International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybridization of Bottlenose Dolphin Optimization and Artificial Fish Swarm Algorithm with Efficient Classifier for Detecting the Network Intrusion in Internet of Things (IoT)

Rekha Gangula[1], Dr. Murali Mohan Vutukuru[2] and Dr. Ranjeeth Kumar M.[3]

Submitted: 26/09/2023 **Revised**: 17/11/2023 **Accepted**: 29/11/2023

Abstract: Due to the current target-oriented assaults aimed at stealing confidential information from a business, Intrusion Detection Systems (IDSs) research is essential in the field of network security. Intrusion classification and detection are difficult yet highly specialised tasks. The accuracy of intrusion detection in network traffic varies for various methods in the current models. The inter domain dispersion disagreement assessment of the current method, unfortunately, has a higher computing complexity as the sample size rises, that might worsen the strategy's capacity in generalise. We suggest a deep transfer learning method based on 1D-CNN for categorising the incursions in order to resolve the issue. Also, a hybrid bottlenose dolphin optimization/artificial fish swarm technique is described for feature selection that can quickly and effectively detect a variety of intrusion behaviours by learning the information associated with typical intrusion characteristics. Using a character that allows the rough set to retain the original dataset's discernibility after reductions, the unique dataset's reductions were computed then utilised to create a neural network for training, improving detection capability. For the purpose of analysis three benchmark dataset such as KDD Cup' 99, NSL-KDD and UNSW-NB15 are used and shows that the suggested Hyb_DOAFS_1D-CNN achieves 96.4% and 99% of accuracy, 92% and 99% of precision, 99% and 97% of recall, 99.4% and 99% of f1-score.

Keywords- *network intrusion, optimization, convolution neural network, classification, feature selection*

1.Introduction

The Internet of Things (IoT) is a technology that has been employed in recent years and is expanding at a rapid rate. It allows for interaction and exchange of information among various objects over a network, which is advancing new business production technologies [1]. The huge increase in cybersecurity threats has consequently led to numerous issues in a variety of areas, including finances, reputation, regulation, and operating models [2]. Typically, cloud technology is used to store IoT data and is designed as a model for providing customers with a variety of resources and services on demand. The amount of human interaction between consumers and providers is often reduced by cloud computing [3]. It has drawn considerable interest from businesses and users as a result of its amazing features. Yet, there may be a number of challenging concerns with the operating method and security while switching to the cloud computing platform [4]. The sensitive data that is kept on servers in the cloud is a source of vulnerability. Because of the security risk, the cloud computing platform is a target for numerous cybercriminals and intrusions, which discourages many users from utilising or switching to it [5]. The current sharp increase in cyberattacks is caused by a number of factors. One of the primary causes is the availability and ease of use of hacking tools, which enable novice hackers to swiftly attack cloud storage without the use of sophisticated techniques or specialised knowledge [6]. Several academic communities have neglected to address various difficulties in the sphere of cyberattacks, such as intrusion detection systems (IDSs), throughout the past few decades [7]. A number of conventional classifiers have effectively tackled the IDS, which is characterised as a feature selection problem[8,9]. They were utilised to address a variety of challenging optimization problems prior to the metaheuristic (MH) revolution [10]. Particle swarm optimization (PSO) [11], crow search algorithm (CSA) [12], genetic algorithm [13,14], random harmony search algorithm (RHSA) [15], and grey wolf optimizer (GWO) [16] are all IDSs that make use of MH in some way.

The purpose of an Intrusion Detection System (IDS) is to detect and respond to intrusions, malware, and other potential security breaches by continuously monitoring and analyzing network traffic, system activity, and other data sources. An intrusion detection system (IDS) monitors a computer network or system for signs of

[1] *Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522503. gangularekha@gmail.com*

[2] *Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522503. muralimohan.klu@gmail.com*

[3] *Assistant Professor, Department of CSE, Kakatiya Institute of Technology and Science, Warangal, Telangana, India-506015*

intrusion, such as unusual activity or patterns that could only have been caused by an outside threat. Two major categories of IDS exist:

- **Network-based IDS (NIDS):** In order to detect malicious activity, this IDS examines data packets and keeps a close eye on network traffic. Network intrusion detection systems (NIDS) are typically installed at key nodes within a network, such as gateways or routers, and are capable of detecting a wide variety of network-based attacks, such as port scanning, DoS attacks, and some forms of malware activity.
- **Host-based IDS (HIDS):** Host-based intrusion detection systems are deployed on individual hosts (servers, workstations) and are optimized for monitoring host-centric activity. File integrity checks, log analysis, and process tracking are all examples. Unauthorized access attempts and modifications to crucial system files are two examples of assaults that HIDS can detect.

When it comes to securing networks, systems, and applications from intrusion, attack, and other forms of cybercrime, intrusion detection systems (IDS) play a key role. These systems perform continuous monitoring of a network or system to look for suspicious activity and then provide alerts or take other measures to deal with it. Threat detection, real-time monitoring, alert generation, incident response, forensics and analysis, network segmentation and access control, compliance and reporting, intrusion prevention, and continuous improvement are some of the main roles of intrusion detection systems in the security domain. When combined with additional security measures like firewalls, antivirus software, and safe coding techniques, Intrusion Detection Systems (IDSs) provide a multi-layered defense against a variety of cyber attacks.

In this paper, we suggest a brand-new, highly effective IDS model that makes use of both hybrid metaheuristic optimization methods and sophisticated deep learning (DL) techniques. By employing a convolutional neural network (CNN) model, the level in ID-CNN initial swiftly and effectively obtained the characteristics. To recover the relevant features, there really are numerous successive convolutional blocks. Only the extracting features step, which enables the extraction of major characteristics that can describe the raw data in a lower-dimensional space, were carried out using the CNN[17]. Moreover, CNNs are well renowned for their quick training times and capacity to acquire intricate features from simpler designs. In order to extract useful information and detect potentially harmful or intrusive behaviors, the fully linked layer is built in the same way that CNN[18]'s blocks are. The artificial fish swarm algorithm (Hyb_DOAFS) and bottlenose dolphin optimization are two new hybrid optimization techniques that are then suggested as a feature selection tool to enhance IDS's classification outcomes. Due to its many amazing qualities, including the fact that it requires few initial parameters, the Hyb DOAFS method is utilised as it is a relatively new but effective one. The derived material is optional for the initial investigation. Simple and straightforward to use, it is. It is adaptable and legal. As a result, it has been put to the test against with a variety of scientific challenges and benchmarking tasks. In light of this, the work's achievements are just as chooses to follow:

- KDD Cup' 99, NSL-KDD, and UNSW-NB15 are three online datasets from which the information were obtained. The information were then preprocessed using the one-hot and min-max normalisation techniques.
- To choose features, a hybrid optimization method like the Bottlenose Dolphin Optimization and artificial fish swarm algorithm (Hyb DOAFS) is used.
- Using 1D-CNN model, ransomware may be classified into appropriate classes when characteristics are chosen. This method is effective and dependable for identifying advanced threats.

