Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 8478457, 13 pages
https://doi.org/10.1155/2023/8478457

Research Article

Improved Grey Wolf Optimization- (IGWO-) Based Feature Selection on Multiview Features and Enhanced Multimodal-Sequential Network Intrusion Detection Approach

M. Yuvaraja[1] **S. Arunkumar**[2] **P. Vinodh Kumar**[3] **and L. Mary Immaculate Sheela**[4]

[1]*Department of ECE, P.A. College of Engineering and Technology, Pollachi, India*
[2]*Department of EEE, Nehru Institute of Engineering and Technology, Coimbatore 641105, India*
[3]*Department of EEE, Coimbatore Institute of Technology, Coimbatore, India*
[4]*FESAC, Pentecost University, Accra, Ghana*

Correspondence should be addressed to L. Mary Immaculate Sheela; misheela@pentvars.edu.gh

Received 27 July 2022; Revised 12 January 2023; Accepted 13 January 2023; Published 1 February 2023

Academic Editor: Shaohua Wan

The goal of the network intrusion detection system (NIDS) is to spot malicious activity in a network. It seeks to do that by examining the behavior of the traffic network. To find abnormalities, the NIDS heavily use machine learning (ML) and data mining techniques. The performance of NIDSs is significantly impacted by feature selection. This is due to the numerous characteristics that are used in anomaly identification, which take a lot of time. The time required to analyze traffic behavior and raise the accuracy level is thus influenced by the feature selection strategy. In the current work, the researcher's goal was to provide a feature selection model for NIDSs. IGWO (improved grey wolf optimizations) for FSs (feature selections) was proposed to address these difficulties. The three primary processes in this proposed study are preprocessing, extractions and classifications of FSs, and evaluations of results. IGWOs are used to choose a subset of input variables by minimizing features to measure the accuracy in the search space and discover the best solution. A particular structure of HPNs (hierarchical progressive networks) is controlled by the MDAEs (multimodal deep autoencoders) and ABLSTMs (attention-based long short-term memories) for enhanced multimodal-sequential IDSs, i.e., AB-LSTMs. It is possible to understand relationships between neighboring network connections automatically and efficiently integrate information from many levels of characteristics inside a network connection using the EMS-DHPN technique simultaneously. This work's suggested hybrid IDSs called IGWO-EMS-DHPN technique were evaluated using two intrusion datasets: UNSW-NB15 and CICIDS-2017 which is compared with other existing classifiers in terms of relative accuracies, precisions, recalls, and $F1$-scores in categorizations. While several classifiers have been developed, the suggested IGWO-EMS-DHPN classifier obtains maximum accuracy.

1. Introduction

Due to the worldwide internet infrastructures' growing interconnection, security specialists discover additional security flaws every month. As a result, cybercriminals can infiltrate these systems and engage in harmful activities [1]. Identifying different network assaults, particularly unanticipated threats, is an inescapable major issue. Since hackers can access databases containing financial, medical, and other sensitive data, cyber security professionals and designers must devise new IDSs to guard against such attacks. IDSs assist in detecting threats by analyzing packets captured from the network. DoS (denial of services) is a leading common threat, as it prevents legitimate users from accessing network resources by introducing unwanted traffic. On the other hand, malware is malicious software that takes advantage of flaws in computer networks to get its objectives [2].

Internet users have significantly benefited from the developments of ICTs (information and communication technologies). However, network intrusion threats, such as

DDOS, cross-site scripting, and probe, are becoming more and more prevalent, making information security a growing concern. The NIDSs are critical security countermeasures for detecting and preventing malicious intrusions in the cybersecurity industry. Network intrusion detection is a standard classification problem; its goal is to watch network behaviors each moment and assess whether to issue an alarm message to the network [3].

IDSs are critical scientific breakthroughs in information security as they can detect invasions ongoing or even if already occurred. IDSs must determine whether attacks are normal or they are DOSs (denial of services), U2Rs (user to roots), probes, and R2Ls (root to locals) [4]. In short, IDSs enhance classifier performances in identifying invasive behavior. Traditional IDSs monitor traffics using detailed descriptions like rules or signatures where positive and negative false detections were common resulting in false alarms. IDSs (intrusion detection systems) can be based on hosts HIDS (Host IDSs) or based on networks NIDSs (network IDSs) or based on signatures/anomalies AIDSs (anomaly IDSs) or HIDSs (hybrid IDSs) [5]. HIDSs combine features of NIDSs and HIDS and are highly reliable security frameworks [6].

MLTs (machine learning technologies) have been extensively employed to determine various sorts of threats and aid network administrators prevent intrusions. However, most typical MLTs are shallow and focus on feature engineering and selections, making them ineffective for huge intrusion data classification. External learning is also unsuitable for high-dimensional learning with huge data. These challenges led the researchers to seek a better approach [7].

On the other hand, DLTs (deep learning techniques) collect better representations from data to generate significantly improved approaches. DLTs have made significant strides in AI (artificial intelligence) during the last decade. DLTs have also outperformed shallow MLTs in areas like finance, automatic machine translations, speech recognition, and computer vision [8]. Singular DLTs usually perform admirably when large amounts of data are available in computer vision. NIDSs can benefit from DLTs, due to their ability to generalize in new environments while handling voluminous data and thus can be applied for identifying new threats [9]. Therefore, intrusion detection models will inevitably lose some information of traffic data and can only use incomplete feature information to classify because in these methods, the complex feature information within a network connection and the temporal information between network connections were either completely ignored or considered simply.

This research work proposes EMS-DHPN framework to utilize multiple information of traffic data and thus enhancing the effectiveness of AIDSs. MDAEs are designed to understand distributions of subfeature vectors. Also, AB-LSTM technology is used to support the methodology's intelligence. Experimental results validate the developed model. This work's significant contributions are as detailed below:

(1) A novel multiview strategy for minimizing feature complications and investigating advanced multimodal DLTs to develop practical feature fusion units for traffic data. In the information security industry, multimodal DLTs are being used to address the challenges of IDSs

(2) Constructing smart AIDSs with structures of HPNs that fully utilize structured traffic information enhance the accuracy of intrusion recognitions

(3) The suggested approach employs IGWOs for its FSs, which minimizes irrelevant and repetitive data, and important characteristics are selected from search spaces with accuracy and optimal solution as the base. And also, indicating that channels of MDAEs and interpreters of each access channel can be altered to detect in new environments

(4) The recommended EMS-DHPN approach's performance in detecting attacks in current networks is benchmarked using two datasets from 2015 to 2017 where accuracy and robustness in binary and multiclass categorizations were tested

The relevant work in the field of intrusion detection is covered in Section 2. Section 3 largely introduced the recommended multimodal-sequential intrusion detection system. Section 4 assessed the efficiency of the classification algorithm on two datasets and reviewed the experimental investigation. Finally, in Section 5, there is a discussion of the paper's conclusion and future work.

2. Literature Review

Many researchers concentrated on classic MLTs in the early stages of studies where predominantly shallow ANNs (artificial neural networks) were used. FFNNs (feedforward neural networks) built classifiers while BPs (backpropagations) trained network classifiers. IDSs have also been proposed based on MLTs including SVMs (support vector machines), RFs (random forests), and NBs (naive Bayes).

