Salud, Ciencia y Tecnología - Serie de Conferencias. 2024; 3:762
doi: 10.56294/sctconf2024762

Category: STEM (Science, Technology, Engineering and Mathematics)

ORIGINAL

A Hybrid Rider Optimization with Deep Learning Driven Intrusion Detection Farmwork in Wireless Sensor Network

Optimización de un usuario híbrido con trabajo agrícola de detección de intrusiones impulsado por aprendizaje profundo en una red de sensores inalámbricos

K Sedhuramalingam[1], Dr.N Saravana Kumar[1]

[1]Assistant Professor, Electronics and Communication Engineering, Arjun College of Technology. Coimbatore, Tamil Nadu, India.
[2]Associate Professor, Electronics and Communication Engineering, Dr. Mahalingam college of Engineering and Technology. Coimbatore, Tamil Nadu, India.

Cite as: Sedhuramalingam K, Saravana Kumar D. A Hybrid Rider Optimization with Deep Learning Driven Intrusion Detection Farmwork in Wireless Sensor Network. Salud, Ciencia y Tecnología - Serie de Conferencias. 2024; 3:762. https://doi.org/10.56294/sctconf2024762

Submitted: 30-12-2023 **Revised:** 18-03-2024 **Accepted:** 12-05-2024 **Published:** 13-05-2024

Editor: Dr. William Castillo-González

ABSTRACT

Introduction: an array of hazards currently exists in cyberspace, prompting extensive research to tackle these concerns. Intrusion Detection Systems (IDS) are a mechanism used to provide security in Wireless Sensor Networks (WSN). The IDS continue to encounter significant challenges in accurately identifying unknown attacks. Conventional Intrusion Detection Systems (IDS) commonly rely on Deep Learning (DL) algorithms, which utilise binary classifiers to classify attacks. The data dimension attribute is affected inside large-scale high-dimensional data sets.
Methods: this research introduces a hybrid GFSO (HGFSO) model combined with Deep Learning Driven Intrusion Detection (HGFSO-DLIDS) to tackle this problem. The HGFSO approach is developed by merging the parameter selection methods of the Felis Margarita Swarm Optimisation (FMSO), the Grampus optimisation algorithm (GOA), and the Deep Convolutional Neural Network (DCNN) with BiLSTM (Bidirectional Long Short-Term Memory) algorithm.
Results: the model training utilised real-time traffic statistics, including the KDDCup 99 and WSN-DS datasets. After being trained and validated using the datasets, the model's performance is assessed by multi-class classification, achieving accuracy rates of 99,89 % and 99,64 % respectively.
Conclusion: as a result, this occurrence leads to a decrease in the overall effectiveness of detecting assaults. Deep learning may enhance the creation of an intrusion detection system by eliminating complex features in the raw data, resulting in a more precise classification method.

Keywords: WSN (Wireless Sensor Network); IDS (Intrusion Detection System); DL (Deep Learning); Hybrid Optimization; DCNN (Deep Convolutional Neural Network); BiLSTM (Bidirectional Lon Short Term Memory).

RESUMEN

Introducción: actualmente existe una variedad de peligros en el ciberespacio, lo que ha llevado a una investigación exhaustiva para abordar estas preocupaciones. Los Sistemas de Detección de Intrusos (IDS) son un mecanismo utilizado para brindar seguridad en las Redes de Sensores Inalámbricos (WSN). El IDS continúa enfrentando desafíos importantes a la hora de identificar con precisión ataques desconocidos. Los sistemas de detección de intrusiones (IDS) convencionales comúnmente se basan en algoritmos de aprendizaje profundo (DL), que utilizan clasificadores binarios para clasificar los ataques. El atributo de dimensión de datos se ve afectado dentro de conjuntos de datos de alta dimensión a gran escala.

Métodos: esta investigación presenta un modelo híbrido GFSO (HGFSO) combinado con detección de intrusiones impulsada por aprendizaje profundo (HGFSO-DLIDS) para abordar este problema. El enfoque HGFSO se desarrolla fusionando los métodos de selección de parámetros de Felis Margarita Swarm Optimization (FMSO), el algoritmo de optimización Grampus (GOA) y la red neuronal convolucional profunda (DCNN) con el algoritmo BiLSTM (memoria bidireccional a largo plazo y corto plazo).
Resultados: el entrenamiento del modelo utilizó estadísticas de tráfico en tiempo real, incluidos los conjuntos de datos KDDCup 99 y WSN-DS. Después de ser entrenado y validado utilizando los conjuntos de datos, el rendimiento del modelo se evalúa mediante una clasificación de clases múltiples, logrando tasas de precisión del 99,89 % y 99,64 % respectivamente.
Conclusión: como resultado, este hecho conduce a una disminución en la efectividad general de la detección de agresiones. El aprendizaje profundo puede mejorar la creación de un sistema de detección de intrusiones al eliminar características complejas en los datos sin procesar, lo que da como resultado un método de clasificación más preciso.