Following this, we present the proposed strategy and methodologies in Section 3, the experimental results and discussion in Section 4, and the chapter's conclusion and directions for further research in Section 5.

2. Related works

One of the greatest techniques for identifying serious attacks on a network is the anomaly-based IDS. The deep learning algorithms are one of the crucial parts of the IDS that it uses to carry out categorization jobs. Investigators evaluate the efficacy of the deep learning - IDS models using several characteristics, such as real positive and real negative rates, to determine such systems' capacity.

Deep neural networks were utilised by the researcher in [19] to extract the best attributes from IoT IDS information. Then, relying on a newly formed SI optimizer known as the Capuchin Search Algorithm, an effective feature selection approach was suggested (CapSA). The gradient-clipping problem is solved by using likely point particle swarm optimization throughout [20], in which the ada boost long-short term memory (ELSTM) technique with recurrent neural network (RNN) was being used. The noted classification model serves to effectively categorise and identify the invasion information from the data samples. The K-Nearest Neighbors (K-NN) algorithm has been applied

[21] to propose a network intrusion detection model for IoT contexts, which improves the data quality and identifies the 10 leading functioning characteristics. By using 5 samples for evaluation and verification, the Exponential Shuffled Shepherded Optimization Algorithm (ExpSSOA) within [22] aims to achieve the same efficiency as current programs. To solve the security issues, the hybrid Hunger Games Search and Remora Optimization Algorithm (HHGS-ROA) was presented by [23]. With an improved detection rate, the suggested technique achieves a reduced false-positive rate (FPR). A brand-new hybrid long short-term memory (LSTM) strategy is put forth in [24]. To extract the temporally and geographically linked elements of the IoT network, a convolutional neural network is used, while an optimized long short-term memory network is used to forecast the program's various attacks. Firefly swarm optimization was used with LSTM to cut down on processing costs and boost predictive performance. Using data from Message Queuing Telemetry Transport (MQTT), [25] suggests a novel method for detecting intrusions. The weighed merged characteristics are generated by fusing the three output features along with the weighting variable. An improved technique was used to set the hyperparameters of the Hybrid Classifier, which performs categorization by including a Fuzzy and a One-Dimensional Convolutional Neural Network. In order to categorise and foresee unexpected cyberattacks, deep neural network (DNN) was utilised on [26] to construct efficient and effective IDS in the IoMT context. By using hyperparameter feature selection algorithms, the connection weights were preprocessed, improved upon, and tweaked. Though the adaptive selection among the most instructive data aspects, [27] provides a lightweight machine learning-based intruder detection approach to minimise the complexity of the massive wireless communication data. Using a convolutional neural network with adaptive particle swarm optimization (APSO-CNN), the authors of [28] offer a novel approach to detecting intrusions into IoT networks. In this case, the PSO technique is used to adaptably optimize the structure parameters of a one-dimensional CNN by changing the inertia weight. Within [29], the researcher created a dataset, normalized it with min-max, then homogenized it with 1-N encoding. The spider monkey optimization (SMO) technique was used to reduce the number of features, and the resulting smaller dataset was then fed into a deep neural network (DNN).

Table-1 Analyzing and contrasting the current practices

Author/year	Method	Advantage	disadvantage
Abd Elaziz et al., (2023)	Capuchin Search Algorithm (CapSA)	More accuracy	There may be isolated issues.
Donkol et al., (2023)	particle swarm optimization	capable of dealing with ongoing information	Bias that vanishes cannot be removed
Mohy-eddine et al., (2023)	K-NN	superior toughness	Architecture is more complex
Pandey et al., (2023)	ExpSSOA	easy to build	maybe reaches a good approximation
Kumar et al.., (2022)	HHGS-ROA	several categorization problems can be solve	selecting virtual networks is challenging
Alqahtani, et al,(2022)	LSTM	Can handle challenging tasks	sluggish training speed
Vijayan et al, (2022)	One-Dimensional Convolutional Neural Network	easy to adjust towards unknowable circumstances	Issue with imbalanced datasets
Maddikunta et al., (2020)	DNN	high efficiency in a given environment & set of parameters	high degree of overall network intricacy
Davahli et al., (2020)	lightweight machine learning	powerful in identifying different dangerous	sluggish computing

		information	
Kan et al., (2021)	APSO-CNN	much rapid computing	The information affects how well the algorithm achieves.
Khare et al., (2020)	SMO-DNN	Less error	More processing time

According to the aforementioned review article, this was determined that difficulties with intrusion detection methods, including a lower false-positive rate, a decrease in computational load, enhanced precision, and efficiency, needed to be addressed. By utilising the 3 main benchmark datasets, the suggested Hyb DOAFS IDCNN classifier approach addresses these problems. The chaotic, unnecessary data is removed using the dimensional reduction technique to improve the intrusion prevention system' rapidity and precision. This really is due to the literature's usage of metaheruistic methods as dimensionality reduction strategies.

3.System model

The recommended design process is depicted in Figure 1. The data is first extracted from the storage system and pre-processed using a one-hot encoding and min-max normalisation procedure. Then, using a hybrid bottlenose dolphin and artificial fish swarm optimization algorithm, the data has been chosen for characteristics from the statistical information, and the principal components have been eventually categorised just use a 1D-CNN with transfer learning model.