Mohammad [10] suggested grey wolf optimization (GWO) and particle swarm optimization (PSO) for IDS. They developed two novel approaches (PSO-GWO-NB and PSO-GWO-ANN) for FSs and IDS. In addition, this study evaluated the most frequently repeated features of PSO and GWO. For assessments, intrusion datasets were used in this study. Furthermore, two classifiers, namely, NB and ANN, were used in evaluations where their trials showed that MRF features produce good precisions and recalls. Their findings revealed that PSO-GWO-NB classifiers outperformed PSO-GWO-ANN classifiers in FSs and IDSs.

Al-Safi et al. [11] used IGs (information gains), SVMs, ABCs (artificial bee colonies), and CSs (cuckoo searches) for identifying anomalies in networks. Their main steps suggested were FSs and categorizations where FSs were based on IGs, and the best features from the NSL-KDD dataset were chosen. Their proposed technique performed well on modern intrusion datasets (NSLKDD). The study used the UCI dataset extensively as a baseline for the evaluation of

IDSs. The results of classification methods were measured by rates of ACCs (accuracy correct classifications), precisions, and recalls. The proposed method outperformed other modern techniques on the NSLKDD dataset in terms of speed and accuracy.

Aghdam and Kabiri [12] suggested ACOs (ant colony optimizations) and nearest neighbors for intrusion identifications. The study followed FSs and categorizations. Initially, FSs converted TCP dump data into feature sets or vectors. Subsequently, ACOs searched and identified all features. The resulting FSs were evaluated by using smaller feature spaces and assessing classification results. This was also followed by untrained ACOs and neighborhood classifiers for identifying fresh attacks. Finally, precision, recall (or) false positive, *F*-measure, and accuracy evaluated classification methods. The suggested scheme surpassed prior approaches, improving accuracies in identifying intrusion attempts and lowering false alarms with fewer features.

Ali et al. [13] presented fast learning network (FLN) based on PSO called PSO-FLN for their proposed IDS. Their proposed scheme consisted of three major steps where the exploration-exploitation trade-off described the ability to evaluate different regions of problem spaces and find optimums. Secondly, particle-based FLN were used which were created by PSO for training classifications of IDS. Creating particles that represent weighed solutions was the first step in PSO-based optimizations of FLN. To improve accuracy, both weights and neuron counts in hidden layers had to be chosen. Their PSO-FLN model was compared to many metaheuristic techniques to train extreme ML and FLN. In terms of learning accuracies, PSO-FLN outperformed other learning algorithms.

Almomani [14] suggested PSO, GWO, firefly optimization (FFO), genetic algorithm (GA), and SVM for NIDS. Their recommended FS reduced investigation times, increased reliability, and relied on PSO, GWO, FFA, and GA for improving the effectiveness of NIDS. The first preprocessing stage involved removing labels, features, label encoding, and data binarization. Using Anaconda Python Open Source, GAs, PSOs, GWOs, and FFAs were used to generate 13 sets of MI (mutual information) rules. Finally, the model's features were classified using MLTs, namely, SVMs and J48, and tested on the UNSW-NB15 dataset. Their proposed IDS with limited parameters were found to be more reliable.

Kuang et al. [15] developed an approach based on SVMs for IDS where their scheme integrated KPCA (kernel principal component analyses) and GAs. The proposed model determined whether an activity was an attack by employing multilayered SVMs in classifications. The dimensionality of feature vectors was reduced using KPCAs for quicker training of SVMs. Furthermore, noises caused by feature differences were reduced while performances were enhanced. Finally, the tube diameters, kernel parameters, and punishment factor C are used to optimize GAs. The research findings demonstrated that the suggested technique beat existing detection algorithms on the KDD CUP99 dataset in terms of predictive accuracies, convergence speeds, and generalizations.

Zhong et al. [16] designed big data-based hierarchical DL systems (BDHDLS) for IDS. The proposed BDHDLS analyzed network traffics and payloads using behavioral and content features. It worked in three steps: (1) utilizing Apache Spark big data techniques for FSs and clustering, (2) combining both behavioral and content-based features in parallel for improved recognition rated, and (3) utilizing multiple DLTs in a hierarchical tree framework to understand independent traffic patterns for all intrusive attacks. Multimodel approaches instead of single-model approaches can improve intrusion detection rates. The effectiveness of IDSs was measured using three metrics: true positive rates, false positive rates, and accuracies. Their results of tests on the CICIDS2017 dataset showed the time taken to build BDHDLS was significantly reduced as big data techniques use many machines in a parallel training strategy.

Haggag et al. [17] devised min-max normalization, SMOTE (synthetic minority oversampling technique), and FFN (fast fusion networks) for intrusion identifications. Their suggested DLS-IDS approach has four major blocks: dataset selection, dataset preprocessing, class imbalance solutions, and Apache Spark model training. Preprocessing consists of two steps: feature preparation and feature scaling. All features were normalized using min-max. SMOTE took care of class imbalances. Spark had three primary components: driver, cluster management, and worker. FFNs were trained on multilayer perceptron (MLP), RNN (recurrent neural Networks), and LSTMs. NSL-KDD dataset was used in a comparison of Apache Spark with conventional implementation calculation delays. The performance of IDS was measured in terms of accuracy and precision. Their modified model also improved attack recognition rates.

Yin et al. [18] proposed DLTs using RNNs called RNN-IDS for their IDSs. Their study's neuron counts and learning rates affect the model's binary and multiclass categorization performances. Therefore, preprocessing and categorization were the main steps in the proposed study. The first stage used numericalization and normalization techniques. The proposed RNN-IDS model had two parts: forward propagations which computed outputs and BP which transferred residuals to upgrade weights, similar to standard training of NNs. The suggested work compared the proposed scheme with J48, ANNs, RFs, SVMs, and other MLTs where their results showed that the RNN-IDS model enhanced recognition rates of IDS.

Mighan and Kahani [19] proposed the use of MLP, SAE (stacked autoencoder), DT, and SVM for their IDS. Their hybrid scheme combined DLTs with MLTs. Their suggested approach had four processes: (1) data preprocesses, (2) latent feature extractions, (3) threat categorizations, and (4) decisions. The first step of data normalization was min-max normalizations, which removed dimension effects for each attribute. Secondly, SAE extracted latent features, i.e., inferences from other variables instead of direct observations. Thirdly, attack classification used latent features extracted by DLTs in an extensive data framework. Fourth, SVMs were trained on datasets for classifications before which DTs trained datasets to reduce false positive attack detections. Following SAE feature extractions, classification-based intrusion detection algorithms like SVMs, RFs, DTs, and NBs were utilized to detect intrusion in huge network traffic data quickly. The proposed SAE-SVM's performances were satisfactory.

Lopez-Martin et al. [20] conducted intrusion categorization in an IoT context using a conditional VAE (CVAE). This program in IDS was said to be the first to conduct feature reconstruction using CVAE. The NSL-KDD dataset was used for the studies, and the authors said that the model outperforms well-known methods like linear SVM, random forest, and multilayer perceptron's in terms of classification accuracy.

