract the features effectively. Through effective learning from labeled datasets, CNNs have the potential to reduce False Positives (FP), improving the accuracy of intrusion prevention and minimizing the chances of misidentifying normal network activities as malicious.

In general, the traditional CNN itself functions as an optimizer initially, as it processes more than 10,000 data within the filters during convolution. However, when dealing with unstructured data in Cyber Security,the impact of anomalies at different levels may affect the classification accuracy due to its complexity. This complexity can lead to significant time consumption on GPUs during the training process.

To overcome this drawback, the proposed CNN is "Optimized" and converted into a lightweight model.

The NIDS dataset is splitted randomly into 80% training and 20% testing in-order to develop our proposed NIPS model. When the suggested CNN is first set up, the weights of the neurons are randomly applied to make it work better. This method makes it easier to learn all the different parts of training.

However, the stochastic nature of the proposed CNN model may lead to variability in model performance across different training runs. Achieving optimal performance might heavily depend on the specific random seed used during initialization or other random processes. Our proposed unstructured data contains numerous irrelevant or predominantly zero values, which significantly impact the classification accuracy, since our input is randomized probability. To overcome this drawback, a WOA optimizer is initialized here, to address the presence of irrelevant or sparse data (potentially zeros), ensuring a more accurate classification process. If the classification accuracy obtained from the proposed CNN is below a predefined threshold, such as less than 60%, the optimization process will be "**RE-ROUTED**" as shown in figure 2.

Subsequently, the test data process involves multiple rounds of optimization when we reapplying the data into CNN. During this iterative process, the input data may exhibit a level of randomness due to the dynamic nature of the optimization. Consequently, the CNN may repeat its operation, and it's noteworthy that the structure remains consistent across CNN layers, including the input layer and fully connected layers. The dynamic nature of this process, influenced by the randomness in the data, leads to a more adaptive and responsive system.

We have optimized our CNN model in-order to perform Normalization technique which results in high classification accuracy. We cascaded the Softmax layer for better adjustment. The size of the fully connected layer is reduced to (100 × 100 × 6) and the convolution layer stride is fixed with (1 × 10). After various analysis and approaches, the above said configurations are well suited to develop our proposed NIPS model, particularly for this NIDS dataset.

This comprehensive approach that dynamically adapts to unstructured data is essential to prevent the Cyber-Attacks. It is also to noteworthy that, the iterative nature of the process and optimization concept, driven by data dynamics which is deployed in CNN makes easy to handle complex and robust solutions. Through repeated optimization and cascaded structures, the proposed **"Optimized Rerouted Convolution Neural Network (OR-CNN)"**is developed. This promising approach enhances the classification accuracy in preventing the Cyber-Attacks with reduced time consumption.

3.1.5.2 UML Flow Sequence

Figure 3 shows the UML flow sequence of the proposed model. The monitor agent (IDS) is responsible for receiving the packet from the network and passing it to the Anomaly identifier. This identifier is responsible for analyzing and reassuring for any suspicious activity. Then the packet passes to the traffic blocking agent which blocks the network traffic by which it identifies as "Anomaly based Threat". Finally the proposed

NIPS model successfully prevents the packet once the model finds it as "True Positive (TP). Both during on-time and off-time, the analyzer agent can be operated. Based on the anomaly pattern approach, it is up-to analyzer agent amenable to examine whether it is a "Normal" (or) "Anomaly packet". The traffic blocking agent receives the packet from the analyzer and check for the type of pattern. If it is "False Positive (FP)" the network will be in In-line mode, but if it is "True Positive (TP)", the network goes to Off-line mode. The administrator user is the first to get an alert message that reads "Network Disconnected" in the event of a suspicious intrusion or threat.