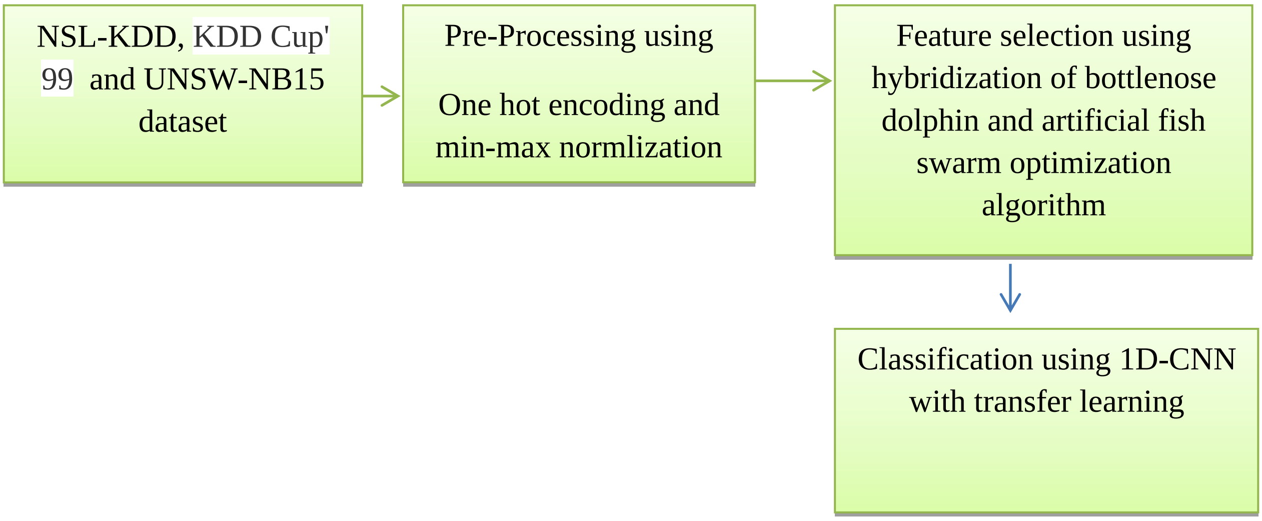
NSL-KDD, KDD Cup' 99 and UNSW-NB15 dataset
Pre-Processing using
One hot encoding and min-max normlization
Feature selection using hybridization of bottlenose dolphin and artificial fish swarm optimization algorithm
Classification using 1D-CNN with transfer learning

Fig-1. block diagram for optimal intrusion detection using optimization method

Dataset description

KDD Cup' 99 constitutes one of the most popular intrusion prevention data sets and is modeled on the DARPA data set. It falls under subcategory alternative because it either uses the standard packet-based structure nor the flow-based structure. The data set lacks IP addresses however includes high-level characteristics such as the number of unsuccessful login attempts as well as fundamental information regarding Data transmission. KDD CUP 99 includes a specific test sample and more than 20 various forms of attacks (such as DoS or buffer overflow). The 7 billion measured values in the data collection are downloadable for free. [30]

The NSL-KDD improves the KDD CUP 99. The high level of duplication in the KDD CUP 99 data collection has drawn significant concern. As a result, the NSL-KDD writers eliminated redundancies from the KDD CUP 99 data set and produced more complex subsets. About 1 million data points make up the final large dataset, which is then separated into subgroups for training and testing intrusion prevention algorithms. It employs the same characteristics as KDD CUP 99 and therefore is classified as other. Nonetheless, it ought to be recognised that the NSL-underlying KDD's internet traffic began in 1998. The data set is accessible to everyone. [31]

UNSW-NB15- It contains both legitimate and malicious network traffic and was produced from over course of 29 hours in a tiny replicated scenario by using

IXIA Tipping Point programme. It consists of following various types of attacks, including viruses, open ports, DoS, vulnerabilities, and sniffers. It is also accessible containing extra properties in flowbased form. The UNSW-NB15 includes divisions for the training and assessment that are already specified. The data set is freely accessible and contains 46 various Internet Protocol (IP) addresses. [32]

Preprocessing of data

In machine learning, the K-Nearest Neighbors (KNN) algorithm is a common and straightforward approach to classification and regression. It is an instance-based and non-parametric learning technique, which means it does not presuppose anything about the data distribution and instead draws inferences directly from the data points themselves.

The OneHotEncoder (OHE) and min-max normalisation stages make up the bulk of the suggested architecture for data preparation. The unlabeled raw is used as a preparatory element. A few arguments, including missing values, strategy, and axis, can indeed be added to the Imputer subclass. For categorization parameters that don't have a hierarchical link, the integer's coding is insufficient. Using this coding and letting the model assume a regular sequence of categories can lead to sluggish performance or surprising outcomes. In this case, one can encode the representation of the number using a onehotencoder. Here, the encoded numeric variables get decoded, and a binary number expression is added for every distinct numerical value. The computation description of data pretreatment seems to be as following, and the suggested data preprocessing is carried out in a Kernel space.

- Load Libraries #Load the packages for Pandas and Sklearn
- Find and open the dataset in CSV file format in step two by importing it.
- Addressing Insufficient Information in the Dataset

a) missing values #detect any set of data numbers that are lacking.

b) approach # Mean, median, or most frequent values are utilised to replace missing data strategically.

- Onehotencoder is used to code categorical data after applying data modification algorithms.
- Phase 5 involves data partitioning into a training set and a test set. A good rule of thumb is to train with 80% of the data and test with the remaining 20%.
- Feature Scaling (Standardization and Grading) (min-max normalization)

The actual data y is transformed linearly into the given range (n_{min}, n_{max})via Min-Max Regularization.

$$y_i = n_{min} + (n_{max} - n_{min}) \times [\frac{y_i - y_{min}}{y_{max} - y_{min}}] \qquad (1)$$

$$y_{max} = \max_{1 \le i \le N} y_i, y_{min} = \min_{1 \le i \le N} y_i, \qquad (2)$$

With this technique, the information is proportionally scaled from (y_{min}, y_{max}) to (n_{min}, n_{max}) . This technique's benefit lies in how it accurately keeps all connections between the values. It wouldn't potentially skew the information in any way.

Feature selection using hybrid optimization algorithm

The information will be cleaned up using feature ranking methods to eliminate features that are redundant and unimportant. Component rating and subgroup selecting are the two main types of algorithms for picking features. All characteristics are scored according to a particular criteria in variable selection, and those that fall short of a predetermined score are eliminated. Whereas the choosing a subset, the best possible subset is sought out, and characteristics were chosen depending on ordering. The search stage, the call phase, the reception phase, and the predation phase are the 4 basic stages that dolphins use to carry out their predatory strategy. The optimal solution to the issue is produced through continuous iteration, and the associated searching connections were created using the behaviour of 4 concurrent phases.