Palabras clave: WSN (Red de Sensores Inalámbricos); IDS (Sistema de Detección de Intrusiones); DL (Aprendizaje Profundo); Optimización Híbrida; DCNN (Red Neuronal Convolucional Profunda); BiLSTM (Memoria Bidireccional Lon a Corto Plazo).

INTRODUCTION

WSN have gained lot of interest in the realm of academic research owing to its extensive array of real-time applications, including but not limited to essential armed investigation, battlegrounds, creating security observing, forest fire observing, and healthcare.[1] It contains several independent sensor nodes that are strategically deployed throughout diverse ROI (Regions Of Interest). These sensor nodes are responsible for collecting vital data, which is then wirelessly sent to a central node named as the sink node or Base Station (BS).[2,3] Specific WSN protocols are required for the data that is sent over the network. In light of this, it is crucial to safeguard WSNs from various security risks. Its restricted properties, particularly their capacity of battery, memory, and energy consumption, make it difficult to accomplish this objective, which is unfortunate. [4] Traditional security solutions like encryption are often insufficient for those net because of their limiting resources.

The wide, distributed behavior of WSNs and the limited properties of the sensor nodes make them very susceptible to assaults. A WSN attacker adversary may readily be injected into a WSN since WSNs need frequent packet broadcasting and the deployment of sensor nodes at random in the environment.[5] A sensor node may be occupied through an attacker, also waste network resources and affect data integrity while listening in on conversations and sending false information. A communal and severe attack that jeopardizes the security of WSNs is the DoS (Denial of Service) attack. This assault takes many different procedures, and its vital goal is to stop or halt the facilities offered through WSNs.[6,7] Since preventing or blocking threats to safety is not always possible, an intrusion detection system (IDS) must recognize and warn sensor nodes of known and new attacks.

IDS may spot odd or suspicious activities and sound a signal when an intrusion happens. Installing IDSs on WSNs is more difficult than on other types of networks since sensor nodes are typically made to be cheap, compact, and low on resources for hardware. Furthermore, the WSN lacks a specialized dataset with typical profiles and assaults that may be utilized to recognize an assailant's sign[3]. When building IDS for WSNs, there are primarily two requirements to take into account: The IDS has to be very accurate in detecting intrusions, even unidentified assaults, and lightweight to put as little strain as possible on the WSN infrastructure.[8]

IDS has recently changed as a result of the fusion of many applications, particularly those that use DL and ML (Machine Learning), which are both features of AI (Artificial Intelligence). The IP-enabled wireless network may be attacked and protected using the IDS inherited by the application. To reduce the effects or adversities of accidents that occur in the WSN, these specific algorithms are, nevertheless, manipulated for defensive mechanisms and persist against security threats. For intrusion detection, various ML and DL-based IDS have been created,[5,6,7] malware recognition,[12,13] cyber-physical assaults,[14] and information security.[15] Machine learning (ML) approaches are generally used to generate error-free clustering, classification, and prediction models.[16] Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Gaussian Nave Bayes (GNB), which are used to predict any form of suspicious assaults, are some examples of machine learning (ML) techniques that play a significant role in intrusion detection in WSN. At various degrees of abstraction, ML employs AI to carry out tasks without requiring human participation.

Depending on the model's structure and makeup, ML is extensively categorized. Supervised Learning (SL), Reinforcement Learning (RL), and Unsupervised Learning are the categories under which it falls. Dwivedi et al.[17] recommended the use of ML algorithms to identify malicious WSN attacks. The study of the data from the

https://doi.org/10.56294/sctconf2024762

inquiry provides the merits and parameters. To find the connection between input and output pairs, several researchers use supervised learning. The process's top output is calculated together with its corresponding input at the conclusion. Applications as diverse as intrusion detection (ID) and data categorization use deep learning techniques.[18] A trained classification model categorizes network traffic as either being attacked or being in the normal class, which is how ID in IoT networks is classified as a Binary classifier. The ultimate objective is to attain the highest Accuracy while lowering the false alarm rates. Restrictive requirements, such as low-level processing capabilities, high-volume data processing, and quick reaction times, must be met by the IDS in IoT-centric smart settings. Because cyberattacks are growing more complex every day, describing knowledge discovery with data mining might help build IDS with resilient behavior and greater accuracy than traditional IDS, which could not be as strong as this paradigm.