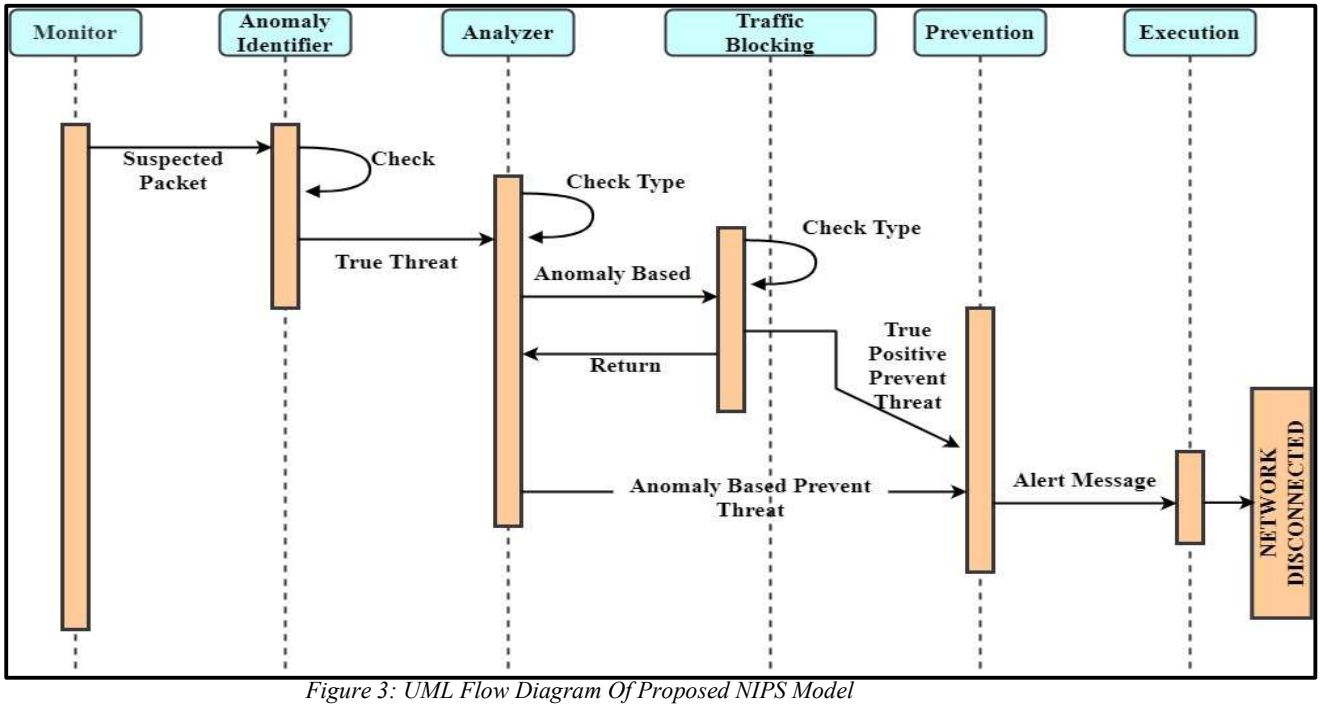


Figure 3: UML Flow Diagram Of Proposed NIPS Model

4. RESULTS AND DISCUSSIONS

This part talks about the research and results of our suggested NIPS method. After that, we gave a total result to show how well the proposed model works at preventing anomalies. We figure out how accurate the Whale Optimization Algorithm (WOA) and the OR-CNN approach are and compared them to other machine learning and deep learning methods that are already existed.

A MATLAB tool of version 2019b is utilized to develop the proposed model. The AMD RADEON GPU processors running at 1.99 GHz and an Intel Core i7-8550U CPU clocked at 1.80 GHz comprise the gear setup for this research work. The computer has 12.00 GB of RAM and is powered by a 64-bit operating system.

We simulated the proposed NIPS model using an average of 200 epoch's in-order to reach the maximum optimal level. The NIDS V.10 2017 dataset, which is frequently used in Cyber-Security research, is utilized to check the performance of the proposed NIPS model. The records and features of this dataset in shown in table 1 of section 3.1.1

4.1 Evaluation Metrics

It is calculated what the True Positive (TuPe), True Negative (TuNe), False Positive (FePe), and False Negative (FeNe) are. The metrics Precision, F-Measure, and Recall are used to get a better idea of how well the suggested NIPS model works.

4.2 Whale Optimization Algorithm (WOA) Results

When applied to optimization issues, WOA attempts to discover the best solution by mimicking the way whales hunt. When plotting the graph of "Best Score vs. Iteration" for the WOA, we typically observed that, how the algorithm progresses over iterations in terms of the best fitness score achieved.

At the beginning of the optimization process (early iterations), the algorithm explores the solution space by employing a population of potential solutions (whales).The optimal score may change as the program searches through different parts of the solution space.

The whales begin to converge toward more optimal options as the program runs. This phase is characterized by a gradual improvement in the best score. The algorithm explores various search spaces to find potential regions that may contain optimal solutions. With further iterations, the algorithm focuses more on exploiting the promising regions identified during the exploration phase. The best score is expected to show a more consistent improvement as the algorithm refines its search in the vicinity of potential optimal solutions. Figure 4 shows the fitness function results of WOA.

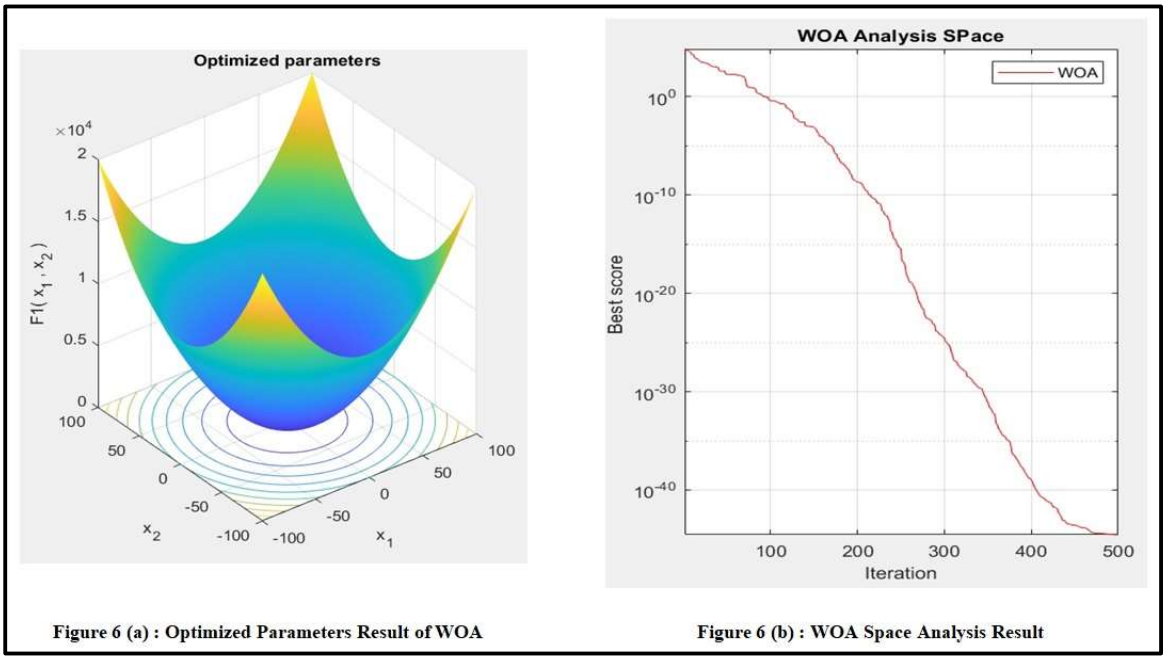


Figure 6 (a) : Optimized Parameters Result of WOA

Figure 6 (b) : WOA Space Analysis Result

Figure 4: Fitness Results Of Whale Optimization Algorithm

In the later stages of the optimization process, the algorithm converges towards the optimal solution or a near-optimal solution. The WOA marginally improves and ideally stabilize in achieving the best score, indicating that the algorithm has found a satisfactory solution.