Wu et al. [21] proposed fast object recognition and picture enhancement tasks may be completed using an edge computing and multitask-driven architecture. To encrypt medical pictures and protect patient privacy and the healthcare environment, Wu et al. [22] presented a unique content-aware deoxyribonucleic acid (DNA) computer system. A two-stage DL model for effective NIDS was proposed by Khan et al. [23] using stacked autoencoder with softmax for classification. It was demonstrated that the model could extract useful feature representations from huge amounts of data.

Keserwani et al. [24] introduced an anomaly-based cloud intrusion detection system (IDS) to discover intrusions in a cloud network. The suggested method employs a deep learning strategy for classification and a hybrid metaheuristic algorithm for feature selection. A mixture of the crow search algorithm (CSA) and grey wolf optimization (GWO) pulls pertinent characteristics from the cloud network connection for the deep learning classifier section to process more efficiently. For classification, a deep sparse autoencoder (DSAE) is used. Accuracy, precision, recall or detection rate (DR), and $F1$-score are the parameters taken into consideration for the performance comparison. Sharing sensitive information online has expanded due to the development of network-based services, putting network security at risk. The number of assaults and invaders is always increasing, making detection increasingly difficult. The manual labeling of audit data takes more time, is more expensive, and is laborious. It is crucial to designate the significant characteristic of network traffic for intrusion detection using a classifier that would obtain a greater performance since the capacity to identify important inputs may minimize size and training time and increase accurate results. To determine the best feature subset that increases classification accuracy, we searched the feature space in this work using the grey wolf optimizer, a swarm-based optimization approach.

3. Proposed Methodology

The proposed work includes preprocessing and feature extraction, feature selection, classification, and results in the evaluation. Develop EMS-DHPN with a unique hierarchical progressive network structure for current assault detection. EMS-DHPN has three layers. Assist in integrating complicated features in each traffic flow using a multimodal fusion method based on the MDAE. The second layer uses AB-LSTM to acquire temporal data between traffic flows. To measure the search space's accuracy and locate the ideal solution, describe the preprocessing and feature extraction module that splits complicated characteristics from traffic data. Figure 1 displays the proposed detection approach for EMS-DHPN.

3.1. Preprocessing and Feature Extraction. The preprocessing and feature extraction module extract features from traffic data. First, acquire the traffic database containing previous network behavior. The connection record describes a succession of TCP packets from sources to destinations, defined as $F = (f\,1, f\,2, .., f\,n)$ in which f implies features and n elements counts in each connection record.

Figure 2 depicts the categorization of each record's features based on packets, traffic, or generic. Segmentations ensured sequential relationships between records were preserved and records counts did not change. Then, for each document, get numerous feature groups, each one a vector, $F\,\text{groups} = \{F\,1, F\,2, .., F\,m\}$, which represent counts of feature groups.

The data process divides the large feature vector into smaller features to reduce feature complexity instead of concatenating them all together as in the simple method. The dividing rule is also adaptable to other observing aids and feature vision. Because data features vary between network data monitoring technologies. The UNSW-NB15 and CICIDS 2017 evaluation datasets had 2 and 3 categories of characteristics, respectively.

3.2. Feature Selection. FS is the procedure of minimizing or discovering the most significant inputs for processing and analysis. FS approaches are used to choose salient characteristics to determine the accuracy in the search space and locate the ideal solution. Optimizations using GWOs and IGWOs are detailed below.

3.2.1. GWOs. GWOs replicate the hunting style of the grey wolf pack. Grey wolves have a tight four-level social order, with collections ranging from 5 to 12. GWO is based on the social intelligence of grey wolves that like to live in packs of 5-12. This program simulates GWO's leadership structure using four levels: alpha, beta, delta, and omega. The primary responsibility of the alpha is to make decisions (e.g., hunting, sleeping place, and wake-up time). Beta is known to help alpha make decisions and provide input. A scout or sentinel is a hunter. Omega wolves are controlled by alpha and beta wolves. Omega wolves must obey all wolves.

The dominant wolves in the group are termed alpha (α). The alpha wolves' subordinates are the beta wolves (β). Omega (ω) wolves are the lowest ranking. Omega wolves must submit to all wolves and eat last in a pack. Other wolves in the collection are termed delta (δ) that consent to alpha and beta wolves but dominate omega wolves [25]. In GWOs α, β, and δ, direct hunting procedures and ω wolves obey.

GWO's circling behavior is estimated as follows:

$$\vec{X}(t+1) = \vec{X}_p(t) + \vec{A}.\vec{D}. \tag{1}$$

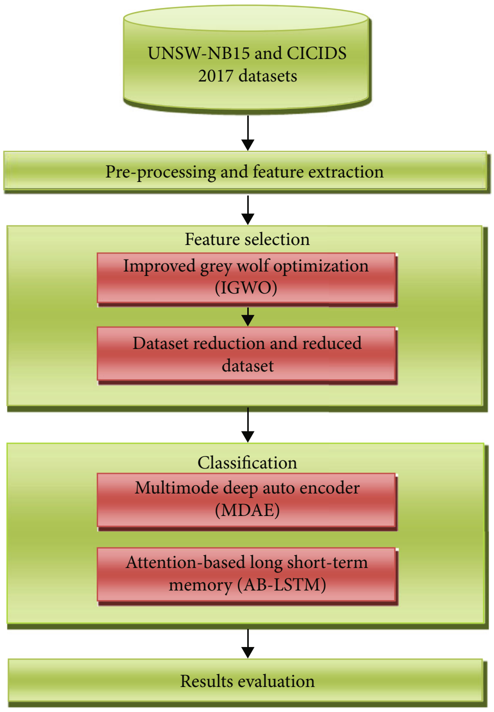
UNSW-NB15 and CICIDS 2017 datasets
Pre-processing and feature extraction
Feature selection
Improved grey wolf optimization (IGWO)
Dataset reduction and reduced dataset
Classification
Multimode deep auto encoder (MDAE)
Attention-based long short-term memory (AB-LSTM)
Results evaluation

Figure 1: Proposed EMS-DHPN framework.

Here $\overrightarrow{AC}$ denote coefficient vectors, $\vec{X}_p$ denote prey's locations vector, X replicates the location of wolves in a d -dimensional space in which d represents the number of variables, (t) represents the iterations number, and $\vec{D}$ is defined as below:

$$\vec{D} = \left|\vec{C}.\vec{X}_p(t) - \vec{X}(t)\right|. \tag{2}$$

Here, $\vec{A}$ and $\vec{C}$ are defined as below:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a}, \tag{3}$$

$$\vec{C} = 2.\vec{r_2}. \tag{4}$$

In which, $\vec{r_1}$ and $\vec{r_2}$ represent vectors randomly in $[0, 1]$.

a denotes encircling coefficient minimized from 2 to 0 as the rounds raise as per the below formula:

$$a = 2 - 2\left(\frac{t}{T}\right), \tag{5}$$

where t represents iterations counts and T stands for max iterations count. For example, grey wolves consider alpha the best candidate for the job, with beta and delta supposed to know the prey's whereabouts. So, until a particular iteration, the best three solutions are preserved, forcing those (e.g., omega) to adjust their locations in the decision space [26].