The pre-processing stage, particularly feature selection, was largely neglected by many of them. Consequently, the classification accuracy of the specific Algorithm is directly impacted. Additionally, the incorrect pre-processing step increases the training period of the algorithm. The back-propagation method used by the current neural network algorithm contributes to an increase in training time as well. DL-based enhanced IDS model for IDS for WSN using hybrid optimization. Two methods are suggested in this work to overcome the above problems: One is for the Classification Module, the other is for Feature Selection. These are the paper's main contributions:

- The HGFSO-DLIDS algorithm utilizes GOA and FMSO to select the hyperparameters of the DCNN-BiLSTM.
- Additionally, to increase Classification Accuracy, a Deep Learning-based Hybrid DCNN-BiLSTM is developed, which calculates optimal weights using the HGFSO Algorithm.
- Finally, the suggested algorithm is evaluated by means of numerous metrics, including Accuracy, Precision, Recall, F-score, Training Time, etc., and exhibits superior performance than the methods currently in use.

The following is how the paper is set up: A short synopsis of the prior contributions is provided in Section 2 before discussing the study. The suggested HGFSO-DLIDS is used in Section 3 in a thorough way. The experimental results on several measures are shown in Section 4 along with comparisons to current methods. In conclusion, Section 5 reviews the conclusions of the planned research and offers suggestions for further research.

Related work

Sharma et al.[19] suggested a new IDS, concentrating on the wireless sensor network, that incorporates the operational and developmental frameworks. Features are extracted and categorized as part of the proposed DevOps-based intrusion detection approach. Early processing of application data includes the feature extraction step, which combines existing characteristics with statistics and higher-order descriptors. The classification technique then makes use of the retrieved features together with an updated DCNN strategy. The approach optimizes filter number and size in input vector and totally connected layers. Critical assets are also in danger from unauthorized system access by fraudsters or intruders in addition to security breaches. Consequently, it is crucial to continue identifying and thwarting possible dangers in the wireless environment.

Gowdhaman et al.[20] presented the DNN (Deep Neural Network)-IDS. Cross-correlation is used to choose the optimal dataset properties to create deep neural networks to detect intrusions. The recommended DNN outperformed SVMs, DTs, and RFs in assault recognition. However, resource-constrained nodes, deployment methods, and communication routes make wireless sensor networks security concerns. The identification of unwanted access is of paramount importance in augmenting the WSNs security features.

Abhale et al.[21] constructed supervised classification models for ID using tools like the RF classifier, SVM, DT classifier, LGBM classifier, Extra Tree classifier, GBC (Gradient Boosting Classifier), Ada boost classifier, KNN classifier, MLP classifier, GNB classifier, and LR classifier. The data set used to test these methods is the NSLKDD or Modified KDD99 Data Set. Research indicates that the SVM has the greatest accuracy in comparison to other categorization methods. Resource-constrained nodes, deployment methods, and communication routes make wireless sensor networks security concerns. Unauthorized access must be identified to improve wireless sensor network security.

Sood et al.[22] CGAN (Conditional Generative Adversarial Network), a DL technique, was used in the suggested IDS model to enable unsupervised learning. An XGBoost (EXtreme Gradient Boosting) classifier was also involved in the model for quicker contrast and outcome presentation. Because the suggested system creates this bogus data, the proposed solution may minimize the requirement to install more sensors by 1,2-2,6 %. The settings were chosen to provide our model with the best outcomes possible while minimizing substantial changes and problems. Through the support of a multi-layer network, the algorithm learns from dataset models for an enhanced training process. In addition, the suggested approach attempted to improve accuracy and reduce false detection in NSL-KDD and CICIDS2017 datasets to identify cyber intrusions. The suggested model has an

about 1,827 % lower false alarm rate. The primary restriction of the study effort is the important and difficult procedure of matching all the patterns for massive networks.

Biswas et al.[23] a wireless sensor network intrusion detection technique utilized GNN (Graph Neural Networks) and Lyapunov optimization. The weights of the synapses between two neurons should be optimized using Lyapunov optimization during the training phase of the GNN. For the GNN, AWID datasets were employed via training and testing. In order to reduce loss, weight is adjusted in accordance with the results of the Lyapunov optimization method. Additionally, displays test results of our methodology utilizing the Accuracy, Sensitivity, Precision, and F1 Score performance matrices. Our technique provides improved detection accuracy when compared to prior studies, as seen by the comparison. An intruder may simply seize and alter sensor nodes placed in a distant area. In the framework of WSNs, ID is thus still a serious problem.