Initialization: Every dolphin in the optimization problem symbolises a workable solution. In this research, the dolphins is described as $dp_i = [y_1, y_2, \dots y_D]K$ $(i = 1,2, \dots J)$ which is a workable D-dimensional solution, wherein J is the amount of dolphins and $y_j(j = 1,2 \dots com)$ is the element of each degree that needs to be optimised. 2 factors relate to the dolphin are the individualized ultimate solution (I) and neighbourhood quality improvement (N).

There seem to be two associated variables $L_i(i = 1,2, \dots J)$ and $N_i(i = 1,2, \dots J)$, for each $dp_i(i = 1,2, \dots J)$, where L_i stands for the optimal solution that the dp_i discovers once, and N_i represents the optimal solution that the dp_i finds independently or obtains from others. The standard for determining if the answer is superior is fitness fit. Several directions are best for bootlenose dolphin locomotion. The first is the distance between dp_i and dp_j, denoted by the names dis_dp_{ij} and named=$|dp_i - dp_j|$ where i, $i, j = 1,2 \dots J, i \neq j$ .The second is the distance between dp_i and N_i, denoted by the names $dist_{N_i} = |dp_i - N_i|, i = 1,2, \dots . J$. The third is the distance between I_i and N_i,, denoted by the names $dist\_IN_i$ and $dist_{IN_i} = \|I_i - N_i\|, i = 1,2, \dots J$

Preying Behavior: This is a fundamental activity in which the fish seeks to relocate the food source; typically, the fish senses the concentration of food in the water to decide whether to move by sight ((vis) or sense (sen) and then selects the propensity. The more (vis) there is, the easier it is for the dolphin to locate the worldwide immense value and congregate. Assume that z_i represents the dolphin's current state and z_j represents a randomly chosen state in z_i 's visual. $dist(z_i, z_j) \leq vis$ are the food intensity ratios (objective function values) related to $z_i \; and \; z_j$, respectively. If $z_i < z_j$ in the maximising problem, it will move in the following direction: $move(z_i < z_j)$. If not, choose a different state z_j at random once more and determine if it fulfills a forward condition. Whenever it fails to fulfill even trying try_num times, it advances a step arbitrarily. The AF can swim randomly when the try_num is tiny, which causes it to veer out of the nearby optimal value field.

Reception stage: The optimal option would be selected by other dolphins when they compare the optimal evidence received to their own optimal solutions. G_i. In Hyb DOAFS, a $N \times N$ order matrix called "transmission time matrix" (trans mat) implements the information flow, along with the preying behaviour as well as leaping activity. The remainder of the discourse in trans mat is represented by $trans\_mat_{i,j}$ rom dp_j to dp_j . If $fitness(N_i) < fitness(N_j)$ and $trans\_mat_{i,j} > \left[\frac{dis\_dp_{ij}}{A.speed}\right]$ are true for N_i, N_j , and, then $trans\_mat_{i,j} = \left[\frac{dis\_dp_{ij}}{A.speed}\right]$ afterwards, $trans\_mat_{i,j} = \left[\frac{dis\_dp_{ij}}{A.speed}\right]$ wherein acc is a characteristic of acceleration that could speed up sound. At this stage, T_2 is labelled as the increased contact time for, $trans\_mat_{i,j}$. N_i is changed to N_j if fitness $fitness\,(N_i) > fitness(N_j)$, ; else, N_i stays the very same. The determined by the distance of a Hyb DOAFS's companions, $avg\_dist(i)$, must first be determined before the effective and strategic of those companions can be determined.

$$avg\_dist(i) = \frac{\sum_j dist(N_i, N_i)}{npf_i}, \text{ such that } dis(N_i, N_j) < vis \quad (3)$$

The Hyb_DOAFS with the euclidean distance that is nearest to the average distance would then be chosen.

$$c = argmin_j \, \{avg_{dist(i)} - dis(N_i, N_j)\} \forall j \; subjet \; to: dis(N_i, N_j) \geq avg_{dist(i)} \quad (4)$$

Such combination methods promote improved exploration, cost-effective search space research, avoiding convergence rate, and the potential of becoming bound by local optima.

Classification of malwares

An CNN architecture, known as a 1-dimensional convolutional neural network, is primarily used to analyse time information. Although 1D CNNs have indeed produced cutting-edge results on practical challenges, the 2D CNN model has a processing gain. With a 1D CNN, computational complexity might be decreased to $O\,(a,$ dim for the same a and dim parameters as above. The main layer of CNN is called the convolution level, and it has a number of learnable components. The much more crucial aspect of the convolution layer is value distribution. It is used to reduce the computing burden and optimise the network settings to prevent overfitting brought on by too many configurations.

$$y_j^n = f(\textstyle\sum_{k \in M_j} y_k^{l-1} * R_{kj}^l + bias_j^l) \quad (5)$$

Where, M and R signify the collection of attribute values and the gaussian kernel, correspondingly, and $f(.)$ and $bias$ denote the nonlinear activation function and the bias term, respectively, and y_j^n signifies the n th feature in the j layer. ReLU is a frequently used activation function and is spelled out as follows:

$$ReLU(x) = maximum(0, y) \quad (6)$$

Typically, the down sampling procedure is carried out by the pooling layer (PL). The major goals of PL are to prevent overfitting, increase the generalizability of the model, and decrease the parameters of the neural network while keeping the representative features. The following can be done to implement a PL procedure:

$$y_j^{n+1} = pool(\boldsymbol{y_j^n}) \quad (7)$$

When $pool(\boldsymbol{y_j^n})$ denotes the action of pooling and y_j^{n+1} denotes the j th feature of the $n+1$ layer, accordingly. The "Descriptor" in the entire neuron is played by the full connection level. At the inputs of the fully linked layer, the result of the final pooling layer will be first extended to create a one-dimensional feature space. When the inputs and outcomes are fully coupled, ReLU serves as the hidden layer's activation function. Lastly, the output layer is calculated using the Softmax function, and the results are as continues to follow:

$$y^n = f(wei^n y^{n-1} + bias^l) \quad (8)$$

where $f(\bullet)$ stands for the non linear activation function, and wei^n and $bias^l$ stand for the weight and bias of the entire connection layer, correspondingly. ReLU was typically employed as the kernel function because the n layer was the hidden layer, but when the $n+\;1$ level was the output of the system, the activation function was switched to Softmax which was provided as

$$soft_\theta(y) = prob(y^n = 1y^n;\theta); prob(y^n = 2y^n;\theta) \ldots..prob(y^i = ky^i;\theta)$$

$$=\frac{1}{\sum_{j=1}^{n} e^j}(k) \quad (9)$$

Whereas is the model's variables, $prob(y^i = 1y^i;\theta)$ denotes the likelihood that input ith sampling attribute y^i corresponds to subcategory θ and $\sum_{j=1}^{n} e^j$ is used to normalise the residual probabilistic model to ensure that the total of all likelihood equals 1.