There may be plateaus or periods where the best score remains relatively constant. This could be due to the algorithm exploring a plateau in the fitness landscape or getting trapped in a local minimum. To overcome this drawback, WOA incorporates a mechanism called as "Diversity Maintenance". This method aids in avoiding local optima and enhances the exploration of the solution space more efficiently. The best score might experience fluctuations during periods of diversity maintenance.

4.3Warning and Alert Messages

Figure 5a indicates that the network connection has been lost.The user can acknowledge the message by clicking OK. This informational alert or a prompt helps the user to take action to resolve the network disconnection.

Figure 5b suggests a network troubleshoot option is available. (1 for "Yes") and (0 for "No"). The user can choose whether or not to initiate network troubleshooting. Responding with 1 might trigger a set of diagnostic tools or procedures to identify and address the network connectivity issue and 0 responds to "Access Denied".

Journal of Theoretical and Applied Information Technology
15th August 2024. Vol.102. No. 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

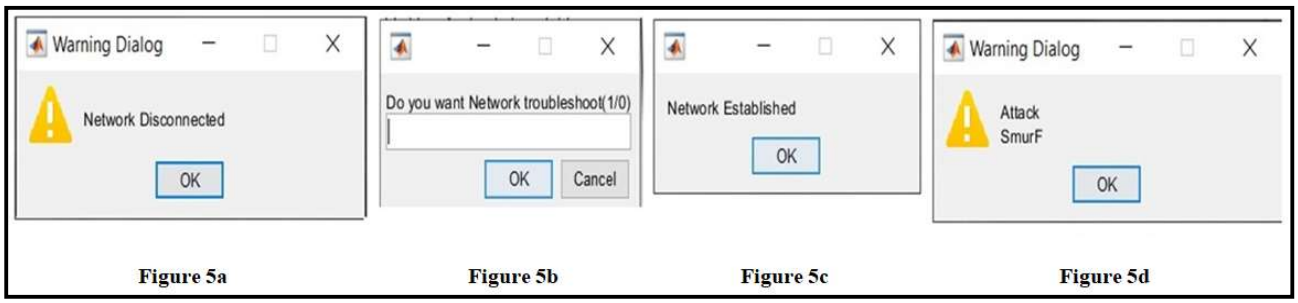


Figure 5a Figure 5b Figure 5c Figure 5d

Figure 5: Warning And Alert Dialog Boxes Of Proposed NIPS Model

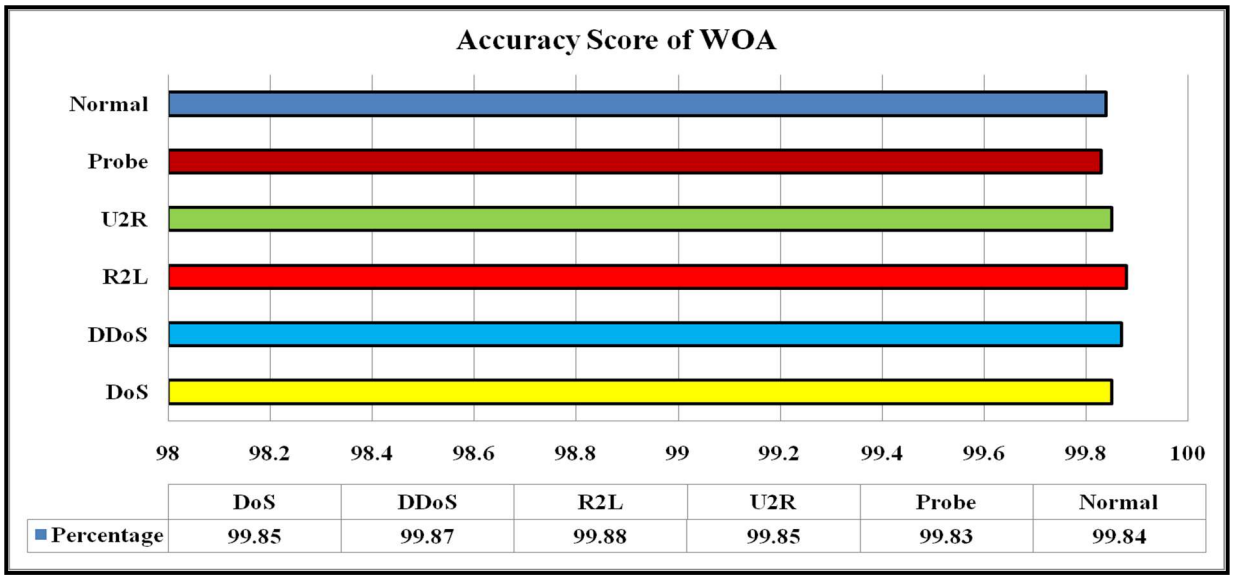
Figure 5c indicates that the network has been successfully reestablished or repaired.The user can acknowledge the message by clicking OK. This confirms that the network is now operational.

Figue 5d indicates the detected Cyber Attack as "SMURF" from the pattern identified.

