International Journal of Digital Crime and Forensics
Volume 13 • Issue 6 • November-December 2021

An Optimal NIDS for VCN Using Feature Selection and Deep Learning Technique: IDS for VCN

Pankaj Kumar Keserwani, National Institute of Technology, Sikkim, India
Mahesh Chandra Govil, National Institute of Technology, Sikkim, India
E. S. Pilli, Malaviya National Institute of Technology, Jaipur, India
Prajjval Govil, J. K. Lakshmipat University, India

ABSTRACT

In this modern era, due to demand for cloud environments in business, the size, complexity, and chance of attacks to virtual cloud network (VCN) are increased. The protection of VCN is required to maintain the faith of the cloud users. Intrusion detection is essential to secure any network. The existing approaches that use the conventional neural network cannot utilize all information for identifying the intrusions. In this paper, the anomaly-based NIDS for VCN is proposed. For feature selection, grey wolf optimization (GWO) is hybridized with a bald eagle search (BES) algorithm. For classification, a deep learning approach—deep sparse auto-encoder (DSAE)—is employed. In this way, this paper proposes a NIDS model for VCN named GWO-DES-DSAE. The proposed system is simulated in the python programming environment. The proposed NIDS model's performance is compared with other recent approaches for both binary and multi-class classification on the considered datasets—NSL-KDD, UNSW-NB15, and CICIDS 2017—and found better than other methods.

KEYWORDS

Dataset, Deep Learning, Feature Selection, Intrusion Detection, Security

1. INTRODUCTION

Cyberspace refers to a complex environment that runs with the support of Information Communication Technology (ICT) devices and networks where several interactions are carried out among people, software, and services. A wide variety of attacks or incidents may occur intentionally or accidentally, natural or mandate. Cybersecurity in the various networked environments has become one of the prime anxieties in this advanced technical environment like a cloud computing environment. The Cloud computing environment utilizes virtualization, integrated tools, and techniques to run the services via standard Internet protocols. Many vulnerabilities are involved in the cloud computing environment, attracting intruders to explore and exploit different attacks. Already existing cloud computing attacks are Address Resolution Protocol (ARP poisoning), IP spoofing, IP Flooding, Domain Name Service (DNS) poisoning, Routing Information Protocol (RIP) attack, Denial of Service (DoS) attack, and Distributed Denial of Service (DDoS) attack. The Firewall provides security from outside attacks, but it fails to provide security against insider attacks.

DOI: 10.4018/IJDCF.20211101.oa10

IDSs can detect malicious activities or intrusions or attacks originated from a system or Internet that harm the network or systems (Selvakumar et al. 2019). The prerequisite of the IDS is high recall, precision, accuracy, and low False Alarm Rate (FAR) in identifying the intrusions or attacks. The IDS uses so many Machine Learning (ML) as well as Deep Learning (DL) based algorithms such as Decision Tree (DT), Support Vector Machines (SVMs), clustering, Artificial Neural Network (ANN), Deep Neural Network (DNN), auto-encoders, Deep Belief Network (DBN), etc. (Zhang et al. 2018). In general, the IDSs are of two types, which are Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS is designated only for one system to analyze its various by accessing and analyzing data from admin files such as logs and config files. It also creates a backup for the config files for restoring against any malicious attack. NIDS examines network traffic to identify any malicious events. It includes a packet sniffer collect and stores the network traffic data for further analysis. NIDS is dynamic, where the rules can be modified as per the requirements, such as capturing selective data for analysis, adding rules only for HTTP or FTP traffics. HIDS or NIDS are further classified broadly in two types – Signature-based and Anomaly-based. Signature-based NIDS tries to match a specific intrusion signature or pattern which are available in its database. It requires regular updates to combats the new attacks. As the size of the database increases, it demands a higher processing cost for analyzing each attack as the size of the signature database increases. In the case of anomaly detection, a normal network distribution pattern is calculated, and if the network packet deviates from the calculated pattern, it is considered an anomaly. It means that an anomaly-based NIDS first builds the profile for normal behaviors from valid network traffic and compares it with the other profiles to assign the score to the new coming profile. If the score crosses the defined threshold, the NIDS model indicates the occurrence of an anomaly. The profiling methods are generally based on machine learning and statistical data mining techniques (Alomari and Othman 2012). The model trained through profiling can detect the new type of attacks but vulnerable to high FAR than signature-based IDS. However, anomaly-based NIDS is useful for predicting a new kind of attack when someone is probing a network prior to the attack. It is used as the first primary and main security tool to monitor a network (Modi et al. 2013) (AlKadi et al. 2019). The NIDS sends alerts to the network administrator in case of intrusion detection or violation of the defined policy (AlKadi et al. 2019).

In recent years, the research community has introduced many approaches for intrusion detection and anomaly detection, where ML techniques, ensemble techniques, DL techniques, and shallow learning techniques are utilized. DL can be realized as the next revolution of the ML. It works as a subset of ML and now becomes a hot topic in research to be applied in different areas such as medical, banking, trading, natural language processing (NLP), speech processing, image to text conversion, etc. DL-based methods can extract latent features of high levels automatically (Mahmud et al. 2019). One of the fields is network intrusion detection (Moustafa and Slay 2015), where DL methods are very supportive for improving the overall performance of any IDS (Dong and Wang 2016)(Yin et al. 2017). DL-based techniques can detect various patterns from cloud network data sequences (Loukas et al. 2017) and provide more effective results than traditional approaches. The modern threats in the VCN are more sophisticated, which creates a challenge in the detection mechanism. Hence, an effective IDS design is necessary to secure the VCN environment.

This paper proposes a method for boosting NIDS's performance by employing anomaly-based detection for VCN since all the unknown threats can be detected through anomaly-based detection. A sub-vector has been used in place of a long feature vector, which reduces the data size and leads to improved performance. Our intelligent methodology provides scalable access to feature selection. The feature selection is carried out using the combination of Grey Wolf Optimization (GWO) and Bald Eagle Search (BES) Algorithms as hybrid GWO-BES, which are nature-inspired metaheuristic optimization algorithms. An advanced deep learning module called Deep Sparse Autoencoder (DSAE) has been utilized to learn the underlying traffic data structure. The proposed system improves performance and, hence producing reliable predictions. Evaluation of the results shows the quality and effectiveness of the proposed NIDS model, and the main contributions of this work are as follows:

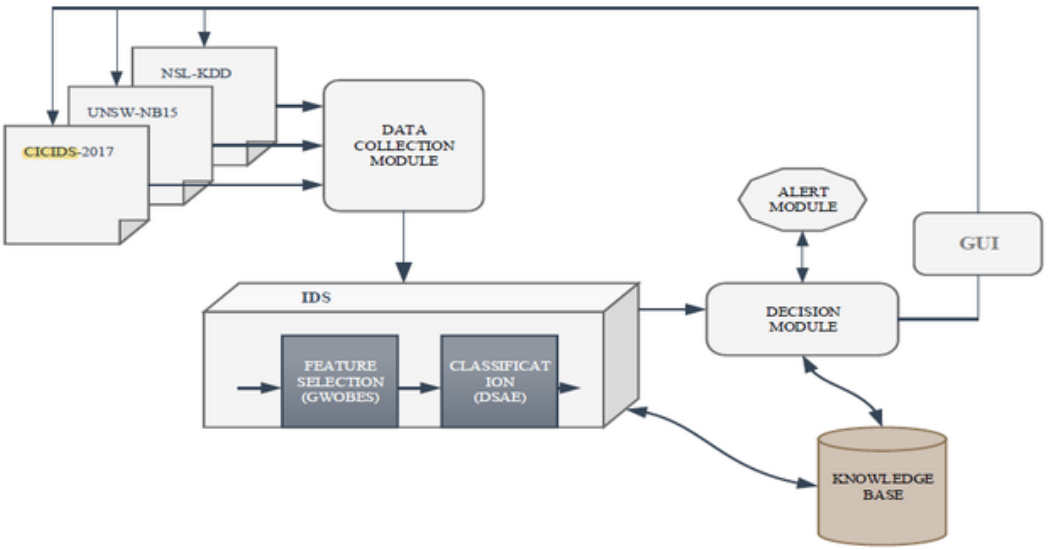
1. An anomaly-based NIDS, namely the GWO-BES-DSAE model, is designed to detect intrusions in the VCN environment with better accuracy and detection rate.
2. For better feature selection, a novel hybrid GWO-BES algorithm is designed and developed.
3. A deep sparse auto-encoder (DSAE) is utilized for the classification of the feasible selected features.

The remaining structure of the paper is described as: Section 2 discusses the related works. Section 3 describes the proposed methodology for intrusion detection. Section 4 presents the results, analysis, and evaluation of the GWO-BES-DSAE NIDS model for VCN. Finally, section 5 provides the overall conclusion.