4. Performance Analysis

As the training dataset for the competition, this dataset contains 11,4780 incidents and nine malware families. This collection contains 25 malware families and 5394 incidents. These datasets were divided into a training set (80%), a validation set (10%), and a testing set (10%) for the research. To choose the best breakpoint during retraining, the test dataset is being used. Once the ideal breakpoint has been found, researchers use the testing data and the learnt modeling to determine the testing accuracy. Designers perform the randomized divisions ten times and then provide the accuracy rate and the accompanying variance to evaluate the resilience of each strategy.

Table-2 description about dataset

dataset	Split	U2R	DoS	R2L	probe	normal
KDD Cup' 99	Test	45	45,792	537	22,347	4567
	Train	89	65,235	589	34,28	2389
NSL-KDD	Test	89	71,458	389	445	5489
	Train	91	157,398	574	5,389	2354
UNSW-NB15	Test	43	458,249	348	5,245	4590
	Train	89	78,234	524	56,234	8024

The performance of our proposed Hyb_DOAFS_IDCNN: Hybridization of Bottlenose Dolphin Optimization and artificial fish swarm algorithm with 1D-Convolution Neural Network classifier is carried out using parameters such as accuracy, precision, recall, F1-score. These parameters are analyzed for three datasets

- **Accuracy:** It is a metric for evaluating the overall accuracy of a deep learning model. The accuracy with which classifier models can foretell whether or not an incident will occur is measured in terms of true positive (TP) and true negative (TN). The number of inaccurate predictions made by the models is represented by the sum of the false positive (FP) and false negative (FN) rates.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Table-3 analysis of accuracy

	test			train		
class	KDD Cup' 99	NSL-KDD	UNSW-NB15	KDD Cup' 99	NSL-KDD	UNSW-NB15
U2R	90.8	78.6	96.4	98.4	79.4	82.4
DoS	89.4	78	97	99	76.3	86
R2L	89	78.4	97.3	97.4	77	85.3
probe	90.4	76	96	97	77.9	82

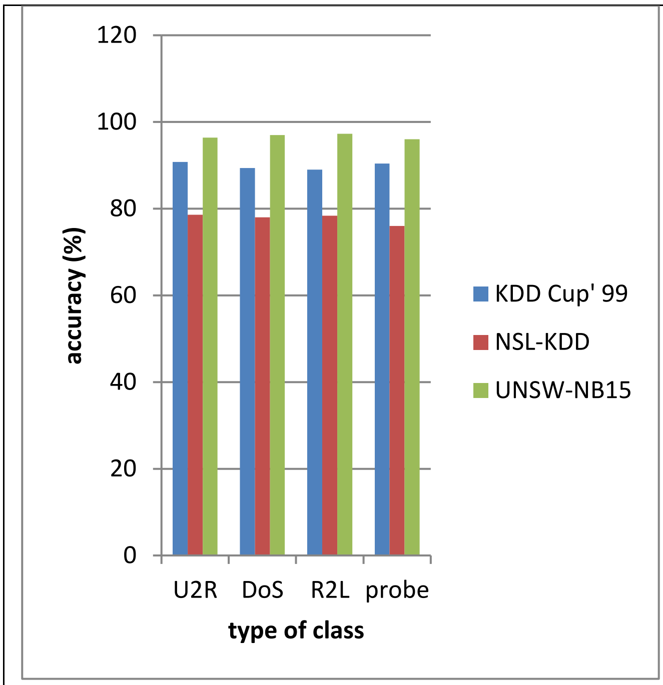

Fig-2 comparison of testing accuracy

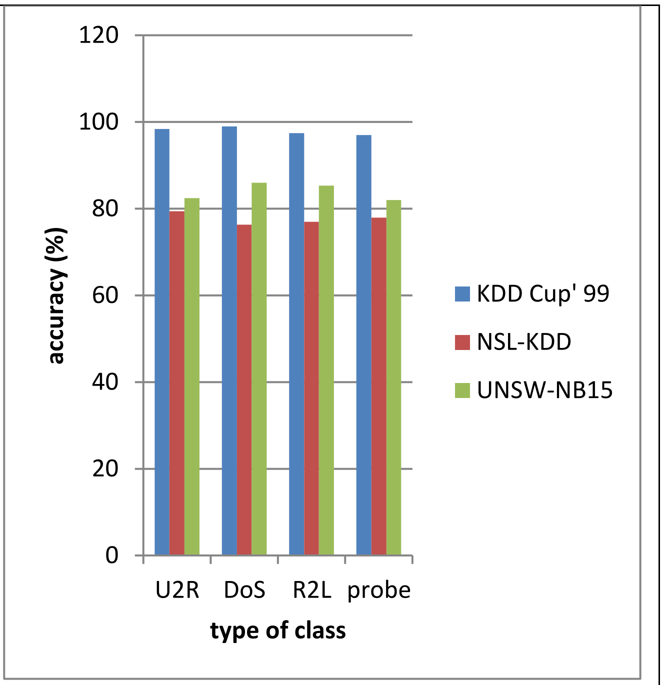

Fig-3 comparison of training accuracy

Figure 2 and 3 indicates testing accuracy and training accuracy for the proposed Hyb_DOAFS_IDCNN X axis and Y axis shows that various classes and the values obtained in percentage respectively. When compared, KDD Cup' 99, NSL-KDD and UNSW-NB15 achieves 90%,78% and 97% of testing accuracy with 98%, 77% and 85% of training accuracy. It is found that the proposed method is better for UNSW-NB15 and KDD Cup' 99 dataset.