The system perfroms network disconnection action, offers a troubleshooting option, seeks confirmation for troubleshooting, and provides feedback on the resolution (network reestablished). These messages are part of a user interface or alert system to keep the user informed about the network status and allow them to take necessary actions. The inclusion of an IPS suggests that the system not only monitors but also takes preventive measures against potential intrusions or disruptions in the network.

4.4 Accuracy Results

The figure 6 presents the accuracy scores of the WOA in classifying various network intrusion types, including DoS, DDoS, R2L, U2R, Probeand Normal activities. The higher accuracy scores, 99.85% for DoS, 99.87% forDDoS, 99.88% for R2L, 99.85% for U2R, 99.83% for Probe, and 99.84% for Normal shows that the WOA performs exceptionally well in accurately classifying network traffic. These high accuracy values reflect the algorithm's robustness and reliability in distinguishing between normal and malicious network behaviors, making it a promising approach for intrusion prevention tasks.

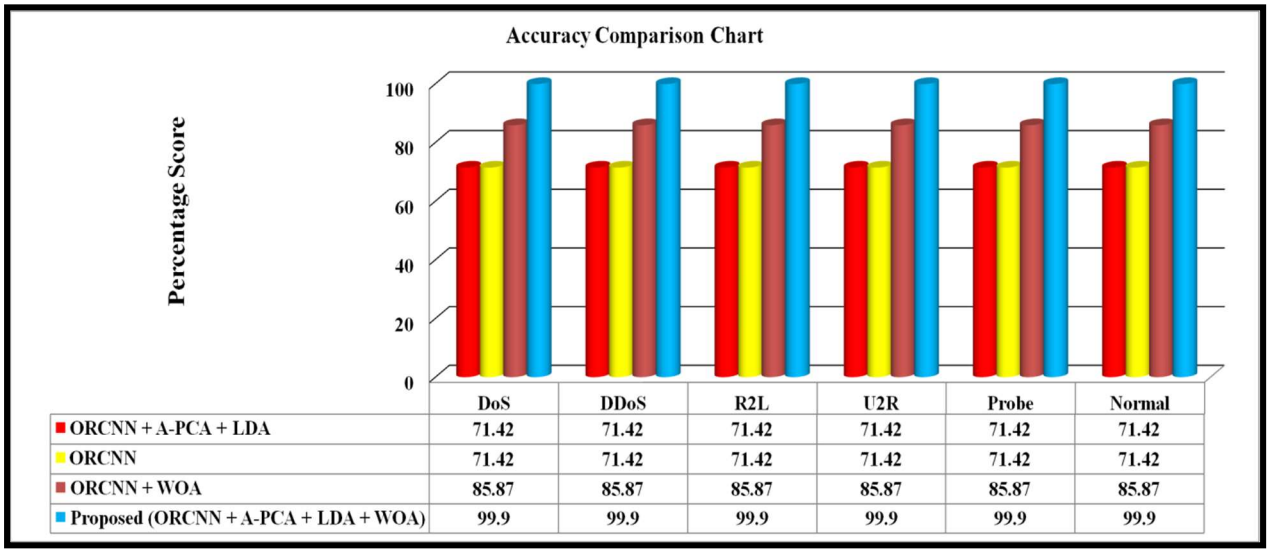


	DoS	DDoS	R2L	U2R	Probe	Normal
Percentage	99.85	99.87	99.88	99.85	99.83	99.84

Figure 6: Accuracy Score Of Whale Optimization Algorithm

4.5 Accuracy Comparison of Proposed Model

The figure 7 presents accuracy scores for various intrusion detection techniques across different attack types, including DoS, DDoS, R2L, U2R, Probe, and Normal. Initially, employing ORCNN combined with A-PCA and LDA yields consistent accuracy scores of 71.42% across all attack types. However, when integrating WOA (Whale Optimization Algorithm), there is a noticeable improvement in accuracy, with scores rising to 85.87% across the board. Notably, the proposed approach, which combines ORCNN with A-PCA, LDA, and WOA, achieves remarkably high accuracy scores of 99.9% for all attack types, demonstrating significant enhancement over previous methods. This suggests that the integration of WOA along with feature reduction techniques like A-PCA and LDA enhances the discriminative ability of the model, leading to highly accurate intrusion detection across a variety of attack types.



	DoS	DDoS	R2L	U2R	Probe	Normal
ORCNN + A-PCA + LDA	71.42	71.42	71.42	71.42	71.42	71.42
ORCNN	71.42	71.42	71.42	71.42	71.42	71.42
ORCNN + WOA	85.87	85.87	85.87	85.87	85.87	85.87
Proposed (ORCNN + A-PCA + LDA + WOA)	99.9	99.9	99.9	99.9	99.9	99.9

Figure 7: Accuracy Comparison Scores Of Proposed NIPS Model

4.6 Performance Metrics Chart

The performance metrics chart shown in figure 8presents high Accuracy, Precision, F-score, and Recall values across various attack types DoS, DDoS, R2L, U2R, Probe and normal instances. The accuracy values are consistently above 99.9%, indicating an overall effective classification of network traffic. Precision values are high, suggesting a low rate of False Positives, while recall values indicate a low rate of False Negatives.