The upgrading locations concept is computed as below:

$$\vec{X}(t+1) = \frac{\overrightarrow{x_1 + x_2 + x_3}}{3}. \tag{6}$$

In which$x_1, x_2\,; x_3$ is described and computed as below:

$$\begin{aligned} \vec{x_1} &= \vec{X_\alpha} - A_1.\left(\vec{D_\alpha}\right), \\ \vec{x_2} &= \vec{X_\beta} - A_2.\left(\vec{D_\beta}\right), \\ \vec{x_3} &= \vec{X_\delta} - A_3.\left(\vec{D_\delta}\right). \end{aligned} \tag{7}$$

Here, $\vec{x_1}$, $\vec{x_2}$, and $\vec{x_3}$ denote the three optimal wolves in the genetic at a given iteration t. Here, A_1, A_2, and A_3 are evaluated as in the above formulae. $\vec{D_\alpha}$, $\vec{D_\beta}$, and $\vec{D_\delta}$ are computed as follows:

$$\begin{aligned} \vec{D_\alpha} &= \left|\vec{C_1}.\vec{X_\alpha} - \vec{X}\right|, \\ \vec{D_\beta} &= \left|\vec{C_2}.\vec{X_\beta} - \vec{X}\right|, \\ \vec{D_\delta} &= \left|\vec{C_3}.\vec{X_\delta} - \vec{X}\right|, \end{aligned} \tag{8}$$

In which, $\vec{C_1}$, $\vec{C_2}$, and $\vec{C_3}$ are evaluated as per the formula mentioned above.

3.2.2. IGWOs. The GWO uses the best three options to update each wolf's location. Omega wolves make up an enormous population and are less fit than alpha, beta, and delta wolves. Realigning the weaker wolves can increase GWO's diversification capacity and find better results.

In the introduced IGWOs, each generation's wolves are ranked by fitness. They are divided into two groups: enhanced grey wolves and grey wolves. Each upgraded grey wolf has a master wolf from which it learns using the equations given below:

$$D_L = \omega|C_4.(x_M - x_I)|. \tag{9}$$

In which DL denotes the fraction of distance between a master and slave wolf, $\omega\epsilon[0\ 1]$ indicates the learning coefficient, C_4 is evaluated by Equation (4), x_M denotes a master wolf, x_I is enhanced wolf, and S and $M$are computed using Equation (10). For N wolves,

$$I = M + \frac{N}{2} M = 1, 2, 3 \cdots. \tag{10}$$

Following equations, new ceaseless locations of enhanced wolves are calculated using

$$x_n = x_M - A_4.D_L. \tag{11}$$

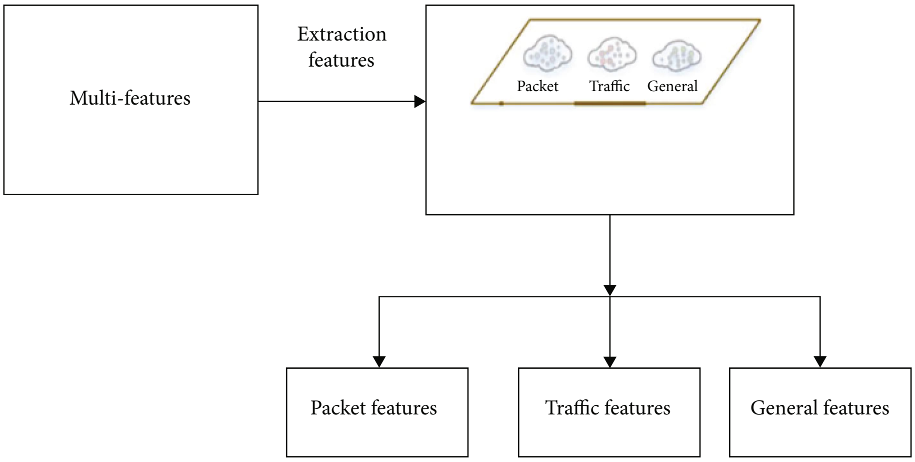

Figure 2: Extraction features from the multifeatures.

Here, x_n is the continuous solution and A_4 is identified in Equation (3).

$$x_{Id}(t+1) = \begin{cases} 1, & \text{if } I(x_n) > \text{rand}, \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

In which x_{Id} is the new binary solution for an improved wolf in dimension d, rand $\epsilon\,[0\,1]$, and I implies sigmoid functions depicted below as

$$I(x) = \frac{1}{1 + \exp\left(-10 * (x - 0.5)\right)}. \tag{13}$$

Equation (12) is used for updating all wolves' locations. However, the rand is now 0.5. Explorations use nonlinear control parameters

$$a = 2\left(1 - \frac{t^2}{T^2}\right). \tag{14}$$

The flow diagram of IGWO is depicted in Figure 3.

3.3. Classification. Classification is the intrusion detection procedure of a particular feature subset. Classifications are sometimes known as targets/labels. Classification is supervised learning with marks and input data (selected features). EMS-DHPN stands for enhanced multimodal sequential intrusion detection. This section has three subsections:

(1) Multimodal fusion model

(2) Sequential learning model

(3) Multimodal real-time model

3.3.1. Multimodal Fusion Model. MDAEs can be used in IDSs using multimodal learning technology [27]. MDAEs assume that associations between traffic flow features are varied and complementary. The architecture of MDAEs is depicted in Figure 4. The number of input channels in the input layer is determined by the feature groups that are specified. Gaussian restricted Boltzmann machines were used as intermediate layers for assessing distributions of input units. The goal is to comprehend the final consensus representation $F^{'} = \{F_{\text{joint}}\}$ given m feature groups $F_{\text{groups}} = \{F_1, F_2, .., F_m\}$.

As demonstrated in Figure 4, MDAE model training processes include forward encodes and backward decodes. Forward encoded compute initial joint representation values and fuse multifeatures. Back decodes adjust weight matrices based on reconstruction errors. RBMs (restricted Boltzmann machines) with their undirected graphical structures use two layers: input and hidden, with the number of hidden neurons varying between ten to hundred and twenty based on inputs. Joint distributions $P(v, h)$ are easily computed via an energy function:

$$\begin{aligned} P(v,h) &= \frac{\exp\left(-E(v,h)\right)}{Z}, \\ E(v,h) &= \frac{1}{2\sigma^2} v^T v - \frac{1}{\sigma^2}\left(c^T v + b^T h + h^T W v\right), \end{aligned} \tag{15}$$

Where Z is a constant value for normalizations while $E(v, h)$ represents energy to function. σ stands for hyperparameter, and W implies weight matrices between visible and hidden layers where c and b are biases for visible and hidden layers, respectively. The conditional probability distributions of the Russian RBM are calculated as below:

$$\begin{aligned} P(h_i = 1|v) &= \text{Sigmoid}(Wv + b), \\ P(v_i|h) &= \mathcal{N}(Wv + b). \end{aligned} \tag{16}$$