- **Precision:** The accuracy of the assault type classification system is evaluated. Accuracy refers to how well a classifier can determine whether or not a disease is present. The true positive (TP) rate, which may be computed using equation (11), describes how often a test result is correct.

$$Precision\ (P) = \frac{TP}{TP+FP} \tag{11}$$

Table-3 analysis of precision

	test			train		
class	**KDD Cup' 99**	**NSL-KDD**	**UNSW-NB15**	**KDD Cup' 99**	**NSL-KDD**	**UNSW-NB15**
U2R	90	89	71	89	79	98
DoS	91	89.6	72	89.6	79.6	98
R2L	92.3	87	72.5	87	79	99.5
probe	92	85	75	87.4	76	98

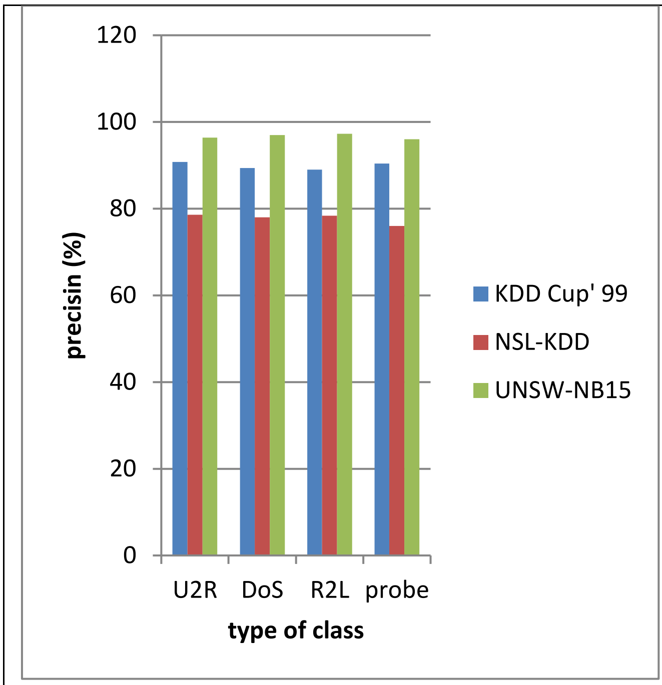

Fig-4 comparison of testing precision

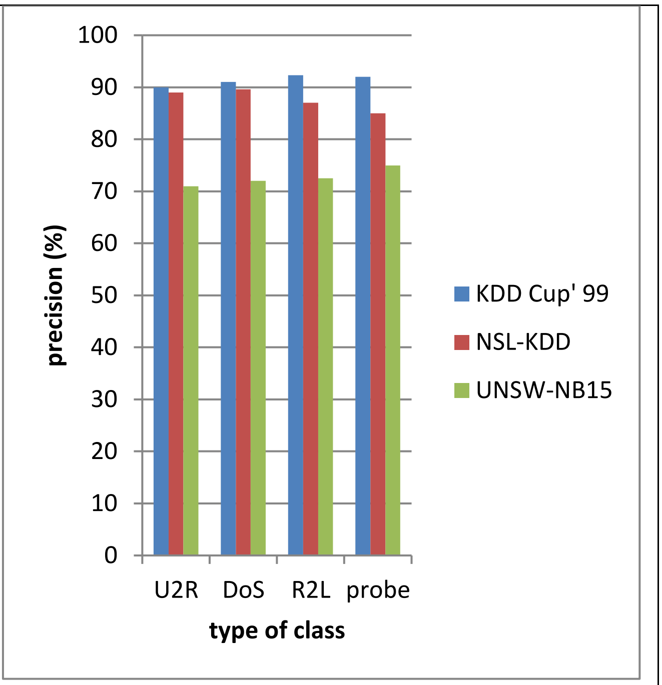

Fig-5 comparison of training precision

Figure 4 and 5 indicates testing precision and training precision for the proposed Hyb_DOAFS_IDCNN X axis and Y axis shows that various classes and the values obtained in percentage respectively. When compared, KDD Cup' 99, NSL-KDD and UNSW-NB15 achieves 92%,87% and 72% of testing precision with 89%, 79% and 99% of training precision. It is found that the proposed method is better for UNSW-NB15 and KDD Cup' 99 dataset.

- **Recall:** In the absence of data, it is the probability that a classifier will make an accurate negative prediction. It can be determined using the true negative (TN) rate formula (12).

$$Recall(R) = \frac{TP}{TP+FN} \quad (12)$$

Table-4 analysis of recall

	test			**train**		
class	**KDD Cup' 99**	**NSL-KDD**	**UNSW-NB15**	**KDD Cup' 99**	**NSL-KDD**	**UNSW-NB15**
U2R	78	66	99	81	71	97
DoS	78.5	68.5	98.6	82.3	72.4	97
R2L	78	69.4	99	82	72	97.4
probe	77.4	68	99.4	83	74	96