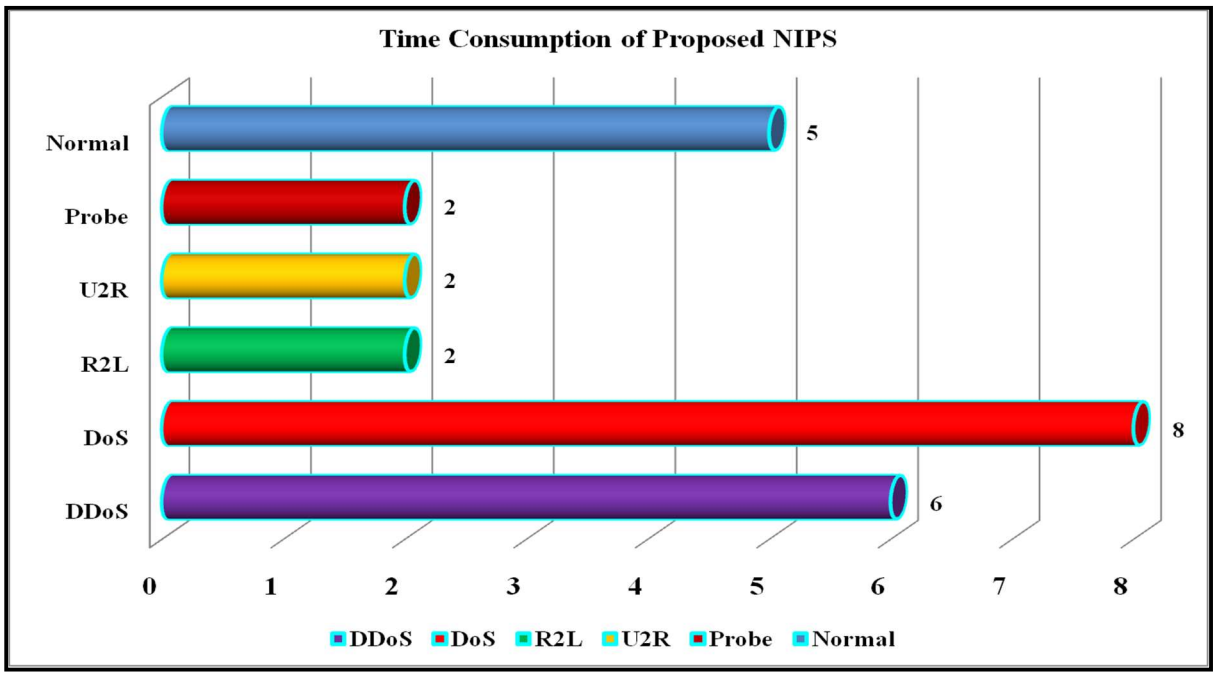
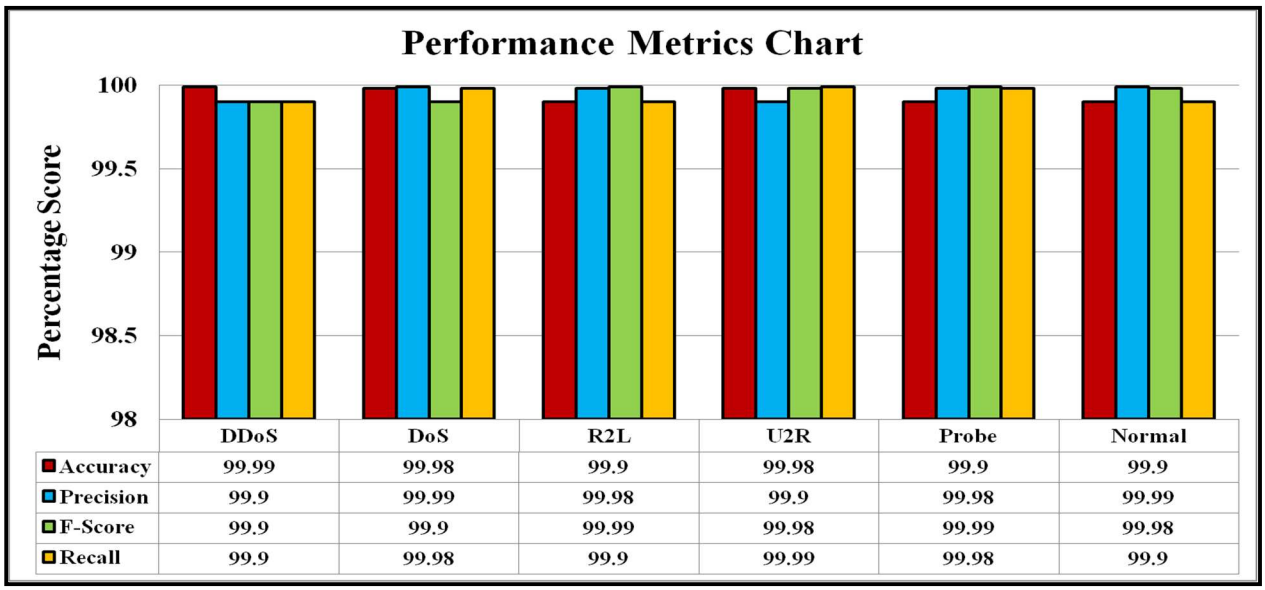


Figure 8: Performance Metrics Chart Of Proposed NIPS Model

Specifically, for the DDoS and Probe categories, precision and recall are both exceptionally high, implying robust prevention capabilities for these attack types. As a measure of the model's balanced performance, the F-scores, which take accuracy and recall into account, are typically near to 100%. All of these measures show that the model is good at identifying typical and unusual occurrences of different types of threats, which proves that it is reliable for use in network security.

4.7 Time Consumption of Proposed NIPS Model

The time consumption graph for the proposed Network Intrusion Prevention System (NIPS) reveals varying processing times associated with different types of network activities. Among the identified categories, DDoS attacks demand the highest processing time at 6 ms, followed closely by DoS attacks at 8 ms. On the other hand, unauthorized access attempts, categorized as R2L and U2R, exhibit lower processing times of 2 ms each. Probe activities also require 2 ms for analysis. Normal network traffic, categorized as benign, has a processing time of 5 ms. This graph implies that the NIPS dedicates more computational resources to mitigating and analyzing potential threats, especially in the case of DDoS and DoS attacks, while efficiently handling normal network traffic with a relatively lower processing time. Figure 9 shows the time consumption of proposed NIPS model.