2. RELATED WORK

The researchers have made efforts to develop efficient NIDS where they have used feature selection and classification techniques. Some of them are: Karimazad and Faraahi (Karimazad and Faraahi 2011) have proposed an anomaly-based intrusion detection model by utilizing the Radial Basis Function Network (RBF-N). The model shows 96% accuracy to detect DDoS on the UCLA dataset. They used a metaheuristic-based genetic algorithm to select features to be applied for detecting the anomalies, and the performance was evaluated on the NSL-KDD dataset. The adaptability and flexibility of the approach need to be enhanced. Dwivedi et al. (Dwivedi et al. 2020) have proposed a method for anomaly detection where combined Ensemble of Feature Selection (EFS) and one optimization algorithm named Adaptive Grasshopper Optimization Algorithm (AGOA) for identifying different types of attacks. EFS's role is to rank the attributes for further feature selection through AGOA to work as the input in predicting network traffic behavior. It is suggested that algorithm-specific parameters can affect the performance of the algorithm, hence the model accuracy also. Hota et al. (Ingre et al. 2014) provided a comparative survey on different hybrid methodologies for binary and multiclass predictions on the NSL-KDD Dataset. The hybrid IG-RF classifier achieved better performance where Information Gain (IG) was applied for feature selection (FS). Enache and Patriciu (Enache and Patriciu 2014) presented two approaches ABC-SVM and PSO-SVM, for intrusion detection on the NSL-KDD dataset in binary classification. Artificial Bee Colony (ABC) and particle swarm optimization (PSO) were used for feature selection. A support vector machine (SVM) was used for classification. PSO-SVM achieved better performance than the ABC-SVM. Eid et al. (Eid et al. 2011) used GA for feature selection and Navie Basian (NB) classifier for their proposed IDS classification. For the evaluation of their proposed IDS model has used the NSL-KDD Dataset with 10-fold cross-validation. They used Entropy Minimization Discretization (EMD) method for discretizing input in the feature selection process. Bamakan et al. (Ingre and Yadav 2015) proposed an IDS framework where they applied time-varying chaos particle swarm optimization (TVCPSO) for feature selection and SVM for classification. The results were evaluated on the NSL-KDD dataset and achieved 97.84% accuracy. Zhao et al. (Zhao and Zhu 2016) proposed using a neural network for a cloud environment where PSO was used to optimize the neural network. The results were obtained as per the expectations, i.e., PSO improved the system performance. Al-Zewairi et al. (Al-Zewairi, Almajali, and Awajan 2017) applied the DL approach for their proposed NIDS. They used a multilayer feed-forward ANN via backpropagation and stochastic gradient descent, termed a binomial classifier. The proposed model achieved an accuracy of 98.99% and FAR 0.56%. Sahil Garg and Shalini Batra (Garg and Batra 2018) proposed a hybrid based anomaly detection model, namely F-CBCT. The model showed its ability with high DR and low FPR in identifying the anomalies. They used DTC, CSO, and K-means in the primary phase, where DTC and CSO were used for feature optimization and K-means for clustering. After that DT and cuckoo search algorithms were combined in the detection phase, where the fuzzy approach was utilized for classification. Taj et al. (Taj et al. 2020) executed classifiers to be combined with the various IDS. They have considered J48, OneR, Naïve Bayes, Hoeffding Tree classifiers and

Figure 1: Proposed Structure of Network Intrusion Detection System



found that J48 achieved 99.45% accuracy on the KDD99 Dataset. Ghosh et al. (Ghosh et al. 2019) have proposed an IDS approach named CS-PSO. The CS-PSO model removes irrelevant features and consumes less memory to the considered dataset. In the CS-PSO model, PSO handled the exploitation phase, and CS dealt with the exploration phase. Pajouh et al. (Pajouh, Dastghaibyfard, and Hashemi 2017) presented the similar work with better classification results on NSL-KDD dataset. Gul et. al (Gul and Hussain 2011) have presented a NIDS for cloud environment to detect attacks on VMs by applying rule-matching and to handle network traffic efficiently.

Most of the NIDS developed for Cloud environments works on traditional techniques like Snort IDS and works only for limited capacities to face the new types of distributed and sophisticated attacks. This demands more efficient and effective NIDS that can handle the data of high dimension to work in a distributed environment such as a Virtual Cloud Network. Hence, in the proposed approach, the size of the dataset has been reduced through the hybrid GWO-BES algorithm, and the DL approach DSAE has been used for the classification as the DL-based classifiers are capable enough to learn the pattern from the large data samples. Hence, the proposed NIDS model is well suited for high dimensional data in cloud computing (CC) with enhanced performances such as accuracy, detection rate, false alarm rate, precision.

3. PROPOSED METHODOLOGY

The proposed methodology for NIDS in VCN is demonstrated in Figure 1. The major units of the anomaly-based NIDS as proposed methodology for VCN are data collection, feature selection, knowledgebase, alert, classification, Decision Manager.

In the data collection module, the data is collected from the VCN and stored in the form a dataset as per the requirement. The dataset of collected data is fed to the preprocessing module, where data cleaning and feature selection are carried out. The feature selection module selects relevant features from the dataset. For feature selection, one novel algorithm is designed and developed from the combination of GWO and BES named as hybrid GWO-BES. On the selected features, a DL-based approach Deep Sparse Auto-Encoder (DSAE) is applied that improves the accuracy in predicting the intrusion by minimizing the errors using Mean Squared Error (MSE) and KL-divergence during

the training phase. Achieving high accuracy and low false positives or low false alarms rate are the proposed system's main targets. A knowledgebase is proposed for good results in the proposed framework. The knowledge base holds the information about all the features and the sufficient rules for selecting the features and making an effective decision on the dataset. This module obtains the information about the features from the knowledge base, obtaining the hierarchy features, and then updates those features information into the knowledge base. The classification rules are also stored in this knowledge base to decide based on instances of features selected. This knowledge base also contains the possible effective rules used for particular attack detection or/and identification. It provides sufficient information to the classification for effective decision making. An alert is generated on the alert module to inform the network administrator that an attack has been detected. The results will be shown in the user interface.

Data Collection and preprocessing: The first step in any network intrusion detection approach is data collection and preprocessing. In the proposed approach, the data is also collected and preprocessed in an IoT network. This is a very important step as it improves the data quality and enhances the correctness of results, accuracy, and performance of the proposed approach. The preprocessing of data include cleansing, encoding, and normalization, as discussed briefly below:

Data Cleansing: Data cleansing or data scrubbing is an essential step of preprocessing as it removes redundancy and noise. Further, collected data may be incomplete, improper, or incorrect, or consist of null values. In data cleaning, all inconsistencies are removed. The Python environment is used to implement the module.

Encoding: Usually, in most of the datasets, some of the columns are categorical or string types containing multiple labels in the form of understandable words to humans, but the machine performs better in numeric data. There are many encodings such as one-hot, label, ordinal, binary, frequency, mean, probability ratio, hashing, backward difference, etc., to convert the string data into numeric. In this work, label encoding has been used that converts the categorical or string type variables values in numeric form using number sequencing **like the string values of a column or attribute good, better and best are sequenced as 0, 1 and 2.**

Normalization: The normalization of an attribute or feature is done to limit the numerical values of data in a range (usually 0 -1) without affecting range differences of actual values or without losing the information. For example, if the first column values range from 0 to 1 and the other column values range from 10,000 to 10,00,000, the variance in the two columns can lead to problems in modeling and analysis. Normalization helps to generate new values within the specified range without affecting the general distribution and results. The used equation to normalize values of various attributes present in them is given below as:

$$Z = \left(\frac{x - \min(x)}{max(x) - \min(x)} \right) \tag{1}$$

Where denotes feature value, and Indicates the smallest and largest value of feature .

3.1 FEATURE SELECTION MODULE

As we know, the dataset contains noise and many numbers of relevant and irrelevant features. The noise is removed in the preprocessing process. The relevant features are selected using the feature selection approach. Feature selection refers to a process where a feature subset from available features in the dataset is extracted or selected. Feature selection is needed because it removes irrelevant features, decreases the computational cost. The collected data from the collection module is fed to hybrid GWO-BES for a better feature selection. As nature-inspired methods are simple, speedy, and faster, convergence is needed to find a globally optimal solution than deterministic approaches. All

the relevant features from the datasets are selected, and the irrelevant ones are ignored. This process reduces the dataset's dimension, which leads to improving the prediction system's performance, i.e., the classification module.

3.1.1 Grey Wolf Optimization (GWO)

Mirjalili et al. (Mirjalili, Mirjalili, and Lewis 2014) proposed this population-based search algorithm. Alpha (α), beta (β), and delta (δ) represent hierarchal leadership from dominant to lower, which are used to calculate the position of the prey. To catch the prey, the wolves update their positions, and the best position is calculated by α, then the lower best positions are calculated by β and δ. The positions of the remaining wolves are updated based on the three best positions.

a) *Encircling prey:* The encircling of prey model is as follows:

$$E_p = \left| B * X_p(t) - X(t) \right| \tag{2}$$

$$X(t+1) = X_p(t) - C * E_p \tag{3}$$

Where B and C are coefficient vectors, X and X_p define the position vectors of grey wolf and prey, t indicates current iteration value, C and B are formulated using:

$$C = 2 * a * r_1 - a \tag{4}$$

$$B = 2 * r_2 \tag{5}$$

Where, r_1 and r_2 are the random numbers between 0 and 1, and a is decreasing linearly from 2 to 0 based on iterations.

b) *Hunting:* When α wolf plan for hunting the prey, β and δ wolves are supporting the α-wolf to participate in the hunting. The best positions of α, β, and δ with respect to prey as three best solutions and position updates for remaining wolves following to α, β, or δ wolves towards the prey are represented by the equations (6), (7), and (8). Equation (9) is the average position updates for the next iteration based on the current position update.

$$E_{p\alpha} = \left| B_1 * X_{p\alpha}(t) - X(t) \right|;\ X_1 = X_{p\alpha}(t) - C_1 * E_{p\alpha} \tag{6}$$

$$E_{p\beta} = \left| B_2 * X_{p\beta}(t) - X(t) \right|;\ X_2 = X_{p\beta}(t) - C_2 * E_{p\beta} \tag{7}$$

$$E_{p\delta} = \left| B_3 * X_{p\delta}(t) - X(t) \right|;\ X_3 = X_{p\delta}(t) - C_3 * E_{p\delta} \tag{8}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{9}$$

c) *Attacking prey:* As the prey stops moving, the grey wolf attack to finish the hunting. The attack process is based on the 'a', which decreases its value from two to zero depending on the iterations.