Divergence algorithm that trains Gaussian RBMs and RBM parameters $\theta(W, b, c)$ can be obtained using a learning rule depicted as

$$\Delta W = E_{\text{data}}(vh) - E_{\text{model}}(vh), \tag{17}$$

where E_{data} is the expectation perceived in the training data and E_{model} is the expectation perceived in the data

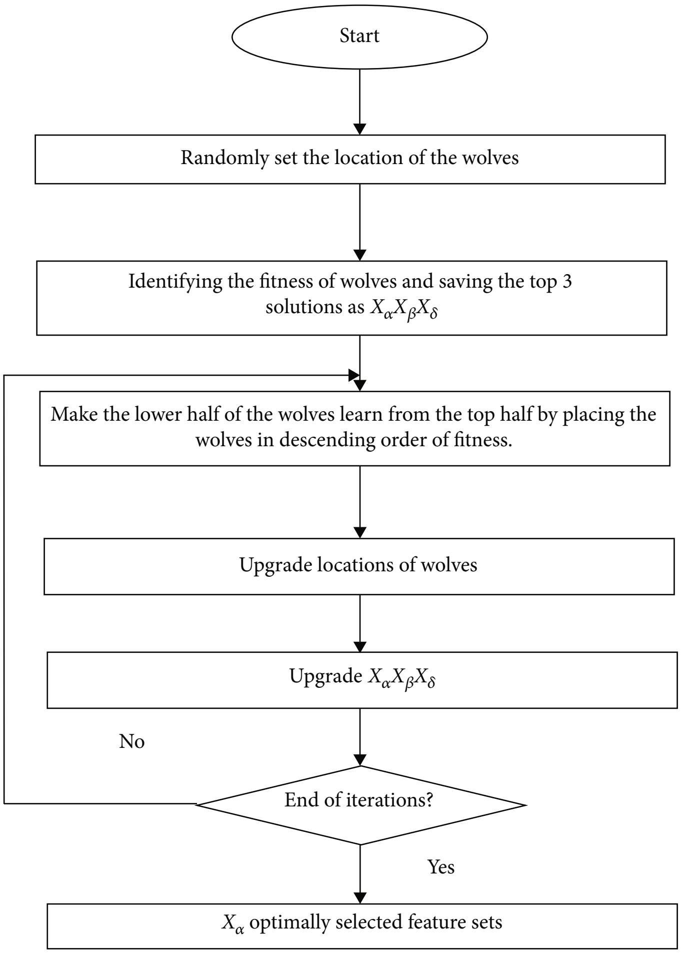
Start
Randomly set the location of the wolves
Identifying the fitness of wolves and saving the top 3 solutions as $X_\alpha X_\beta X_\delta$
Make the lower half of the wolves learn from the top half by placing the wolves in descending order of fitness.
Upgrade locations of wolves
Upgrade $X_\alpha X_\beta X_\delta$
No
End of iterations?
Yes
X_α optimally selected feature sets

Figure 3: Flowchart of IGWO.

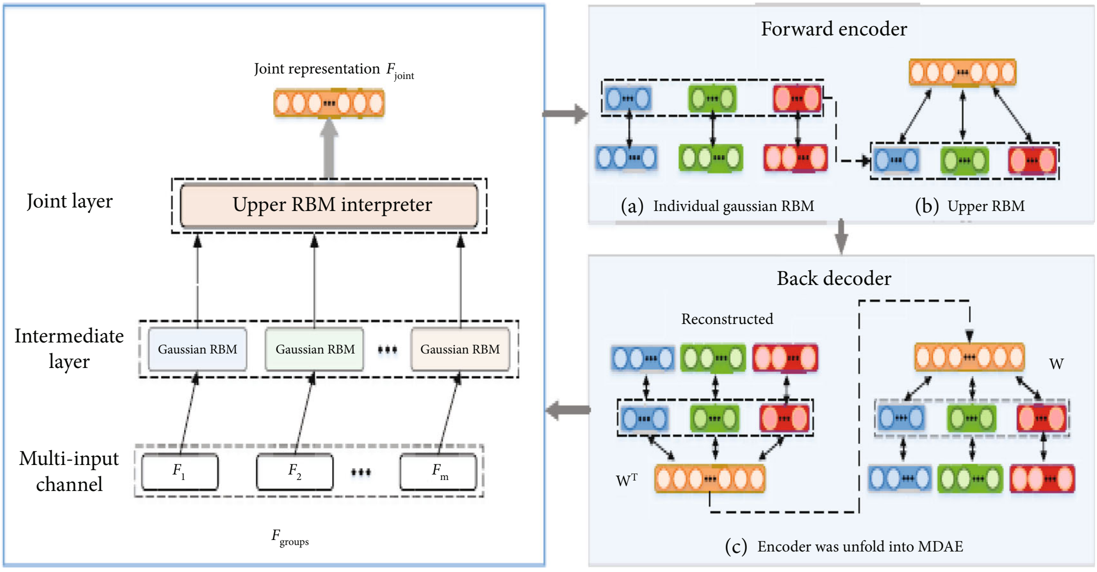

Figure 4: The architecture of NN-based MDAE and construction processes.

```
Input: multifeature sets $(F_1, F_2, .., F_m)$
Output: combined representations $F_{\text{joint}}$
Step 1: forward encodes
for $F = 1$ to $m$ do
    Train Gaussian RBMs for currently chosen feature sets
    Preserve Gaussian RBMs as independent interpreters
    Combine hidden layers of Gaussian RBMs with intermediate layers
    Use intermediate layers as inputs to train subsequent RBMs as joint interpreters
    Acquire preliminary joint representations $F'_{\text{joint}}$
Step 2: backward decodes
    Distribute RBMs uniformly to create MDAEs
    Upgrade MDAEs as per rebuilding faults of picked feature sets
end
```

Algorithm 1: Procedures for training MDAEs.

```
Input: flow information from network's connections $x_i = \{F_1, F_2, F_m\}, (i = 1, 2,., T)$
Output: the probability of categories in flows $y_i$
Step 1: model for multimodal fusions
    Generation of MDAE based on Algorithm 2
    Integration of MDAE with multiple features internally $x'_i = \{F_{\text{fusion}}^{\text{temporal}}\}$
Step 2: model for sequential learning
for $i$ from 1 to $T$ do
    Use STM on externally selected temporal features $x_i^n = \{F_{\text{fusion}}^{\text{temporal}}\}$
end for
Step 3: generate progressive hierarchical networks by joining MDAE with AB-LSTM
    Creating EMS-DHPN by adding softmax layers at the end of AB-LSTMs
    While train not to end do
        Obtain probable outputs $\hat{y}_t$ by $x_i^n = \{F_{\text{fusion}}^{\text{temporal}}\}$
        Calculate cross-entropies $L$ and update EMS-DHPN correspondingly
Step 4: test
    Test the EMS-DHPN model with the network's traffic flows
    Return a probable list of categories
```

Algorithm 2: EMS-DHPN's procedures while testing and training.

Table 1: Main information on two datasets.