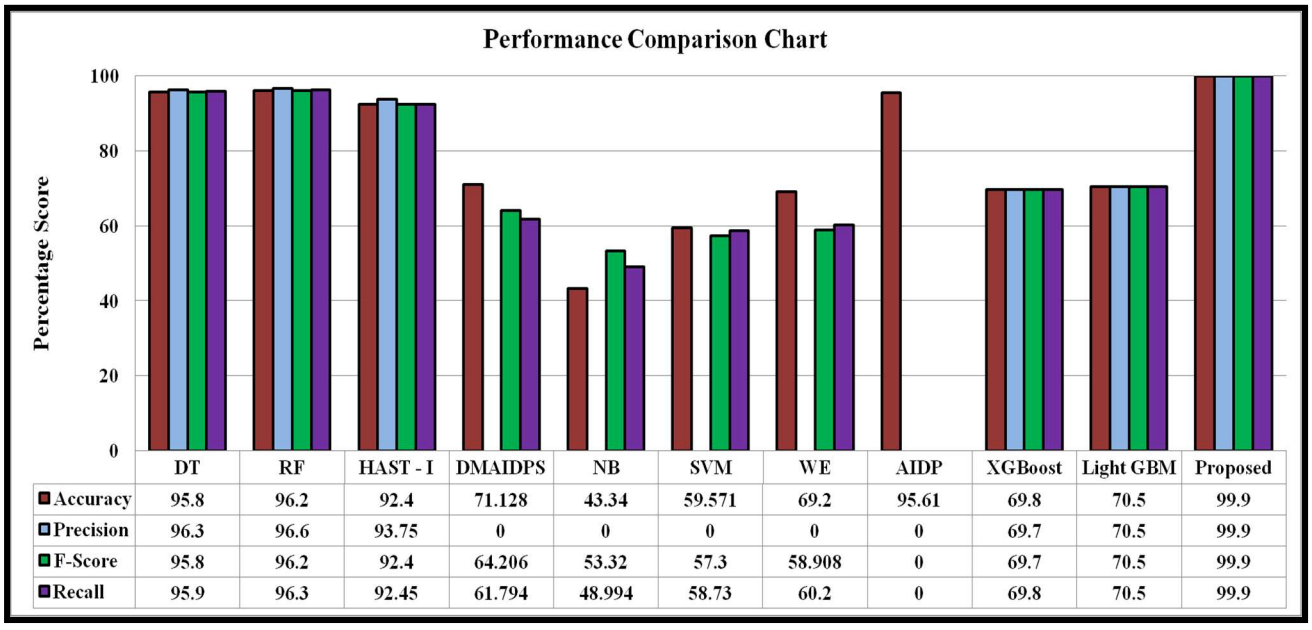


	DDoS	DoS	R2L	U2R	Probe	Normal
Accuracy	99.99	99.98	99.9	99.98	99.9	99.9
Precision	99.9	99.99	99.98	99.9	99.98	99.99
F-Score	99.9	99.9	99.99	99.98	99.99	99.98
Recall	99.9	99.98	99.9	99.99	99.98	99.9

Figure 9: Time Consumption Of Proposed NIPS Model

4.8 Comparison Chart of Proposed NIPS Model with Various Algorithms

Figure 10 presents the performance metrics, including Precision, Accuracy, Recall and F-score for various machine learning models in a classification task. Decision Tree (DT) and Random Forest (RF) exhibit high overall performance with accuracy values of 95.8% and 96.2%, respectively.



	DT	RF	HAST - I	DMAIDPS	NB	SVM	WE	AIDP	XGBoost	Light GBM	Proposed
Accuracy	95.8	96.2	92.4	71.128	43.34	59.571	69.2	95.61	69.8	70.5	99.9
Precision	96.3	96.6	93.75	0	0	0	0	0	69.7	70.5	99.9
F-Score	95.8	96.2	92.4	64.206	53.32	57.3	58.908	0	69.7	70.5	99.9
Recall	95.9	96.3	92.45	61.794	48.994	58.73	60.2	0	69.8	70.5	99.9

Figure 10: Performance Comparison Of Proposed NIPS Model With Other Existing Algorithms

These models also demonstrate excellent precision, recall, and F-score metrics, indicating a balanced performance across different evaluation criteria. While the HAST-I model demonstrates somewhat lower metrics, it still manages to achieve an accuracy of 92.4% and precision, recall, and F-score values of around 93.75%, 92.4%, and 92.45%, respectively. The

DMAIDPS model shows a considerable decline in performance, with an accuracy of 71.128% and an F-score of 64.206% and recall of 61.794%, respectively. Both the SVM and NB models show support vector machines relatively poor performance with accuracy values of 43.34% and 59.571%, respectively, and other metrics in the range of 48.994% to 58.73%. Light GBM also does somewhat better than the WE model and XGBoost, with a 70.5% accuracy rate and a balanced set of precision, recall, and F-score values. Interestingly, the proposed NIPS model stands out with exceptional performance across all metrics, achieving 99.9% accuracy, precision, recall, and F-score, showcasing its superiority in the classification task [40] – [43].