3.1.2 Bald Eagle search (BES) Algorithm

Due to the huge size of bald eagles, they are top in the food chain (Alsattar, Zaidan, and Zaidan 2019). The main capability of bald is to catch fish in water from a large distance. Bald eagles, when searching for foodstuff, has three main stages. In the initial step, the eagle selects a search area to move towards it. In the second phase, the eagle starts searching in the designated space, followed by the third stage, where the eagle moves towards the prey. The main step of the BES algorithm is the select stage, search stage, and swooping stage.

i) *Select phase*: In this phase, bald eagles pick a range where more prey are available. The bald eagle selects an area somewhat different from the previously selected search area based on the previous stage data. The mathematical representation is:

$$P_{n,i} = P_{best} + \alpha * q(P_{avg} - P_i) \tag{10}$$

Where that controls changes in position takes a range from 1.5 to 2. q is a random number ranging from 0 and 1. It represents the search area selected by bald eagles currently. Information used by the eagles in the previous points is denoted by P_{avg} .

ii) *Search phase*: In this step, searching for prey by the eagles in the chosen area is done in a spiral manner. A mathematical explanation for the best position for swooping is given below:

$$P_{i,n} = P_i + y(i)(P_i - P_{i+1}) + x(i)(P_i - P_{avg}) \tag{11}$$

$$x(i) = \frac{xq(i)}{\max(\left|xq\right|)}\ y(i) = \frac{yq(i)}{\max(\left|yq\right|)} \tag{12}$$

$$xq(i) = q(i) * \mathrm{Sin}(\theta(i));\ yq(i) = q(i) * \mathrm{Cos}(\theta(i)) \tag{13}$$

$$\theta(i) = a * \pi * rand\,;\; q(i) = \theta(i) + R * rand \tag{14}$$

Where '*a*' finds the corners among point search, it takes the interval [5, 10]. Value of '*R*'ϵ [0.5, 2] and is used for finding the number of search phases. The change in spiral shape is achieved by changing '*a*' and '*R*'.

iii) *Swooping phase*: The best point to hunt is identified, and the eagle changes from the second phase's position. The behavior is defined by,

$$P_{i,n} = rand * P_{best} + x_1(i)(P_i - c_1 * P_{avg}) + y_1(i)(P_i - c_2 * P_{best}) \tag{15}$$

Where c_1 *and* c_2 indicate the eagle's movement's power to the best position and range a value between 1 and 2.

3.1.3 Hybrid GWO – BES

GWO algorithm has various advantages like working with a smaller number of parameters, the ability to achieve solid global optimization, and easy implementation. However, for some cases like multi-objective problems, working with more variables, it gets trapped into local minima due to deficiency of proper position updates of alpha, beta, and delta wolves. This probability of getting trapped into local solutions is more when the GWO algorithm tries to solve a problem with numerous variables and many local solutions at a fast convergence speed and accelerates exploitation (Faris et al. 2018). This chapter removes grey wolf optimization (GWO) such as low accuracy and poor local searching ability by combining the BES in the position update equation. This hybridization is utilized for selecting the best feasible features. At the initial stage, all feature values are given as input to the hybrid GWO-BES to select only the relevant features and ignore the irrelevant features by satisfying the objective function's criteria. In the feature selection, a hybrid GWO-BES algorithm has been designed to overcome the drawbacks of GWO leads to improve classification performance. First, all features of the dataset are initialized. Then hybrid GWO-BES is used for the selection of the feasible features. Position updates for the hybrid GWO-BES algorithm are as following:

$$F_{best} = r_{and} * X_1 + x(i)(X(t) - c_1 X_2) + y(i)(X(t) - c_2 X_3) \tag{16}$$

The pseudo-code of hybrid GWO-BES for FS is given below in Figure 2.

Fitness function $\mathrm{F}\left(x\right)$ in hybrid GWO_BES is the squared sum of each agent (attribute) is:

$$\mathrm{F}\left(x\right) = \sum_{i=0}^{n} x_i^2 \tag{17}$$

The selected features are the input for the classification module where DSAE has been used.

3.2 Deep Sparse Auto-encoder (DSAE) for Classification

In the sparse encoder, neurons become active only for the meaningful pattern otherwise inactive. The Kullback-Leibler (*KL)* divergence in the loss function regularizes the learning and reduces the overfitting situation. This KL divergence depending on the hidden unit output, which again depending on the weight matrix. An auto-encoder has three layers: the input layer, one or more hidden layers,

Figure 2. Proposed Algorithm for feature selection as Hybrid GWO-BES

Algorithm 1 Hybrid GWO-BES for feature selection

1: **Input:** Dataset, maximum iteration, fitness function
2: **Output:** best selected feature indexes
3: Initialize Positions from Dataset, t, position of α, β, and δ.
4: Initialize a=2, B, C for each leader wolves as B_1, C_1, B_2, C_2, and B_3, C_3 using Eq. (4) and (5).
5: Calculate fitness for each attribute or search agent using fitness function.
6: Calculate the best positions of α, β, and δ with respect to prey and position updates for remaining wolves following to α, β, or δ wolves towards the prey using Eq. (6), (7), and (8).
7: $X_{p\alpha}$= Position of grey wolf with first maximum fitness then X_1
8: $X_{p\beta}$= Position of grey wolf with second maximum fitness then X_2
9: $X_{p\delta}$= Position of grey wolf with third maximum fitness then X_3
10: **while** (t<maximum iteration) **do**
11: **for** (each attribute or search agent) **do**
12: Update the position of current search agent using Eq. (16)
13: **end for**
14: Calculate a $=(2 - t * \frac{2}{maximum\ iteration})$
15: Update the parameters B and C for each leader wolves as B_1, C_1, B_2, C_2, and B_3, C_3 using Eq. (4) and (5).
16: Calculate fitness for each attribute or search agent using fitness function.
17: Update $X_{p\alpha}$, $X_{p\beta}$, and $X_{p\delta}$ as well as X_1 X_2 and X_3
18: t=t+1
19: **end while**
20: Return the selected features as the oprimal feature subset

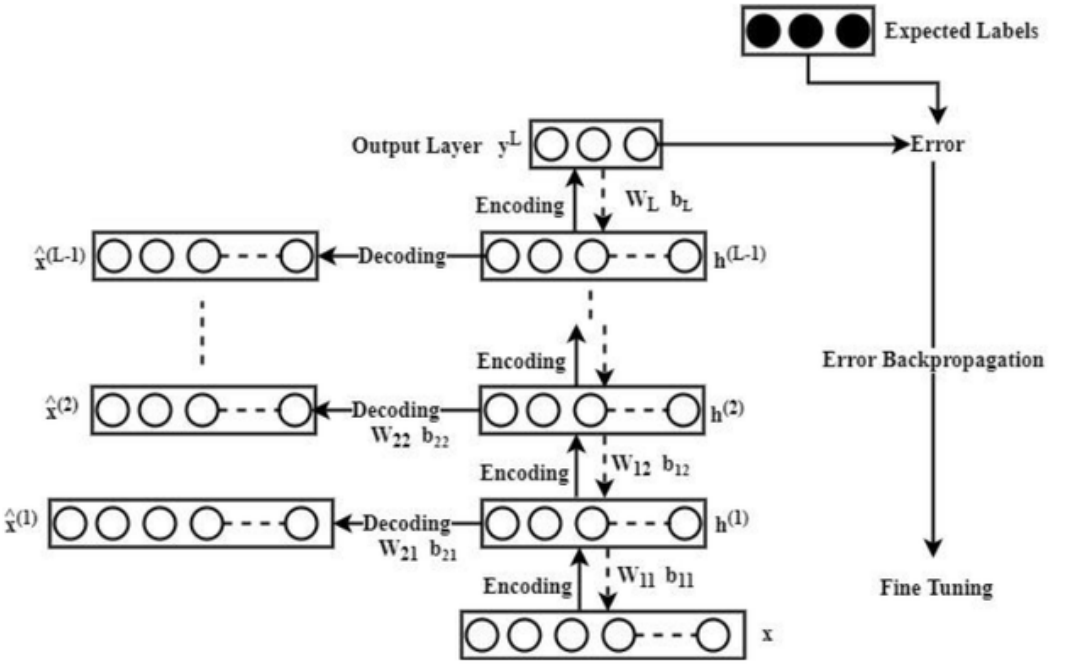
and one output layer, where the next layer represents its previous layer. The sparse auto-encoders hidden layer contains more neurons than the input layer. The reconstruction error of sparse auto-encoder is given as:

$$L_{sparse}(x,\hat{x}) = \frac{1}{2}\sum_{i=1}^{N} || \, x_i - \hat{x}_i \, ||^2 + \beta \sum_{j=1}^{m} KL(\rho \, || \, \hat{\rho}) \tag{17}$$

Here $\hat{x}_i$ is the indication of output, which is calculated through a sigmoid function (σ). σ depends on the bias between the hidden layer and output layer and the weight matrix. Beta (β) indicates the sparse penalty term's weight, indicating the sparsity near zero. Figure 3 gives the structure of the DSAE.