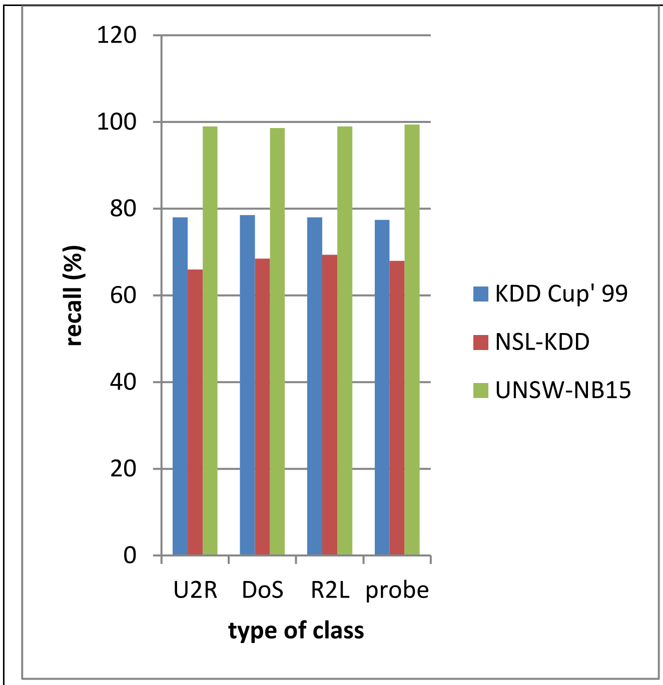

Fig- 6 comparison of testing recall

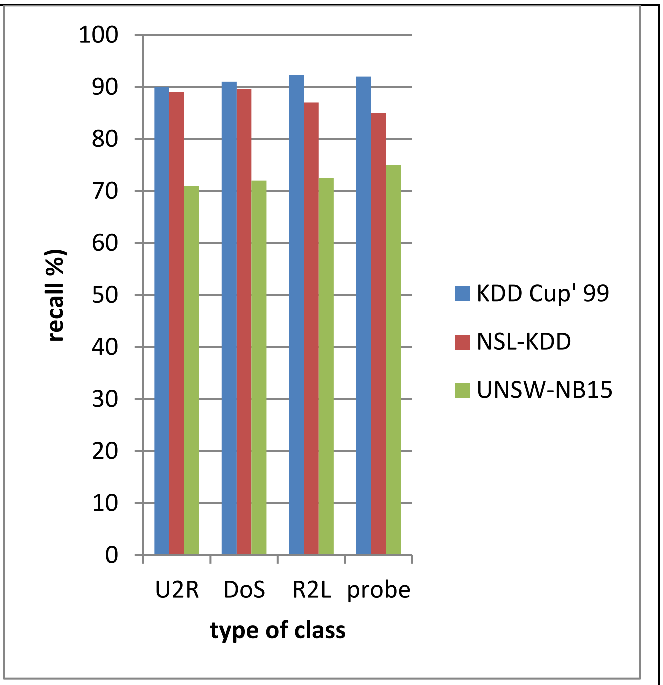

Fig-7 comparison of training recall

Figure 6 and 7 indicates testing precision and training precision for the proposed Hyb_DOAFS_IDCNN X axis and Y axis shows that various classes and the values obtained in percentage respectively. When compared, KDD Cup' 99, NSL-KDD and UNSW-NB15 achieves 78%,69% and 99% of testing accuracy with 82%, 72.5% and 97% of training accuracy. It is found that the proposed method is better for UNSW-NB15 dataset.

- The F1-Score measures how well the forecast was made. It is the sum (or averaged harmony) of the accuracy and recall rates. The highest possible score is 1, while the lowest is 0. When calculating F-values, the TNs are ignored. The formula (13) can be used to determine the F1-Score:

$$F1 - Score = \frac{2*P*R}{P+R} \quad (13)$$

Table-4 analysis of f1-score

	test			train		
class	**KDD Cup' 99**	**NSL-KDD**	**UNSW-NB15**	**KDD Cup' 99**	**NSL-KDD**	**UNSW-NB15**
U2R	75	98	87	89	96	89
DoS	75.6	99.7	87.5	89.4	96.4	89.44
R2L	75	98	85	89	96.9	89
probe	74	99	85.9	89.3	96	89.5

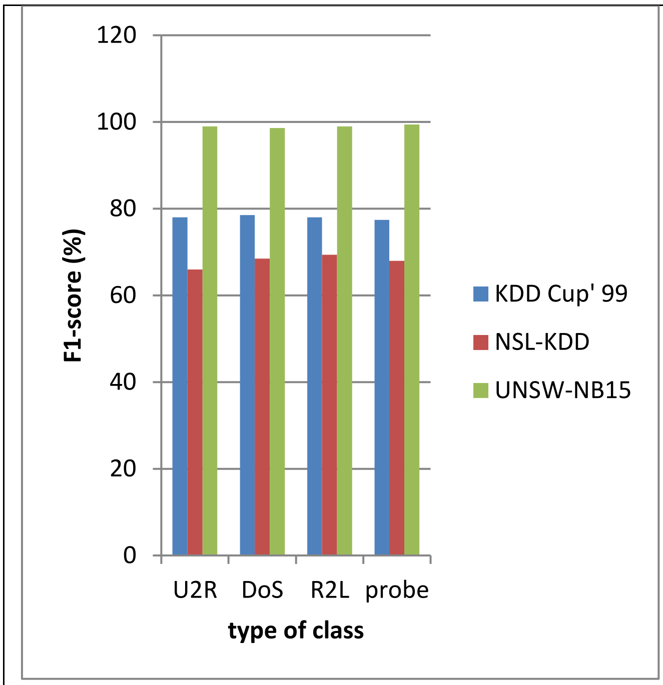

Fig- 8 comparison of testing f1-score

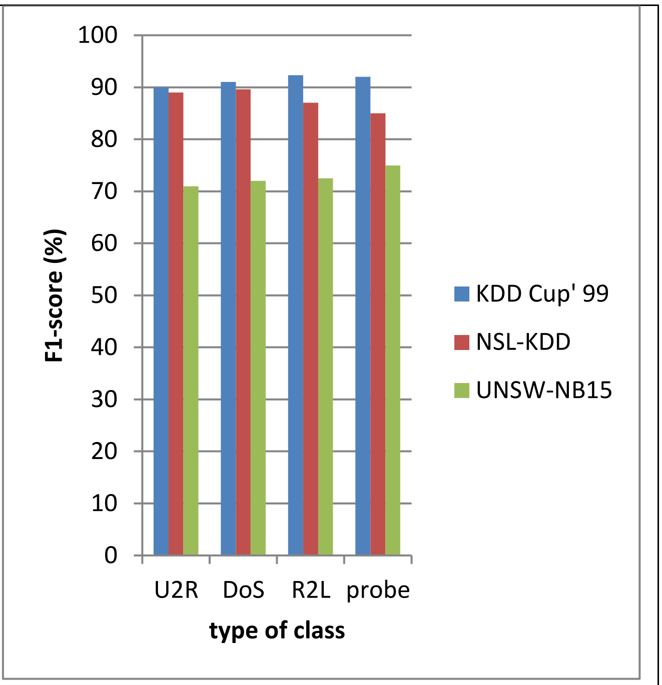

Fig-9 comparison of training f1-score

Figure 8 and 9 indicates testing f1-score and training f1-score for the proposed Hyb_DOAFS_IDCNN X axis and Y axis shows that various classes and the values obtained in percentage respectively. When compared, KDD Cup' 99, NSL-KDD and UNSW-NB15 achieves 75%,99% and 87% of testing accuracy with 89%, 96% and 89% of training accuracy. It is found that the proposed method is better for NSL-KDD dataset.

5. Conclusion

The initial of its kind new Hyb DOAFS IDCNN intrusion detection model is given in this research paper. To use the benchmark datasets KDD Cup' 99, NSL-KDD, and UNSW-NB15, the suggested approach was tested and implemented. This method makes use of the hybrid optimizer's benefits to reduce the dimension, and then it uses a deep neural network to do binary classification. Accuracy, precision, recall, and F1-score were used as performance indicators in the assessments of the suggested model. Also, the suggested Hyb DOAFS IDCNN model's effectiveness is determined using the IDCNN's time training. The proposed model has the limitation of being applicable solely to classification tasks. Multiclass classification expansion will allow for future evaluation of the work's efficacy.

Conflicts of Interest: The authors declare no conflict of interest

Reference

[1] I. Lee, "'e internet of things for enterprises: an ecosystem, architecture, and iot service business model," Internet of Hings, vol. 7, Article ID 100078, 2019.