5. CONCLUSION AND FUTURE SCOPE

The utilization of WOA algorithm for convergence, effectively explores the solution space, progressively refines its search, and ultimately converges towards an optimal solution. The incorporation of "Diversity Maintenance" mitigates the risk of getting trapped in local optima, enhancing the algorithm's effectiveness. The warning and alert prompts helps the user to take necessary action which enhances the overall usability of the proposed model. The WOA achieves a high accuracy score of more than 99% in finding the optimal solution followed by the OR-CNN model results in high accuracy of 99.9% indicating a high level of correctness in classifying instances of different network activities. The performance metrics result shows a balanced performance in preventing the Cyber Attacks by minimizing False Positives and False Negatives. The computational time consumed by our proposed model shows an enhanced efficiency by maintaining a very less processing time in preventing the Cyber Attacks. Finally the comparison graph once again proves that the proposed model outperforms other existing ML and DL methods with a remarkable accuracy of 99.9% and a balanced Precision, F-Score and Recall values. In conclusion, the proposed NIPS model, integrating WOA optimization and an OR-CNN technique, stands out as a highly effective and reliable solution for Network Intrusion Prevention System. Further exploration of ensemble techniques incorporating diverse optimization algorithms could potentially enhance the robustness and scalability of the proposed NIPS model, ensuring its effectiveness across a wider range of network environments and cyber threat landscapes.

REFERENCES

[1] Wu, S.X. and Banzhaf, W., 2010. The use of computational intelligence in intrusion detection systems: A review. *Applied soft computing*, *10*(1), pp.1-35. https://doi.org/10.1016/j.asoc.2009.06.019.

[2] Ektefa, M., Memar, S., Sidi, F. and Affendey, L.S., 2010, March. Intrusion detection using data mining techniques. In *2010 International Conference on Information Retrieval & Knowledge Management (CAMP)* (pp. 200-203). IEEE. https://doi.org/10.1109/INFRKM.2010.5466919.

[3] Bilge, L. and Dumitraş, T., 2012, October. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 833-844). https://doi.org/10.1145/2382196.2382284.

[4] Roesch, M., 1999, November. Snort: Lightweight intrusion detection for networks. In *Lisa* (Vol. 99, No. 1, pp. 229-238).

[5] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y. and Zhu, M., 2017. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE access*, *6*, pp.1792-1806. https://doi.org/10.1109/ACCESS.2017.2780250.

[6] Kruegel, C. and Toth, T., 2003, September. Using decision trees to improve signature-based intrusion detection. In *International workshop on recent advances in intrusion detection* (pp. 173-191). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-45248-5_10.

[7] Seo, W. and Pak, W., 2021. Real-time network intrusion prevention system based on hybrid machine learning. *IEEE Access*, *9*, pp.46386-46397. https://doi.org/10.1109/ACCESS.2021.3066620.

[8] Rose, T., Kifayat, K., Abbas, S. and Asim, M., 2020. A hybrid anomaly-based intrusion detection system to improve time complexity in the Internet of Energy environment. *Journal of Parallel and Distributed Computing*, *145*, pp.124-139. https://doi.org/10.1016/j.jpdc.2020.06.012.

[9] Ouiazzane, S., BarramoU, F. and Addou, M., 2020. Towards a multi-agent based network intrusion detection system for a fleet of drones. *International Journal of Advanced Computer Science and Applications*, *11*(10). https://doi.org/10.14569/ijacsa.2020.0111044.

[10] Jayalaxmi, P.L.S., Saha, R., Kumar, G., Conti, M. and Kim, T.H., 2022. Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. *IEEE Access*.

[11] R. Nikhil, B. S. Anisha, and R. Kumar P., ``Real-time monitoring of agricultural land with crop prediction and animal intrusion prevention using Internet of Things and machine learning at edge," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2020, pp. 1_6.

[12] N. R. Sai, A. G. Raghavendra, N. C. M. Deepak, and M. Poojitha, ``A machine learning intrusion prevention and detection system using securing smart grid," *Int. J. Recent Technol. Eng.*, vol. 8, no. 5, pp. 2278_3878, 2020.

[13] A. Krishna, A. J. Mathewkutty, D. S. Jacob, and M. Hari, ``Intrusion detection and prevention system using deep learning," in *Proc. Int. Conf.Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 273_278.

[14] G. Xian, ``Cyber intrusion prevention for large-scale semi-supervised deep learning based on local and non-local regularization," *IEEE Access*, vol. 8, pp. 55526_55539, 2020.

[15] V. Balamurugan and R. Saravanan, ``Enhanced intrusion detection and prevention system on cloud environment using hybrid classi_cation and OTS generation," *Cluster Comput.*, vol. 22, no. S6, pp. 13027_13039,Nov. 2019.

[16] A. Ali and M. M. Yousaf, ``Novel three-tier intrusion detection and prevention system in software defined network," *IEEE Access*, vol. 8, pp. 109662_109676, 2020.

[17] M. Islabudeen and M. K. K. Devi, ``A smart approach for intrusion detection and prevention system in mobile ad hoc networks against security attacks," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 193_224, May 2020.

[18] F. James, ``IoT cybersecurity based smart home intrusion prevention system," in *Proc. 3rd Cyber Secur. Netw. Conf. (CSNet)*, 2019, pp. 107_113, doi: 10.1109/CSNet47905.2019.9108938.

[19] Abdelmoumin G.; Rawat D.B.; Rahman A. On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things. IEEE Internet Things J. 2022, 9, 4280–4290.

[20] Lin W.-H.; Wang P.; Chao K.-M.; Lin H.-C.; Yang Z.-Y.; Lai Y.-H. Deep-Learning Model Selection and Parameter Estimation from a Wind Power Farm in Taiwan. Appl. Sci. 2022, 12, 7067.

[21] Gonc¸alves C.B.; Souza J.R.; Fernandes H. CNN architecture optimization using bio-inspired algorithms for breast cancer detection in infrared images. Comput. Biol. Med. 2022, 142, 105205. https://doi.org/10.1016/j.compbiomed.2021.105205 PMID: 35065408.