When sparse auto-encoders are stacked in a Deep Neural Network, it is known as deep sparse auto-encoder (DSAE) (Dhanabal and Shantharajah 2015), and a softmax function is added at the output layer for classification. Every hidden layer consists of an encoder and decoder pair. In the pre-training phase, the DSAE is trained. All stacked sparse auto-encoders are trained from bottom to top till the previous output layer. An error backpropagation is used to fine-tune weights and biases of the whole network. The objective function of DSAE with L number of hidden layers is as follows:

Figure 3. General Structure of the DSAE



$$J(W_L, W_{1,K}, B_{1,K}) = \arg \min_{w_l, w_{1,k}, b_{1,k}} \frac{1}{2N} \sum_{i=1}^{N} || \, y_i - g_l(f_l(h_i^{L-1})) \, ||_2^2 \tag{18}$$

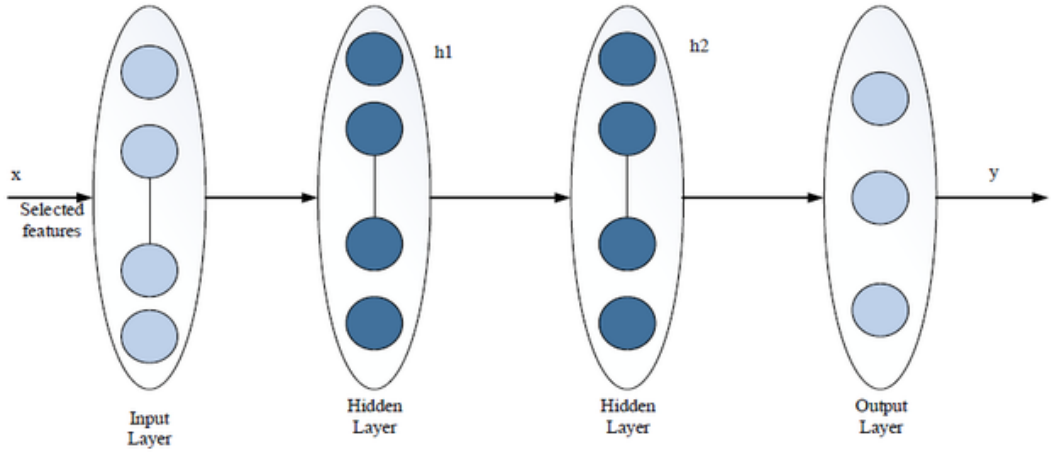
Where h_i^{L-1} is the activation value at $\left(L-1\right)^{th}$ hidden layer, $w_{1,k}$ and $b_{1,k}$ are the weight and bias of k^{th} layer, y_i is the label as output. W_L is the weight of the last layer. The whole network's parameters, i.e., weight and bias of each layer updated regularly until it reaches the objective function's constant or max-epoch.

The selected features are given to the DSAE input layer, where the data are reconstructed in an unsupervised manner. *The number of neurons at the input layer is equal to the number of selected features.* In this paper, two hidden layers are considered for the DSAE in a study in (Jia et al. 2019) to achieve better classification accuracy in the IDS, as indicated by Figure 4.

4. RESULTS AND ANALYSIS

In this section, the performance of the proposed model is determined and analyzed. The proposed approach is implemented using a python environment. Anaconda Navigator has been used to create the Python programming language (Version 3.7.3). The developed models' performance assessment has been carried out on an HP machine operated on Intel Core i7-5500U CPU @ 2.40 GHz 2401 Mhz, 2 Cores 4 Logical processors with 12 GB RAM running Microsoft Windows10 Professional. This section discusses data collection, performance metrics, analysis, and discussion of results obtained after rigorous implementation and analysis on varied three datasets.

Figure 4. Considered Structure of DSAE



4.1 Dataset Description

The proposed GWO-BES-DSAE IDS model for VCN has been evaluated on ***NSL-KDD, UNSW-NB15, and CICIDS-2017*** datasets. The CICIDS-2017 dataset (Sharafaldin, Lashkari, and Ghorbani 2018) was generated with the real traces of benign (normal) and most common attacks from the network traffic: 2,830,108 records are available with 84 features. The dataset is available in eight CSV files. Namely, there are eight CSV files, namely Friday-WorkingHours-Afternoon-DDos.csv, Friday-WorkingHours-Morning.csv, Friday-WorkingHours-Afternoon-PortScan.csv, Monday-Working Hours.csv, Thursday-WorkingHours-Morning-WebAttacks.csv, Thursday-WorkingHours- Afternoon-Infilteration.csv, Tuesday-WorkingHours.csv, and WednesdayworkingHours.csv. The used file is Thursday-WorkingHours-Morning-WebAttacks.csv, where 225,745 records are there with over 80 features. Three types of attacks are in the considered file: web attack brute force, web attack XSS, and Web attack SQL injection. The UNSW-NB15 Dataset (Moustafa and Slay 2015) has 2,540,044 records with 49 features and nine types of attacks: Reconnaissance, Shellcode, Worm, Genetic, and Exploit, DoS, Backdoor, Analysis, and Fuzzers. NSL-KDD Dataset (Dhanabal and Shantharajah 2015) is considered a developed version of network data of the KDD 99 dataset. NDL-KDD contains 41 features with four types of attacks: DoS, Probe, R2L, and U2R. The UNSW-NB15 dataset has ten categories: normal, Reconnaissance, Shellcode, Worm, Genetic, Exploit, DoS, Backdoor, Analysis, and Fuzzers.

4.2 Feature Selection

The proposed novel hybrid GWO-BES algorithm is used for selecting the optimized subsets of features from the above datasets. The extracted features from different datasets using the hybrid GWO-BES algorithm written in Python are summarized in Table 1.

4.3 Performance Metrics

The considered metrics for evaluating the performance of the proposed NIDS model for VCN areas following:

a) ***Recall or Attack detection rate (ADR):*** defines the capability of detecting various attacks in the NIDS.

Table 1. The extracted features from considered datasets on applying the hybrid GWO-BES

Dataset	Number of selected features	Extracted Features as per their ranking
CICIDS-2017	30	‘ Bwd PSH Flags’, ‘ Fwd URG Flags’, ‘ Bwd URG Flags’, ‘ CWE Flag Count’, ‘Fwd Avg Bytes/Bulk’, ‘ Fwd Avg Packets/Bulk’, ‘ Fwd Avg Bulk Rate’, ‘ Bwd Avg Bytes/Bulk’, ‘ Bwd Avg Packets/Bulk’, ‘Bwd Avg Bulk Rate’, ‘ RST Flag Count’, ‘ ECE Flag Count’, ‘FIN Flag Count’, ‘Fwd PSH Flags’, ‘ SYN Flag Count’, ‘ URG Flag Count’, ‘ PSH Flag Count’, ‘ ACK Flag Count’, ‘ Down/Up Ratio’, ‘Bwd Packet Length Mean’, ‘Avg Bwd Segment Size’, ‘Packet Length Std’, ‘Max Packet Length’, ‘Destination Port’, ‘Packet Length Mean’, ‘Bwd Packet Length Std’, ‘FIN Flag Count’, ‘Average Packet Size’, ‘Fwd IAT Mean’, ‘Flow IAT Std’
UNSW-NB15	31	‘ackdat’, ‘is_ftp_login’, ‘ct_ftp_cmd’, ‘synack’, ‘is_sm_ips_ports’, ‘tcprtt’, ‘trans_depth’, ‘ct_flw_http_mthd’, ‘ct_state_ttl’, ‘ct_dst_sport_ltm’, ‘dur’, ‘ct_src_dport_ltm’, ‘ct_dst_ltm’, ‘ct_src_ltm’, ‘ct_dst_src_ltm’, ‘ct_srv_ dst’, ‘ct_srv_src’, ‘proto’, ‘dloss’, ‘sloss’, ‘dpkts’, ‘spkts’, ‘dttl’, ‘dwin’, ‘sttl’, ‘swin’, ‘smean’, ‘dmean’, ‘dinpkt’, ‘djit’, ‘sinpkt’.
NSL-KDD	25	‘num_outbound_cmds’, ‘is_host_login’, ‘land’, ‘urgent’, ‘num_shells’, ‘root_shell’, ‘su_attempted’, ‘num_failed_logins’, ‘is_guest_login’, ‘num_access_files’, ‘dst_host_srv_diff_host_rate’, ‘diff_srv_rate’, ‘dst_host_diff_srv_rate’, ‘wrong_fragment’, ‘srv_diff_host_rate’, ‘dst_host_rerror_rate’, ‘dst_host_srv_rerror_rate’, ‘rerror_rate’, ‘dst_host_ same_src_port_rate’, ‘srv_rerror_rate’, ‘num_file_creations’, ‘dst_host_ srv_serror_rate’, ‘dst_host_serror_rate’, ‘srv_serror_rate’, ‘serror_rate’

$$ADR = \frac{TP}{TP + FN}$$

TP is the number of intrusive networks categorized correctly as intrusions, and FN is the number of intrusive networks categorized incorrectly as normal.