Dataset	Features	Labels	Instances of train	Instances of test	Years
UNSW-NB15	42	10	175,341	82,332	2015
CICIDS 2017	83	8	93,500	28,481	2017

generated by the RBM model. Backward decodes involved unfolding forward stacked RBMs into deeper autoencode having multiple inputs/outputs. Algorithm 1 describes forward encoding and its reverse encodes.

3.3.2. Sequential Learning Model. There are two layers in the AB-LSTM model: LSTMs and attentions which learn to find solutions and manage more complex sequences and relationships. There are three nonlinear gating units in LSTM's RNNs, namely, forget, input, and output gates [28]. LSTM storage units decide on obtaining new data while discarding old data. This work added an attention layer to LSTMs for focusing on significant data. Long-term dependences were simulated in AB-LSTM by the addition of an attention layer to outputs from LSTMs which can also serve in detecting intruders. Equation (18) is the network's input gate that contains the level of the new memory. The memory quantity of the forget gate is controlled by Equation (19). Last but not

Table 2: Confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	Tp	Fp
	Negative	Fn	Tn

Table 3: Result comparison of classification methods vs. datasets.

Datasets	Methods			Metrics		
		Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)	FAR (%)
UNSW-NB15	PSO-FLN	79.9	80.1	80.8	81.8	0.9
	MS-DHPN	84.4	86.2	85.3	86.2	0.7
	IGWO-EMS-DHPN	86.5	87.5	88.6	92.6	0.4
CICIDS 2017	PSO-FLN	96.8	97.1	97.3	97.5	0.6
	MS-DHPN	98.2	98.1	98.1	98.6	0.3
	IGWO-EMS-DHPN	98.9	98.9	99.3	99.7	0.2

least, Equation (22) changes the memory storage of output gates while LSTMs calculate control states h_i and cell states c_i.

$$i_i = \sigma(W_i * [h_{i-1}, x_i] + b_i), \tag{18}$$

$$f_i = \sigma\left(W_f * [h_{i-1}, x_i] + b_f\right), \tag{19}$$

$$\tilde{c}_i = \tan h(W_c * [h_{i-1}, x_i] + b_c), \tag{20}$$

$$c_i = f_i \bigotimes c_{i-1} + i_i \otimes \tilde{c}_i, \tag{21}$$

$$o_i = \sigma(W_O * [h_{i-1}, x_i] + b_O), \tag{22}$$

$$h_i = o_i \bigotimes \tan h(c_i), \tag{23}$$

where σ and $\tan h$ represent the Sigmoid and activation function, respectively; $\bigotimes$ denotes multiplication of the elements; $[h_i, x_i]$ is h_{i-1} and x_i concatenation; W_i, W_f, W_c, and W_o denote learning weight parameters; b_i, b_f, b_c, and b_o represent learning bias parameters. Also, an attention mechanism improves the LSTM model's accuracy [29, 30]. To choose the critical outputs from prior layers, the attention layer helps. It helps networks focus on important information. The attention model's functions are as follows:

$$\begin{aligned} M &= \tanh \begin{bmatrix} W_h H \\ W_u u_a \otimes e_n \end{bmatrix}, \\ \alpha &= \operatorname{sofmax}\left(\omega^T M\right), \\ r &= H\alpha^T, \end{aligned} \tag{24}$$

where H is the matrix of picked features $[l_{t1}, l_{t2}, \cdots, l_{tn}]$, $e_n \varepsilon R^n$ is a vector, u_a is the embedded attention mechanism, α is the vector form of picked features H attention weights, and r is the resultant attention model's weighted sum of picked features H.

3.3.3. Multimodal Real-Time Model. Multimodal fusion and sequential learning algorithms can gather and combine network flow data with improving threat identification performance. They can only express part of the traffic data. To address this issue, EMS-DHPN was developed. To make EMS-DHPN more usable, flexible MDAEs were proposed. To classify network traffics, two layered MDAEs are built ahead of time, and softmax functions are used at the end of AB-LSTMs. The loss functions also compute the differences between actual and predicted labels y_t. The cross-entropy loss function is used for binary classification:

$$L = -\sum_{t=1}^{T} y_t \log(\widehat{y}_t) + (1 - y_t)\log(1 - \widehat{y}_t). \tag{25}$$

The entropy computations for assessments of losses in the case of multiple class classifications are depicted below:

$$L = -\sum_{t=1}^{T} y_t \log(\widehat{y}_t). \tag{26}$$

Algorithm 2 is described below.

4. Results and Discussion

This section includes performance assessments of the EMS-DHPN model on the CICIDS2017 dataset produced by the Australian Centre of Cyber Security (ACCS), as well as stage-by-stage experimental findings for each model used in the study to test IDSs [31] and UNSW-NB15 dataset of Canadian Institutes for Cyber security (CIC) [32, 33] where initially, MDAEs were used to learn multimodal feature representations from traffic data.

The final experiment used EMS-DHPN approaches to detect attacks. The proposed IGWO-EMS-DHPN technique for detecting attacks in modern networks was evaluated. Table 1 summarizes two datasets. Algorithmic categorizations were compared in terms of accuracies, precisions, recalls, and $F1$-scores (or) F-measures.

4.1. Evaluation Metrics. The suggested approach's performances were measured using relative accuracies, precision, recall, and $F1$ scores with accuracy being the most significant performance indicator. The suggested experiment also uses the classification model's confusion matrix.

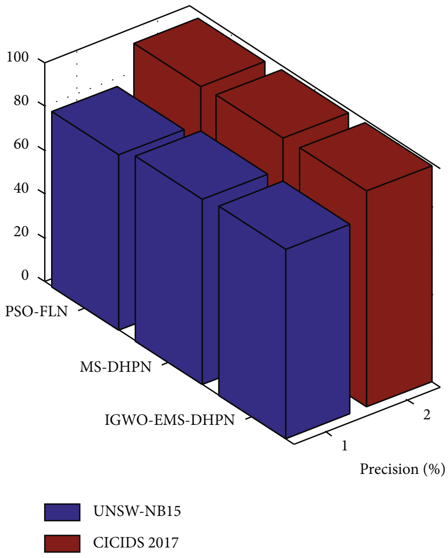

FIGURE 5: Precision performance comparison in various classification methods.

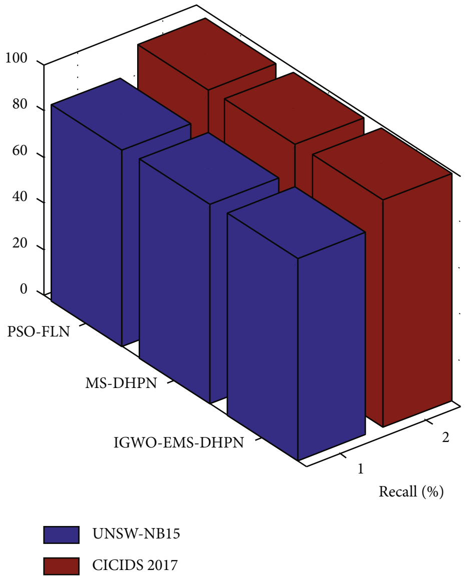

FIGURE 6: Recall performance comparison in various classification methods.