[22] Lokku Gurukumar, Harinatha Reddy G., Giri Prasad M.N., OPFaceNet: OPtimized Face Recognition Network for noise and occlusion affected face images using Hyperparameters tuned Convolutional Neural Network, Applied Soft Computing, Volume 117,2022,108365, ISSN 1568-4946.

[23] Wu J.; Chen X.Y.; Zhang H.; Xiong L.D.; Lei H.; Deng S.H. Hyperparameter optimization for machine learning models based on Bayesian optimization. J. Electron. Sci. Technol. 2019, 17, 26–40.

[24] Elmasry W.; Akbulut A.; Zaim A.H. Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Comput. Netw. 2020, 168, 107042.

[25] Kan X.; Fan Y.; Fang Z.; Cao L.; Xiong N.N.; Yang D., et al. A novel IoT network intrusion detection approach based on Adaptive Particle Swarm Optimization Convolutional Neural Network. Inf. Sci. 2021, 568, 147–162.

[26] Alharbi A.; Alosaimi W.; Alyami H.; Rauf H.; Damasˇevičius R. Botnet Attack Detection Using Local Global Best Bat Algorithm for Industrial Internet of Things. Electronics 2021, 10, 1341.

[27] Nematzadeh S.; Kiani F.; Torkamanian-Afshar M.; Aydin N. Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. Comput. Biol. Chem. 2021, 97, 107619. https://doi.org/10.1016/j.compbiolchem. 2021.107619 PMID: 35033837.

[28] Brodzicki A.; Piekarski M.; Jaworek-Korjakowska J. The Whale Optimization Algorithm Approach for Deep Neural Networks. Sensors 2021, 21, 8003. https://doi.org/10.3390/s21238003 PMID: 34884004

[29] Ali M.H.; Jaber M.M.; Abd S.K.; Rehman A.; Awan M.J.; Damasˇevičius R., et al. Threat Analysis and Distributed Denial of Service (DDoS) Attack Recognition in the Internet of Things (IoT). Electronics 2022, 11, 494.

[30] Alzaqebah A.; Aljarah I.; Al-Kadi O.; Damasˇeviˇcius R. A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System. Mathematics 2022, 10, 999.

[31] https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection.

[32] Gao, J.; Chai, S.; Zhang, B.; Xia, Y. Research on Network Intrusion Detection Based on Incremental Extreme Learning Machine and Adaptive Principal Component Analysis. *Energies* **2019**, *12*, 1223. https://doi.org/10.3390/en12071223.

[33] Dini, P.; Elhanashi, A.; Begni, A.; Saponara, S.; Zheng, Q.; Gasmi, K. Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity. *Appl. Sci.* **2023**, *13*, 7507. https://doi.org/10.3390/app13137507.

[34] G. Y. Ning and D. Q. Cao, "Improved whale optimization algorithm for solving constrained optimization problems", *Discret. Dyn. Nat. Soc.*, Vol. 2021, 2021, https://doi.org/10.1155/2021/8832251.

[35] Sunaryono, D., Siswantoro, J., Raharjo, A.B., Ridho, R., Sarno, R., Sabilla, S.I. and Susilo, R.I., 2023. Optimized One-Dimension Convolutional Neural Network for Seizure Classification from EEG Signal based on Whale Optimization Algorithm. *International Journal of Intelligent Engineering and Systems*, *16*(3), pp.310-322.

[36] Ding, T.; Chang, L.; Li, C.; Feng, C.; Zhang, N. A Mixed-Strategy-Based Whale Optimization Algorithm for Parameter Identification of Hydraulic Turbine Governing Systems with a Delayed Water Hammer Effect. *Energies* **2018**, *11*, 2367. https://doi.org/10.3390/en11092367.

[37] Faruqui, N., Yousuf, M.A., Whaiduzzaman, M., Azad, A.K.M., Alyami, S.A., Liò, P., Kabir, M.A. and Moni, M.A., 2023. SafetyMed: a novel IoMT intrusion detection system using CNN-LSTM hybridization. *Electronics*, *12*(17), p.3541. https://doi.org/10.3390/electronics12173541 .

[38] Li, X.; Orabona, F. On the convergence of stochastic gradient descent with adaptive stepsizes. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics PMLR, Naha, Japan, 16–18 April 2019; pp. 983–992.

[39] Awad, M. and Fraihat, S., 2023. Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems. *Journal of*

Sensor and Actuator Networks, *12*(5), p.67. https://doi.org/10.3390/jsan12050067.

[40] Seo, W. and Pak, W., 2021. Real-time network intrusion prevention system based on hybrid machine learning. *IEEE Access*, *9*, pp.46386-46397. https://doi.org/10.1109/ACCESS.2021.3066620.

[41] Javadpour, A., Pinto, P., Ja'fari, F. and Zhang, W., 2023. DMAIDPS: a distributed multi-agent intrusion detection and prevention system for cloud IoT environments. *Cluster Computing*, *26*(1), pp.367-384. https://doi.org/10.1007/s10586-022-03621-3.

[42] Prabu, K. and Sudhakar, P., 2023. An Automated Intrusion Detection and Prevention Model for Enhanced Network Security and Threat Assessment. *Int. J. Comput. Netw. Appl*, *10*(4). : https://doi.org/10.22247/ijcna/2023/223316.

[43] Alnashwan, R.A., Mashaabi, M.O., Alotaibi, A.S., Qudaih, H.M. and ALBRAHEEM, L.I.A., IoT Based Accident Prevention System using Machine Learning techniques, https://www.researchgate.net/publication/369595854