b) ***F_1 score*** defines the accuracy based on precision rate and recall rate.

$$F_1\ Score = \frac{2TP}{2TP + FP + FN}$$

Where FP is the number of normal networks classified incorrectly as an attack.

c) ***False Alarm Rate (FAR):*** The value of FAR should be as low as possible so that it will reduce the FP alarms; otherwise, the system admin will get confused.

$$FAR = \frac{FP}{FP + TN}$$

Where TN is the volume of normal networks classified correctly as normal.

d) ***Precision:*** indicate the ratio of positively predicted values that are actually being positive. Higher PR indicates low FPR.

$$Precision = \frac{TP}{TP + FP}$$

e) ***False Negative Rate (FNR):*** defined as ratio of total misidentified attack to the total identified attacks. A greater value of FNR indicates a greater rate of negative alarms.

$$FNR = \frac{FN}{FN + TP}$$

f) ***Accuracy:*** Ratio of the number of correctly classified samples to the total number of samples for a given test data set fed to classifier during the testing phase:

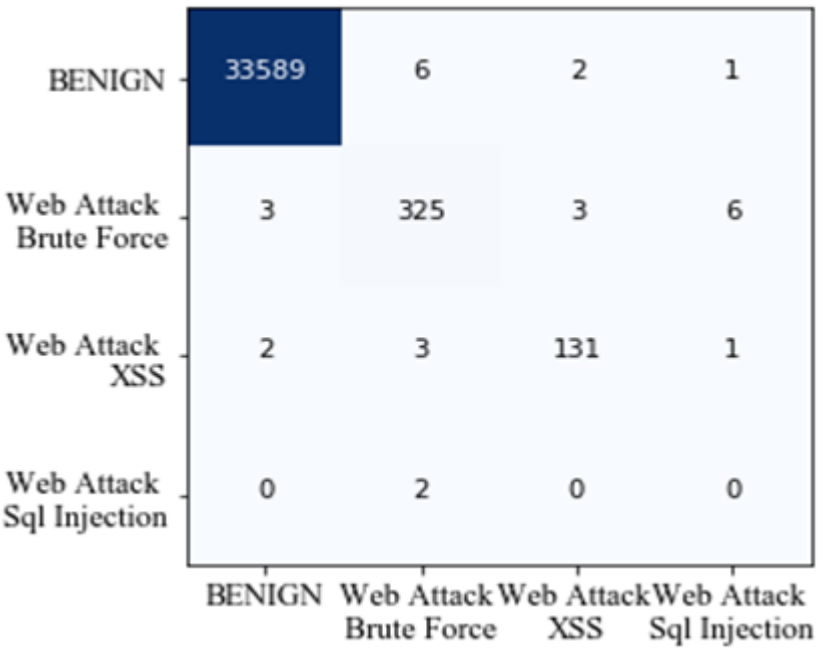
$$Accuracy = \frac{TP + TN}{TN + FP + FN + TP}$$

4.4 Analysis and Discussion of Results

For the evaluation of performance for both binary classification and multiclass classification on the considered three network datasets - CICIDS-2017, UNSW-NB15 and NSL-KDD are carried out in two scenarios – (i) on using the train test split validation method and (ii) On publicly available independent testing sets of the datasets. The intense simulation and regression analysis for both scenarios are summarized as follows:

1) For the datasets CICIDS-2017, UNSW-NB-15 and NSL-KDD, the considered file names are Thursday-WorkingHours-Morning-WebAttacks.csv, KDDTrain+.csv, and UNSW_NB15_ training-set.csv, respectively. The considered validation method is the train test split, where the test set size is 20% data, i.e., the proposed model has been trained on 80% and tested in 20%. After applying the GWO-BES-DSAE NIDS model on the test sets, generated confusion matrixes are depicted in Figure 5, Figure 6, and Figure 7, respectively. Confusion matrix (CM) provides the performance evaluation based on the sum of attacks correctly categorized and the number of attacks incorrectly identified. The obtained performance metrics - accuracy, precision, F1 Score, Recall, FAR, and FAR values for binary classification and multiclass classification on the CICIDS-2017, UNSW-NB-15, NSL-KDD datasets are provided in Table 2, Table 3. The obtained performance metrics accuracy, precision, recall, and F1 score values of the proposed GWO-BES-DSAE NIDS model are compared with others' works (He et al. 2019) for binary as well as multiclass classification that are presented in Table 4 and Table 5. After that, the average accuracy comparisons are presented in Figure 8. Training time, testing time, and prediction time of the proposed GWO-BES-DSAE NIDS model are depicted in Table 6.
2) The proposed model has also been validated on an independent testing set available for NSL-KDD and the UNSW-NB Datasets. The obtained CMs are demonstrated in Figure 9 and Figure 10. The obtained performance metrics - accuracy, precision, F1 Score, Recall, FAR, and FAR values for binary classification and multiclass classification on the NSL-KDD and UNSW-NB15 datasets are presented in Table 7. The obtained performance metrics of the proposed NIDS model

Figure 5. CM for testing part of Thursday-WorkingHours-Morning-WebAttacks.csv of CICIDS dataset for GWO-BES-DSAE



are also compared with the work of others and presented in Table 8. The comparisons of the processes involved in the proposed NIDS model and the processes involved in the other works are carried out and demonstrated in Table 9.

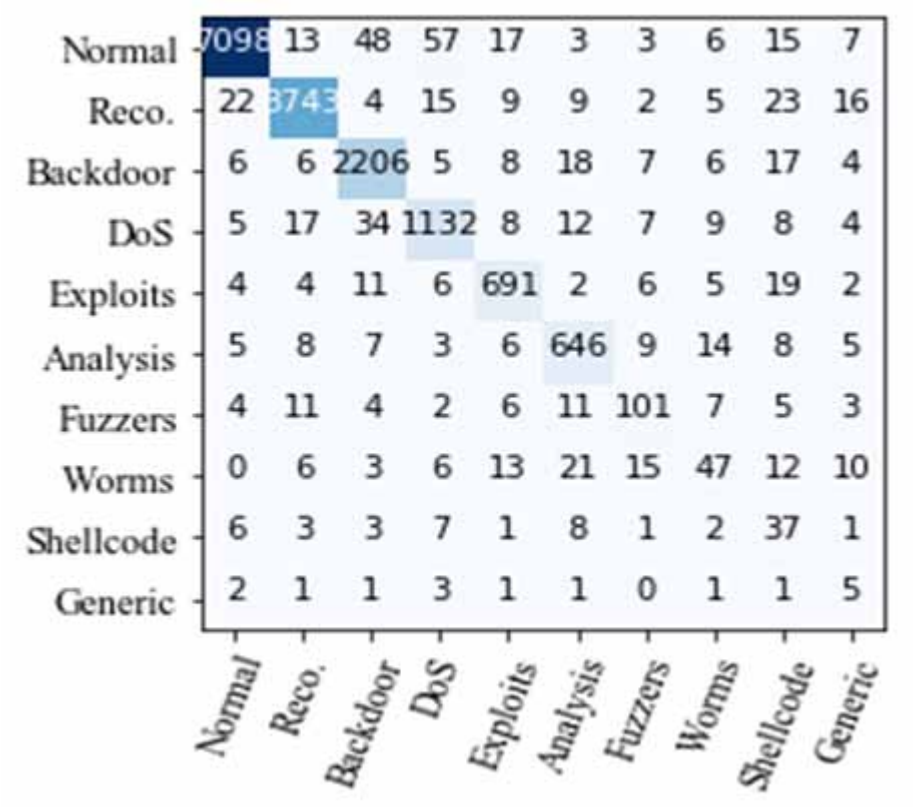
Based on the accuracy recall, F1 score, and precision, the proposed NIDS model is compared with some existing methods - SVM, DNN, MS-DHPN (He et al. 2019). Table 4 depicts a binary classification of the considered three datasets -NSL-KDD, UNSW-NB-15 CICICS-2017 datasets.

Table 5 depicts multiclass classification performance comparisons of the proposed GWO-BES-DSAE NIDS model with some existing methods - SVM, DNN, and MS-DHPN, as reported in (He et al. 2019). In multiclassification, the model is trained for many categories classes instead of two classes, which affects the performance of the system. In multi-classification, the NSL-KDD datasets have five categories: normal, DoS, Probe, R2L, and U2R. The UNSW-NB15 datasets have ten classes: Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shell-code, and Worm. The CICDIS-2017 dataset has five classes: web attack brute force, web attack XSS, Web attack SQL injection, and normal.

The average performance of the proposed NIDS model is better than other models presented in (He et al. 2019). The average accuracy is also compared and presented in Figure 8.