[2] I. Lee, "Internet of things (iot) cybersecurity: literature review and iot cyber risk management," Future Internet, vol. 12, no. 9, p. 157, 2020.

[3] G. S. Kushwah and V. Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing," Journal of Information Security and Applications, vol. 53, Article ID 102532, 2020.

[4] P. Louvieris, N. Clewley, and X. Liu, "Effects-based feature identification for network intrusion detection," Neurocomputing, vol. 121, pp. 265–273, 2013.

[5] J. Man and G. Sun, "A residual learning-based network intrusion detection system," Security and Communication Networks, vol. 2021, Article ID 5593435, 9 pages, 2021.

[6] B. Arandian, and A. Chapnevis, "Covid-19 diagnosis using capsule network and fuzzy-means and mayfly optimization algorithm," BioMed Research International, vol. 2021, Article ID 2295920, 11 pages, 2021.

[7] L. Mohammadpour, T C. Ling, C. L. Sun, and A. Aryanfar, "A Mean Convolutional Layer for Intrusion Detection System," Security and

Communication Networks, vol. 2020, Article ID 8891185, 16 pages, 2020.

[8] B. Zhou and B. Arandian, “An improved cnn architecture to diagnose skin cancer in dermoscopic images based on wildebeest herd optimization algorithm,” Computational Intelligence and Neuroscience, vol. 2021, Article ID 7567870, 9 pages, 2021.

[9] P. Mishra, V. Varadharajan, and U. Tupakula, “Intrusion detection techniques in cloud environment: a survey,” Journal of Network and Computer Applications, vol. 77, pp. 18–47, 2017.

[10] C. Modi, D. Patel, B. Borisanya, A. Patel, and M. Rajarajan, “A novel framework for intrusion detection in cloud,” in Proceedings of the fifth international conference on security of information and networks, Association for Computing Machinery, Jaipur, India, pp. 67–74, October 2012

[11] P. Ghosh, A. Karmakar, J. Sharma, and S. Phadikar, “Cs-pso based intrusion detection system in cloud environment,” in Emerging Technologies in Data Mining and Information Security, pp. 261–269, Springer, New York, NY, USA, 2019.

[12] R. SaiSindhu'eja and G. K. Shyam, “An efficient metaheuristic algorithm based feature selection and recurrent neural network for dos attack detection in cloud computing environment,” Applied Soft Computing, vol. 100, Article ID 106997, 2021.

[13] M. T. Nguyen and K. Kim, “Genetic convolutional neural network for intrusion detection systems,” Future Generation Computer Systems, vol. 113, pp. 418–427, 2020.

[14] N. Somu, K. Kirthivasan, and R. Liscano, “An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine,” Knowledge-Based Systems, vol. 134, pp. 1–12, 2017.

[15] S. Malhotra, V. Bali, and K. Paliwal, “Genetic programming and k-nearest neighbour classifier based intrusion detection model,” in Proceedings of the 2017 7th International Conference on Cloud Computing, Data Science & EngineeringConfluence, pages, pp. 42–46, IEEE, Noida, India, January 2017.

[16] M. Mayuranathan, M. Murugan, and V. Dhanakoti, “Best features based intrusion detection system by rbm model for detecting ddos in cloud environment,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 3, pp. 3609–3619, 2019.

[17] S. Sharma, A. Gupta, and S. Agrawal, “An intrusion detection system for detecting denial-of-service attack in cloud using artificial bee colony,” in Proceedings of the International Congress on Information and Communication Technology, pp. 137–145, Springer, Bangkok, 'ailand, December 2016.

[18] T. Dash, “A study on intrusion detection using neural networks trained with evolutionary algorithms,” Soft Computing, vol. 21, no. 10, pp. 2687–2700, 2017.

[19] Abd Elaziz, M., Al-qaness, M. A., Dahou, A., Ibrahim, R. A., & Abd El-Latif, A. A. (2023). Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm. *Advances in Engineering Software*, 103402.

[20] Donkol, A. A., Hafez, A. G., Hussein, A. I., & Mabrook, M. M. (2023). Optimization of Intrusion Detection Using Likely Point PSO and Enhanced LSTM-RNN Hybrid Technique in communication Networks. *IEEE Access*.

[21] Mohy-eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. *Multimedia Tools and Applications*, 1-19.

[22] Pandey, B. K., Veeramanickam, M. R. M., Ahmad, S., Rodriguez, C., & Esenarro, D. (2023). ExpSSOA-Deep maxout: Exponential Shuffled shepherd optimization based Deep maxout network for intrusion detection using big data in cloud computing framework. *Computers & Security*, *124*, 102975.

[23] Kumar, R., Malik, A., & Ranga, V. (2022). An intellectual intrusion detection system using Hybrid Hunger Games Search and Remora Optimization Algorithm for IoT wireless networks. *Knowledge-Based Systems*, *256*, 109762.

[24] Alqahtani, A. S. (2022). FSO-LSTM IDS: Hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks. *The Journal of Supercomputing*, *78*(7), 9438-9455.

[25] Vijayan, P. M., & Sundar, S. (2022). Hybrid MQTTNet: An Intrusion Detection System Using Heuristic-Based Optimal Feature Integration and Hybrid Fuzzy with 1DCNN. *Cybernetics and Systems*, 1-34.

[26] RM, S. P., Maddikunta, P. K. R., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. *Computer Communications*, *160*, 139-149.

[27] Davahli, A., Shamsi, M., & Abaei, G. (2020). Hybridizing genetic algorithm and grey wolf

optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks. *Journal of Ambient Intelligence and Humanized Computing*, *11*, 5581-5609.

[28] Kan, X., Fan, Y., Fang, Z., Cao, L., Xiong, N. N., Yang, D., & Li, X. (2021). A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network. *Information Sciences*, *568*, 147-162.

[29] Khare, N., Devan, P., Chowdhary, C. L., Bhattacharya, S., Singh, G., Singh, S., & Yoon, B. (2020). Smo-dnn: Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection. *Electronics*, *9*(4), 692.

[30] N. Moustafa, J. Slay, UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems, in: Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942

[31] A. Shiravi, H. Shiravi, M. Tavallaee, A. A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for 16 intrusion detection, Computers & Security 31 (3) (2012) 357–374. doi:10.1016/j.cose.2011.12.012

[32] A. S. Tanenbaum, D. Wetherall, Computer Networks, 5th Edition, Pearson, 2011