The confusion matrix in Table 2 determines the metric used for two class categorizations. For example, two rows and two columns in a confusion matrix (Table 2) represent the number of Fp, Fn, Tp, and Tn in predictive analytics (Tn).

TP represents attack record counts that were correctly classified as attacks while TN implies normal records classified correctly. FP stands for inaccurately classified normal records as attacks while FN denotes attack record counts that were incorrectly classified as normal records. Using these four, the following metrics were computed for examining the efficiency of classifiers.

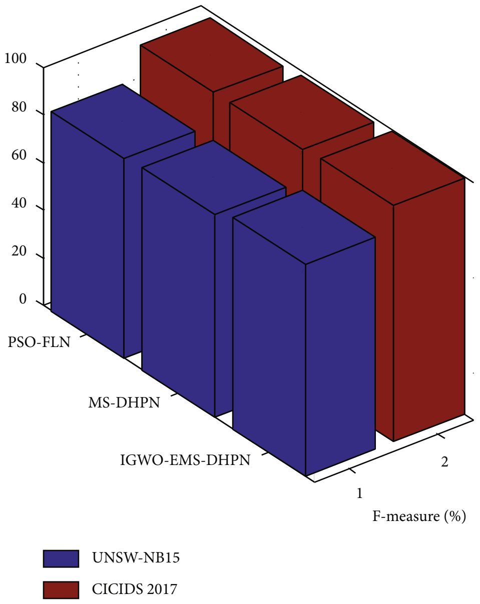

FIGURE 7: *F*-measure performance comparison in various classification methods.

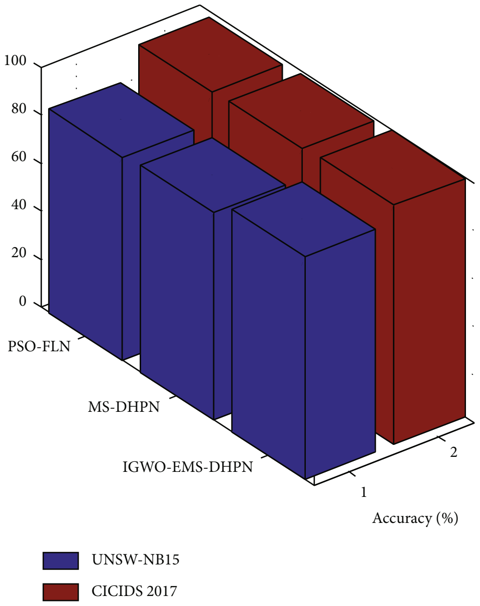

FIGURE 8: Accuracy performance comparison in various classification methods.

4.1.1. *Accuracy.*

Accuracy is the rate of all records accurately categorized total records.

$$\text{Accuracy} = \frac{\text{Tp} + \text{Tn}}{\text{Tp} + \text{Tn} + \text{Fp} + \text{Fn}}. \tag{27}$$

4.1.2. *Precision.*

Precision is the rate of the accurately recognized threat records in all detected threats records.

$$\text{Precision} = \frac{\text{Tp}}{\text{Tp} + \text{Fp}}. \tag{28}$$

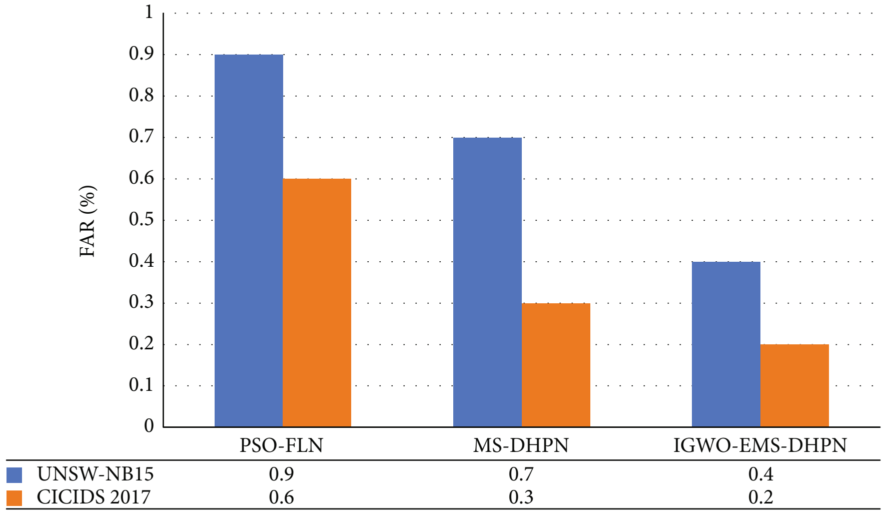

Figure 9: FAR performance comparison in various classification methods.

4.1.3. Recall. Recall is the rate of the accurately recognized threat records in all threat records, also known as the true positive rate (TPR).

$$\text{Recall} = \frac{\text{Tp}}{\text{Tp} + \text{Fn}}. \quad (29)$$

4.1.4. F1-Score or F-Measure. $F1$-score or F-measure is the harmonic mean of precision and recall.

$$\text{F1-score (or)F-measure} = \frac{2(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}}. \quad (30)$$

4.1.5. False Alarm Rate. FAR is the likelihood of incorrectly rejecting the null hypothesis for a given test is measured by a false positive ratio, also known as a fall-out ratio or false alarm ratio.

$$\text{FAR} = \frac{F_p}{F_p + T_p}. \quad (31)$$

4.2. Result Comparison. Table 3 lists comparative experimental results obtained by classifiers on the intrusion detection datasets.

4.2.1. Precision Result Comparison. As shown in Figure 5, the suggested IGWO-EMS-DHPN and existing methods like PSO-FLN and MS-DHPN are compared in terms of precision. Compared to existing approaches, the recommended approach has high precision rates of 86.5% and 98.9%, two separate datasets. In contrast, traditional techniques like PSO-FLN, MS-DHPN, and MS-DHPN have lower precision of 79.9%, 96.8%, 84.4%, and 98.2%, respectively. Thus, the suggested approach is valuable and practical for recognizing short-term threats.

4.2.2. Recall Result Comparison. Figure 6 depicts the recall analysis of the proposed suggested IGWO-EMS-DHPN and other categorization methods, such as PSO-FLN and MS-DHPN. There are high recall rates of 87.5% for one dataset and 98.9% for the IGWO-EMS-DHPN approach. We know that IGWO-EMS-DHPN can obtain high recall rates, which indicates a high detection rate, whereas traditional techniques like PSO-FLN and MS-DHPN provide lower recall rates of 80.1% and 97.1% and 86.2% and 98.1%, respectively.

4.2.3. F-Measure Result Comparison. F-measure comparisons between the suggested IGWO-EMS-DHPN and traditional approaches, such as PSO-FLN and MS-DHPN, are shown in Figure 7. IGWO-EMS-DHPN is well known for its high F-measure, exhibiting excellent attack detection based on the results. Compared to other methods like PSO-FLN and MS-DHPN, which provide F-measure rates of 80.8% and 97.3% and 85.3% and 98.1%, the proposed work can provide better attack detection results with two different datasets than the other previous techniques.