In the train-test split validation method, the training time, prediction time, and testing time for both binary and multiclass classification for all the three datasets are shown in Table 6. The training time required for the UNSW-NB 15 dataset is low compared to the other two datasets. Prediction time describes the time consumed by the system to predict the normal and attack categories for a particular instance, and the prediction time is high for the CICIDS 2017 dataset. **Training time time is considered from the selected feature subset to get trained model. Testing time is considered from selected feature subset to prediction result of the testing dataset in the form of accuracy**

Figure 6. CM for testing part of UNSW_NB15_training-set.csv of UNSWNB-15 Dataset forGWO-BES-DSAE



and classification report (precision, recall, f1-score). Prediction time is considered from selected feature subset to prediction of a sample or record of the network flow.

The proposed model has also been validated on an independent testing set available in the UNSW-NB Dataset and NSL-KDD Dataset, i.e., the proposed model is trained on the independent training set and tested on the independent testing set available in the UNSW-NB15 Dataset and NSL-KDD Dataset. Table 7 presents the performance values for binary classification and multiclass classification, respectively.

Table 8 gives the performance comparison of detection rate (DR) and false alarm rate (FAR) for NSL-KDD and UNSW-NB15 datasets for various existing other methods. For the comparison of DR and FAR with the proposed system, HIDCC and DT-EnSVM are the existing methods for the NSL-KDD Dataset. For the UNSW-NB15 Dataset, HLDNS and RB-IDS are the existing methods considered. Our proposed system achieved a better result compared with the current methods. For both datasets, the proposed GWO-BES-DSAE NIDS model is reporting higher accuracy than other existing approaches, as depicted in Table 8. As depicted in Table 8 the FAR of the proposed NIDS model is higher than DT-EnSVM (Gu et al. 2019) but lower than HIDCC (Hatef et al. 2018) for the NSL-KDD dataset. Table 8 depicts that the FAR of the proposed NIDS model on the UNSW-NB15 dataset is lower than RB-IDS (Kumar et al. 2019) and HLDNS (Patil, Dudeja, and Modi 2019).

A comparison of the proposed approach with current studies or methods is presented in Table 9.

Figure 7. CM for testing part of KDDTrain+.csv of NSLKDD dataset for GWO-BES-DSAE

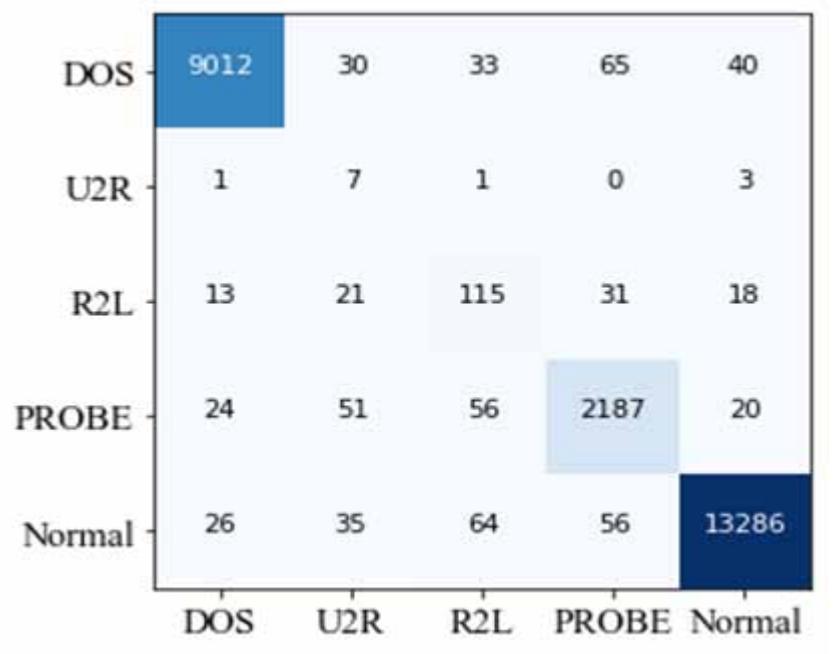


Table 2. Performance values for binary classification of the proposed system

Datasets	Accuracy	Precision	F_1 Score	Recall	FNR	FAR
CICIDS-2017	98.12	98.25	98.12	98	2	0.186
UNSW-NB15	99.82	99.59	99.74	99.18	0.089	0.184
NSL-KDD	99.82	99.59	99.74	99.91	0.089	0.184

Table 3. Performance values for multiclass classification of the proposed system

Datasets	Accuracy	Precision	F_1 Score	Recall	FNR	FAR
CICIDS-2017	96.06	96	96.04	96.13	3.875	0.394
UNSW-NB15	99.68	99.32	99.57	99.81	0.185	0.313
NSL-KDD	99.68	99.33	99.57	99.81	0.185	0.313

4.5 Discussion

From the above comparisons, it has been observed that the overall performance of the proposed GWO-BES-DSAE NIDS model is better than other recent approaches. In this approach, the hybrid version of GWO with BES is introduced for FS. It provides a better selection of relevant feasible features, and the irrelevant features are ignored. Usually, most swarm intelligent algorithms lack a

Table 4. Test values for binary classification of various existing methods

Datasets	Methods	Accuracy	Precision	Recall	F_1 Score
NSL-KDD	SVM	83.7	76.9	99.3	86.7
	DNN	80.1	96.6	67.4	79.4
	MS-DHPN	85.9	94.9	79.9	86.8
	GWO-BES-DSAE	99.82	99.59	99.91	99.74
UNSW-NB15	SVM	65.3	99.8	49.2	65.9
	DNN	78.4	94.4	72.5	82.0
	MS-DHPN	96.8	95.1	99.3	97.1
	GWO-BES-DSAE	99.75	99.59	99.10	99.74
CICIDS-2017	SVM	79.9	99.2	32.8	49.3
	DNN	93.1	82.7	97.4	89.4
	MS-DHPN	99.9	99.9	99.8	99.9
	GWO-BES-DSAE	98.12	98.25	98.0	98.12

Table 5. Test values for multiclass classification of various existing methods

Datasets	Methods	Accuracy	Precision	Recall	F_1 Score
NSL-KDD	SVM	70.2	68.9	70.2	65.6
	DNN	78.5	81.0	78.5	76.5
	MR-DHPN	80.2	80.6	80.2	80.4
	GWO-BES-DSAE	99.68	99.33	99.81	99.57
UNSW-NB15	SVM	58.1	58.6	58.1	49.6
	DNN	64.5	61.4	64.5	58.6
	MR-DHPN	86.2	84.4	86.2	85.3
	GWO-BES-DSAE	99.67	99.32	99.81	99.57
CICIDS-2017	SVM	79.9	75.7	79.9	72.3
	DNN	94.8	96.5	94.8	95.3
	MR-DHPN	98.6	98.6	99.6	98.6
	GWO-BES-DSAE	96.06	96.00	96.13	96.04

leader to control. In contrast, the combination of grey wolves and bald eagles maintains the social hierarchy and leadership nature make the GWO with BES a better choice. Sparsity is used in deep learning approach DSAE for reducing the links within the network and, thus, generally increases the generalization performance of the deep learning technique. The KL-divergence added in the loss function as regularization, which separates the overlapping spectrum of different attacks, making the learning of correct patterns easier during the model's training phase, which is also known as the sparsity term. It overcome the overfitting during individual auto-encoder; it helps the auto-encoder learn more generalized features during the pre-training phase. Due to better feature selection and classification, the computational complexity becomes reduced, and efficiency is improved. Because of this, the proposed model produced a high accuracy in the prediction.

Figure 8. Average Performance Comparison of all Methods in Binary and Multi-class Classification.

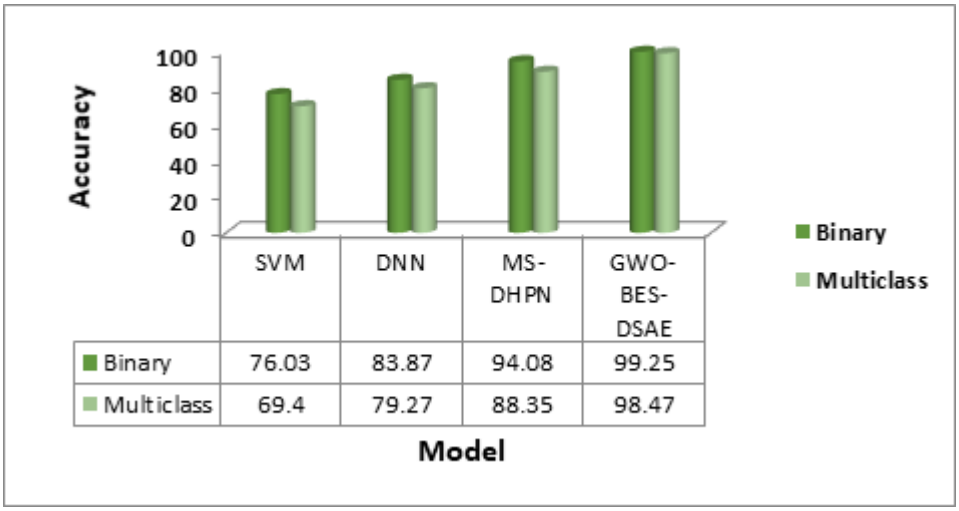


Table 6. Training, testing, and prediction time results

	NSL-KDD		UNSW-NB15		CICIDS 2017	
	Binary	**Multiclass**	**Binary**	**Multiclass**	**Binary**	**Multiclass**
Training time (s)	4.785	4.76	3.93	3.94	6.39	6.45
Testing time(s)	2.53	2.56	1.11	1.175	4.6	4.675
Prediction time (s)	0.104	0.049	0.406	0.058	0.169	0.159

Figure 9. CM for independent testing part of NSLKDD dataset for GWO-BES-DSAE

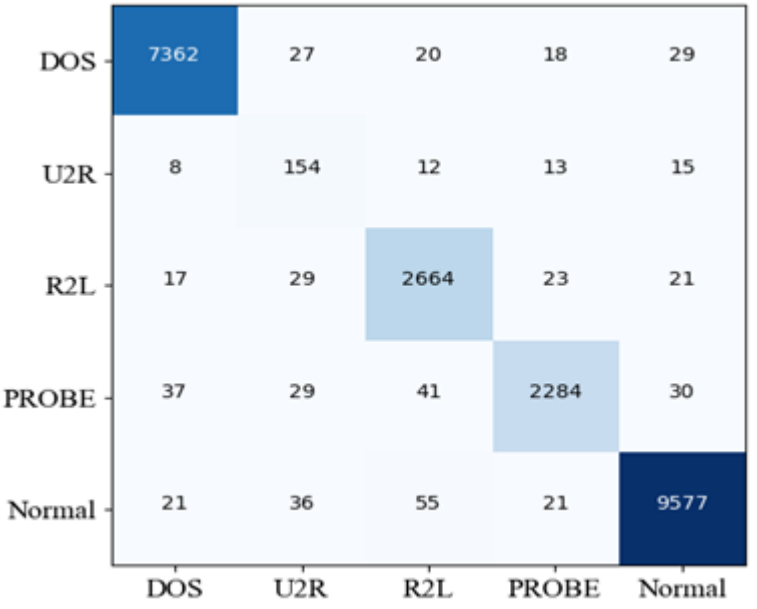


Figure 10. CM for independent testing part of UNSW-NB15 Dataset for GWO-BES-DSAE

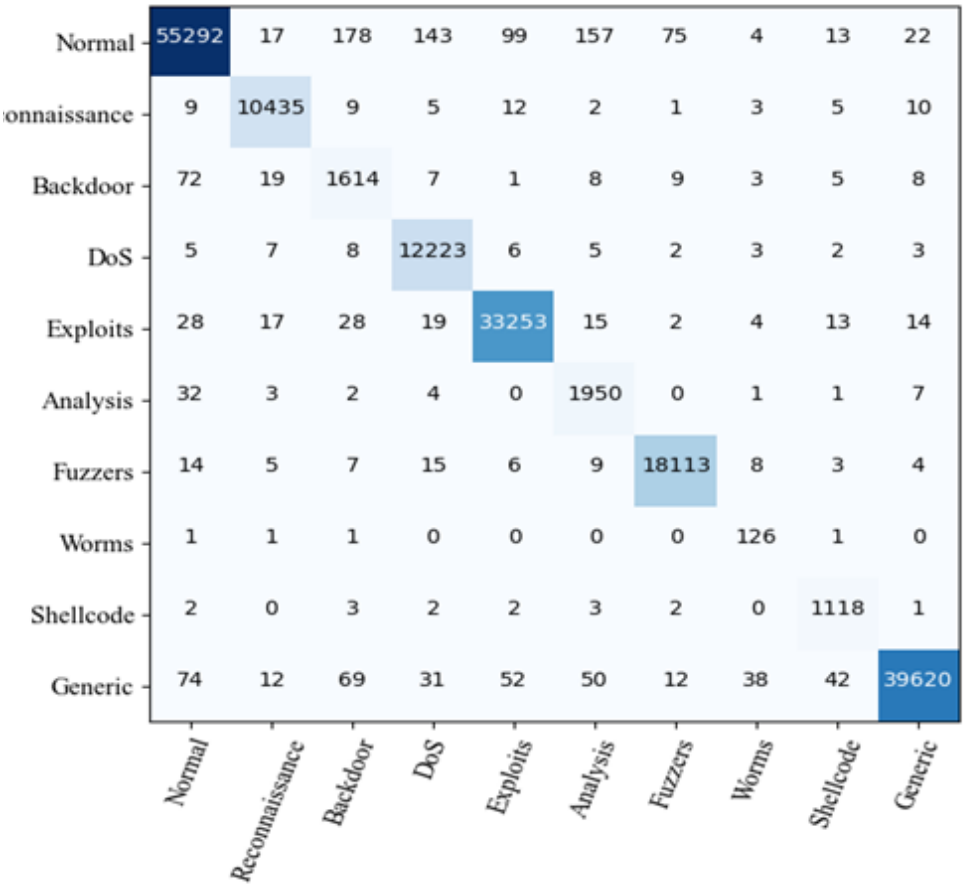


Table 7. Performance values for binary class and multi class classification of proposed NIDS model

Binary class classification performance report						
Datasets	**Accuracy**	**Precision**	**F_1 Score**	**Recall**	**FNR**	**FAR**
NSL-KDD	99.47	99.35	99.47	99.6	0.4	0.523
UNSW-NB15	99.50	99.53	99.50	99.46	0.53	0.50
Multiclass classification performance report						
NSL-KDD	99.42	99.35	99.42	99.51	0.50	0.575
UNSW-NB15	99.43	99.40	99.43	99.46	0.53	0.566

Table 8. Performance values for DR and FAR for NSL-KDD and UNSW-NB15 Dataset

Dataset	METHODS	DR	FAR
NSL-KDD	PROPOSED	99.44	0.55
	HIDCC (Hatef et al. 2018)	99.38	0.7
	DT-EnSVM (Gu et al. 2019)	99.07	0.38
UNSW-NB15	PROPOSED	99.47	0.53
	HLDNS (Patil, Dudeja, and Modi 2019)	99.09	0.63
	RB-IDS (Kumar et al. 2019)	90.32	2.01

Table 9. Comparative analysis of the proposed approach with current studies or methods

SI.	Approach	Proposed Approach	Tama et al. (Tama, Comuzzi, and Rhee 2019)	Alamiedy et al. (Alamiedy et al. 2019)	Dwivedi et al. (Dwivedi et al. 2019)	Negandhi et al. (Negandhi, Trivedi, and Mangrulkar 2019)
1	Feature Selection Method	GWO-BES	Particle swarm optimization (PSO), ant colony optimization (ACO), Genetic Algorithm (GA)	Grey wolf optimization (GWO)	Adaptive grasshopper optimization algorithm (AGOA)	Gini importance
2	Classification Method	DSAE	Combined Rotation forest and Bagging	SVM	SVM	Random forest
3	Validation Method	Train-Test Split	10-fold cross-validation	Train-Test Split	10-fold cross-validation	Train-Test Split
4	Used dataset (s)	NSL-KDD, UNSW-NB-15, and CICIDS-2017	NSL-KDD and UNSW-NB-15	NSL-KDD	ISCX 2012	NSL–KDD
5	Work Proposed For	Anomaly Detection	Anomaly Detection	Anomaly Detection	Anomaly Detection	Anomaly Detection
6	Number of Attributes	NSL-KDD -25 UNSW-NB-15: 31 & CICIDS-2017: 30	NSL-KDD -37 and UNSW-NB-15: 19	4	13	25
6	Classification Accuracy (%)	NSL-KDD -99.66 UNSW-NB-15: 99.63 and CICIDS-2017: 98.12	NSL-KDD -99.557 and UNSW-NB-15: 97.055	87.59	98.96	99.80

5. CONCLUSION

In this paper, an innovative intrusion detection model named GWO-BES-DSAE IDS was proposed to secure the virtual cloud network (VCN) by uncovering the known and the unknown threats. Hence, preserving confidentiality, integrity, and availably of information, and maintaining the performance of cloud assets and the quality and offered services. The proposed system utilizes feature selection and classification for intrusion detection. For feature selection, grey wolf optimization GWO is a hybrid with a bald eagle search (BES) algorithm. It provides a better selection of relevant feasible features, and the irrelevant features are ignored. While, for the classification of intrusions in the network, a deep sparse auto-encoder (DSAE) was applied. The anomaly-based NIDS developed comprises six modules: data collection, feature selection, knowledgebase, alert, classification, and Decision Manager. The performance was assessed on the popular networking datasets, namely NSLKDD, UNSW-NB15, CICIDS- 2017. Hence, the gained outcomes demonstrate that the suggested GWO-BES-DRAE NIDS model yields better performance than other recent approaches. The proposed NIDS model for VCN has achieved an average accuracy of 99.20% in binary classification and 98.42% in multiclass classification on the considered datasets.

REFERENCES

Al-Zewairi, M., Almajali, S., & Awajan, A. (2017). Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System. *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, 167–72. doi:10.1109/ICTCS.2017.29

Alamiedy, , & Anbar, , Alqattan, & Alzubi. (2019). Anomaly-Based Intrusion Detection System Using Multi-Objective Grey Wolf Optimisation Algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1–22.