4.2.4. Accuracy Result Comparison. Figure 8 depicts a comparison of the accuracy of various classification techniques, including the proposed IGWO-EMS-DHPN and other current methods such as PSO-FLN and MS-DHPN. According to the graph, the proposed method has high accuracy when compared to previous techniques. The suggested IGWO-EMS-DHPN is an excellent method of accurately detecting attacks, with high accuracy rates of 92.6% and 99.7% for two separate datasets. When comparing the accuracy of existing approaches, PSO-FLN and MS-DHPN provide lower rates of 81.8%, 97.5%, and 86.2%, respectively. The experiments demonstrated that the suggested system is far superior to the conventional techniques.

4.2.5. FAR Result Comparison. Figure 9 depicts a comparison of the FAR of various classification techniques, including the proposed IGWO-EMS-DHPN and other current methods such as PSO-FLN and MS-DHPN. According to the graph, the proposed method has lesser FAR when compared to previous techniques. The suggested IGWO-EMS-DHPN is an excellent method of accurately detecting attacks, with lesser FAR rates for two separate datasets.

5. Conclusion and Future Work

The proposed work uses MDAE and AB-LSTM learning to classify attacks (or intrusions) using UNSW-NB15 and CICIDS 2017 datasets. Each of the four steps in the proposed IGWO-EMS-DHPN framework is described below. Preprocessing and feature extraction stages can be processed separately. A second feature selection step allows IGWOs to narrow the search field and locate the best solution. Finally, MDAE and AB-LSTM enable EMS-DHPN. A network connection using the EMS-DHPN technique may efficiently integrate multiple levels of selected features while learning temporal information across nearby network connections. The IGWO-EMS-DHPN detection technique was developed for attack detection and tested on two intrusion datasets. The findings of several experiments were compared to other methodologies. The suggested IGWO-EMS-DHPN was examined along with other techniques and compared in terms of accuracies, precisions, recalls, and F-measures. Compared to existing PSO-FLN and MS-DHPN, the suggested IGWO-EMS-DHPN achieves high accuracy rates of 92.6% and 99.7% for two separate datasets. Create your traffic gathering system. Find new assaults to validate the suggested model and research data multimodality in information security from a more fundamental perspective to increase intrusion recognition rate.

Data Availability

The (CICIDS2017 dataset) information used to support the study's conclusions has been deposited in the (kaggle) repository (10.1109/ACCESS.2020.3009843/https://www.kaggle.com/datasets/cicdataset/cicids2017). The (UNSW-NB15 dataset) data used to support the findings of this study have been deposited in the (UNSW-NB15 dataset) repository (10.1109/MilCIS.2015.7348942).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Li, Y. Yu, S. Bai, Y. Hou, and X. Chen, "An effective two-step intrusion detection approach based on binary classification and k-NN," *IEEE Access*, vol. 6, pp. 12060–12073, 2018.

[2] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.

[3] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[4] W. Zhong and F. Gu, "A multi-level deep learning system for malware detection," *Expert Systems with Applications*, vol. 133, pp. 151–162, 2019.

[5] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13546–13560, 2019.

[6] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[7] S. M. Kasongo and Y. Sun, "A deep learning method with filter based feature engineering for wireless intrusion detection system," *IEEE access*, vol. 7, pp. 38597–38607, 2019.

[8] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," *Ieee Access*, vol. 6, pp. 13624–13631, 2018.

[9] S. M. H. Bamakan, B. Amiri, M. Mirzabagheri, and Y. Shi, "A new intrusion detection approach using PSO based multiple criteria linear programming," *Procedia Computer Science*, vol. 55, pp. 231–237, 2015.

[10] A. H. Mohammad, "Intrusion detection using a new hybrid feature selection model," *Intelligent Automation And Soft Computing*, vol. 29, no. 3, pp. 65–80, 2021.

[11] A. H. S. Al-Safi, Z. I. R. Hani, and M. M. A. Zahra, "Using a hybrid algorithm and feature selection for network anomaly intrusion detection," *Journal of Mechanical Engineering Research and Developments*, vol. 44, pp. 253–262, 2021.

[12] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security*, vol. 18, no. 3, pp. 420–432, 2016.

[13] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.

[14] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA Algorithms," *Symmetry*, vol. 12, no. 6, pp. 1046–1065, 2020.

[15] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing*, vol. 18, pp. 178–184, 2014.

[16] W. Zhong, N. Yu, and C. Ai, "Applying big data based deep learning system to intrusion detection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181–195, 2020.

[17] M. Haggag, M. M. Tantawy, and M. M. El-Soudani, "Implementing a deep learning model for intrusion detection on apache spark platform," *IEEE Access*, vol. 8, pp. 163660–163672, 2020.

[18] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21954–21961, 2017.

[19] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *International Journal of Information Security*, vol. 20, no. 3, pp. 387–403, 2021.

[20] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT," *Sensors*, vol. 17, no. 9, p. 1967, 2017.

[21] Y. Wu, H. Guo, C. Chakraborty, M. Khosravi, S. Berretti, and S. Wan, "Edge computing driven low-light image dynamic enhancement for object detection," *IEEE Transactions on Network Science and Engineering*, 2022.

[22] Y. Wu, L. Zhang, S. Berretti, and S. Wan, "Medical image encryption by content-aware DNA computing for secure healthcare," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2089–2098, 2023.

[23] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[24] P. K. Keserwani, M. C. Govil, and E. S. Pilli, "An optimal intrusion detection system using GWO-CSA-DSAE model," *Cyber-Physical Systems*, vol. 7, no. 4, pp. 197–220, 2021.

[25] K. Chaimaa, M. Yahlali, M. A. Boudia, A. Amine, R. M. Hamou, and K. Siham, "Intrusion detection system with grey wolf optimizer (GWO)," *International Journal of Informatics and Applied Mathematics*, vol. 2, no. 2, pp. 45–60, 2020.

[26] E. Momanyi and D. Segera, "A master-slave binary grey wolf optimizer for optimal feature selection in biomedical data classification," *Bio Med Research International*, vol. 2021, pp. 1–12, 2021.

[27] H. Wang, J. Wang, C. Dong, Y. Lian, D. Liu, and Z. Yan, "A novel approach for drug-target interactions prediction based on multimodal deep autoencoder," *Frontiers in Pharmacology*, vol. 10, pp. 1592–1610, 2020.

[28] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," in *The International Conference on Learning Representations (ICLR)*, pp. 1–6, San Diego, CA, USA, 2015.

[29] Y. Jin, D. Wu, and W. Guo, "Attention-based LSTM with filter mechanism for entity relation classification," *Symmetry*, vol. 12, no. 10, pp. 1729–1743, 2020.

[30] T. Mamo and F. K. Wang, "Attention-based long short-term memory recurrent neural network for capacity degradation of lithium-ion batteries," *Batteries*, vol. 7, no. 4, pp. 66–74, 2021.

[31] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[32] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, vol. 1, pp. 108–116, Hyderabad, India, December 2018.

[33] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (Mil CIS)*, Canberra, ACT, Australia, November 2015.