AlKadi, O., Moustafa, N., Turnbull, B., & Choo, K.-K. R. (2019). Mixture Localization-Based Outliers Models for Securing Data Migration in Cloud Centers. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 114607–114618. doi:10.1109/ACCESS.2019.2935142

Alomari, O., & Othman, Z. A. (2012). Bees Algorithm for Feature Selection in Network Anomaly Detection. *Journal of Applied Sciences Research*, *8*(3), 1748–1756.

Alsattar, H. A., Zaidan, A. A., & Zaidan, B. B. (2019). Novel Meta-Heuristic Bald Eagle Search Optimisation Algorithm. *Artificial Intelligence Review*, 1–28.

Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 446–452.

Dong, B., & Wang, X. (2016). Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection. *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, 581–85. doi:10.1109/ICCSN.2016.7586590

Dwivedi, S., Vardhan, M., Tripathi, S., & Kumar, A. (2020). Implementation of Adaptive Scheme in Evolutionary Technique for Anomaly - Based Intrusion Detection. *Evolutionary Intelligence*, *13*(1), 103–117. doi:10.1007/s12065-019-00293-8

Dwivedi, S., Vardhan, M., Tripathi, S., & Shukla, A. K. (2019). Implementation of Adaptive Scheme in Evolutionary Technique for Anomaly-Based Intrusion Detection. *Evolutionary Intelligence*, 1–15.

Eid, H. F., Darwish, A., Hassanien, A. E., & Tai-hoon, K. (2011). Intelligent Hybrid Anomaly Network Intrusion Detection System. *International Conference on Future Generation Communication and Networking*, 209–18. doi:10.1007/978-3-642-27192-2_25

Enache, A.-C., & Patriciu, V. V. (2014). Intrusions Detection Based on Support Vector Machine Optimized with Swarm Intelligence. *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 153–58. doi:10.1109/SACI.2014.6840052

Faris, H., Aljarah, I., Azmi, M., & Al-betar, S. M. (2018). Grey Wolf Optimizer : A Review of Recent Variants and Applications. *Neural Computing & Applications*, *30*(2), 413–435. doi:10.1007/s00521-017-3272-5

Garg, S., & Batra, S. (2018). Fuzzified Cuckoo Based Clustering Technique for Network Anomaly Detection. *Computers & Electrical Engineering*, *71*, 798–817. doi:10.1016/j.compeleceng.2017.07.008

Ghosh, P., Karmakar, A., Sharma, J., & Phadikar, S. (2019). CS-PSO Based Intrusion Detection System in Cloud Environment. In *Emerging Technologies in Data Mining and Information Security* (pp. 261–269). Springer. doi:10.1007/978-981-13-1951-8_24

Gu, J., Wang, L., Wang, H., & Wang, S. (2019). A Novel Approach to Intrusion Detection Using SVM Ensemble with Feature Augmentation. *Computers & Security*, *86*, 53–62. doi:10.1016/j.cose.2019.05.022

Gul, I., & Hussain, M. (2011). Distributed Cloud Intrusion Detection Model. *International Journal of Advanced Science and Technology*, *34*(38), 135.

Hatef, M. A., Shaker, V., Jabbarpour, M. R., Jung, J., & Zarrabi, H. (2018). HIDCC: A Hybrid Intrusion Detection Approach in Cloud Computing. *Concurrency and Computation*, *30*(3), e4171. doi:10.1002/cpe.4171

He, H., Sun, X., He, H., Zhao, G., He, L., & Ren, J. (2019). A Novel Multimodal-Sequential Approach Based on Multi-View Features for Network Intrusion Detection. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 183207–183221. doi:10.1109/ACCESS.2019.2959131

Ingre, B. (2014). Data Mining Approach for Developing Various Models Based on Types of Attack and Feature Selection as Intrusion Detection Systems (IDS). In *Intelligent Computing, Networking, and Informatics* (pp. 92–96). Springer.

Ingre, B., & Yadav, A. (2015). Performance Analysis of NSL-KDD Dataset Using ANN. *2015 International Conference on Signal Processing and Communication Engineering Systems*, 92–96. doi:10.1109/SPACES.2015.7058223

Jia, W., Muhammad, K., Wang, S.-H., & Zhang, Y.-D. (2019). Five-Category Classification of Pathological Brain Images Based on Deep Stacked Sparse Autoencoder. *Multimedia Tools and Applications*, *78*(4), 4045–4064. doi:10.1007/s11042-017-5174-z

Karimazad, R., & Faraahi, A. (2011). An Anomaly-Based Method for DDoS Attacks Detection Using RBF Neural Networks. *Proceedings of the International Conference on Network and Electronics Engineering.*

Kumar, V. (2019). An Integrated Rule Based Intrusion Detection System: Analysis on UNSW-NB15 Data Set and the Real Time Online Dataset. *Cluster Computing*, 1–22.

Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., & Gan, D. (2017). Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 3491–3508. doi:10.1109/ACCESS.2017.2782159

Mahmud, M. T., Rahman, M. O., Hassan, M. M., Almogren, A., & Zhou, M. (2019). An Efficient Cooperative Medium Access Control Protocol for Wireless IoT Networks in Smart World System. *Journal of Network and Computer Applications*, *133*, 26–38. doi:10.1016/j.jnca.2019.02.011

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A Survey of Intrusion Detection Techniques in Cloud. *Journal of Network and Computer Applications*, *36*(1), 42–57. doi:10.1016/j.jnca.2012.05.003

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

Negandhi, P., Trivedi, Y., & Mangrulkar, R. (2019). Intrusion Detection System Using Random Forest on the NSL-KDD Dataset. In *Emerging Research in Computing, Information, Communication and Applications* (pp. 519–531). Springer. doi:10.1007/978-981-13-6001-5_43

Pajouh, H. H., Dastghaibyfard, G. H., & Hashemi, S. (2017). Two-Tier Network Anomaly Detection Model: A Machine Learning Approach. *Journal of Intelligent Information Systems*, *48*(1), 61–74. doi:10.1007/s10844-015-0388-x

Patil, R., Dudeja, H., & Modi, C. (2019). Designing an Efficient Security Framework for Detecting Intrusions in Virtual Network of Cloud Computing. *Computers & Security*, *85*, 402–422. doi:10.1016/j.cose.2019.05.016

Selvakumar, K., Karuppiah, M., SaiRamesh, L., Islam, S. K. H., Hassan, M. M., Fortino, G., & Choo, K.-K. R. (2019). Intelligent Temporal Classification and Fuzzy Rough Set-Based Feature Selection Algorithm for Intrusion Detection System in WSNs. *Information Sciences*, *497*, 77–90. doi:10.1016/j.ins.2019.05.040

Sharafaldin, Lashkari, & Ghorbani. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *ICISSP*, 108–16.

Taj, M. S., Ullah, S. I., Salam, A., & Khan, W. U. (2020). Enhancing Anomaly Based Intrusion Detection Techniques for Virtualization in Cloud Computing Using Machine Learning. *International Journal of Computer Science and Information Security*, *18*(5).

Tama, B. A., Comuzzi, M., & Rhee, K.-H. (2019). TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 94497–94507. doi:10.1109/ACCESS.2019.2928048

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access: Practical Innovations, Open Solutions*, *5*, 21954–21961. doi:10.1109/ACCESS.2017.2762418

Zhang, H. (2018). An Effective Deep Learning Based Scheme for Network Intrusion Detection. *2018 24th International Conference on Pattern Recognition (ICPR)*, 682–87. doi:10.1109/ICPR.2018.8546162

Zhao, J., & Zhu, Y. (2016). Research on Intrusion Detection Method Based on Som Neural Network in Cloud Environment. *Computer Science and Application*, *6*(8), 505–513. doi:10.12677/CSA.2016.68063

Pankaj Kumar Keserwani is an Assistant Professor, Department of Computer Science and Engineering at National Institute of Technology Sikkim, India. He is pursuing his Ph.D. with an active research interest in Information Security, Machine Learning, Deep Learning, and Cloud Computing. He gets his inspiration from the works of pioneers in the field.

Pilli Emmanuel Shubhakar has 23 years of teaching, research and administrative experience. He was awarded PhD from Indian Institute of Technology, Roorkee for the thesis on "A Framework for Network Forensic Analysis in 2012. He completed a research project "Investigating the Source of Spoofed E- mails" from UCOST, Dehradun in 2016. He has coauthored a book "Fundamentals of Network Forensics - A Research Perspective" for Springer in 2016. He is guiding 7 M. Tech and 11 Ph. D students in Security & Forensics, Cloud Computing, Big Data, IoT, and Blockchain. He is an honorary Dean(Network & Hardware) in Rajasthan ILD Skill University, Jaipur. He is Senior Member of both IEEE and ACM. He is member of Cloud Computing Innovation Council of India (CCICI) and Forensic Science Workgroup on Cloud Computing of the NIST, USA.

Prajjval Govil is pursuing his B.Tech degree in Computer science and Engineering with specialization in Big Data Analytics from JK Lakshmipat University, Jaipur, Rajasthan, India. His research interest includes Machine Learning, Deep Learning, Big data Analytics, Data Science. He has completed more than 10 undergraduate projects on the topic mentioned above.