Otoum et al.[24] presented a comprehensive study of WSN IDS systems using ML and DL. Restricted Boltzmann Clustered-IDS (RBC-IDS), a DL-IDS for WSN monitoring of crucial substructures, is an additional choice. Additionally, examine the efficiency of RBC-IDS and relate it to the ASCH-IDS (Adaptively Supervised and Clustered Hybrid IDS), an adaptive ML-IDS that was previously recommended. Although RBC-IDS's detection time is almost twice as long as ASCH-IDS's, numerical data reveal that both technologies attain the similar recognition and accuracy ratios. The introduction of a variety of difficulties that might have a detrimental effect on WSN performance is brought about by the adoption of IDS in WSNs, however. It's also important to take into account attacks on WSN layers and protocols other than LEACH.

Zhao et al.[25] suggested a light-weight dynamic autoencoder network (LDAN) approach for NID, using lightweight structural design to extract features. Experimental data shows that our model is accurate and resilient while reducing computation cost and model size. However, DL's high computational complexity hinders its practical use, particularly in WSN devices with significant processing capabilities due to power limits.

Halbouni et al.[26] developed a hybrid IDS algorithm using the temporal and spatial information mining abilities of LSTM Networks and CNNs. For better model performance, we included batch normalization and dropout layers. It was trained using binary and multiclass classification datasets CIC-IDS 2017, UNSW-NB15, and WSN-DS. The effectiveness of a system is defined by the confusion matrix, which encompasses performance metrics like accuracy, completeness, recall, F1-score, and false alarm rate (FAR). The experimental results presented in this study provide evidence of a maximum detection ratio, favorable accuracy, and a comparatively low incorrect acceptance ratio, thereby confirming the efficacy of the proposed model. Researchers started to depend on DL, nevertheless, through the development of artificial NN and DL algorithms that can produce features automatically without human interaction.

Yao et al.[27] the proposed study focuses on WSNs' susceptibility to assaults and their devices' limited storage capacity. To resolve this problem, the researchers suggest a new approach that associates PCA (Principle Component Analysis) with a DCNN for the purpose of detecting DoS traffic anomalies in WSNs. The lightweight model can recognize network abnormal traffic in WSN devices with restricted store capacity better than the standard deep learning structure. In order to verify the model's classification results, ROC curves, additional classification metrics, and confusion matrices are utilized. THGFSO the suggested model beats other popular anomalous traffic detection algorithms by the classification impact via experimental comparison, even with a modest model size. The completely linked layer, however, includes a lot of characteristics, which greatly raises the difficulty of building models.

Dener et al.[28] to identify DoS assaults targeted at WSNs, a special DoS IDS (DDS) has been suggested. The LightGBM ML procedure, data balancing, and FS (Feature Selection) are used by the ensemble IDS STLGBM-DDS. Google Colab's Apache Spark big data technology powers it. Data imbalance processing utilizing SMOTE and STL was used to reduce system performance. As a further FS method during the data preprocessing step, Information Gain Ratio was used. We looked at how the system's detection performance was affected by the data balancing and feature selection phases. Using the parameters for Accuracy, F-Measure, Precision, Recall, ROC Curve, and Precision-Recall Curve, the findings were assessed. A total accuracy of 99,95 % was attained by the suggested procedure as a result. Furthermore, according to the Normal, Grayhole, Blackhole, TDMA, and Flooding classes, it performed with accuracy rates of 99,99 %, 99,96 %, 99,98 %, 99,92 %, and 99,87 %, respectively. But, ML techniques are insufficient for feature learning, particularly in increasingly intricate nowadays and unpredictable network contexts.

Inference

Related research shows that classification-based assault detection has been the main topic of study in most cases. Many of the offered methods make use of datasets that are out of date and include conventional network-specific data. Additionally, the issue of data imbalance has not been adequately addressed in the majority of research that suggests an intrusion detection system. Additionally, the majority of research has overlooked hyperparameters selection. The IDS for WSNs has been developed using hybrid deep learning techniques in this work. Additionally, the effectiveness of intrusion detection performance was assessed in relation to data

https://doi.org/10.56294/sctconf2024762

balance and feature selection strategies.

METHODOLOGY

The DCNN-BiLSTM's main block diagram is presented in figure 2. The CNN-based network IDS features a four-phase process. First, perform intrusion detection preprocessing on each record in the information gathering then the CNN model may use the record's data type and format as input. The cleaned data are then supplied into the input layer of the DCNN-BiLSTM, to precisely extract the unique information present in each individual record, the convolution layer performs a convolution process. The pooling layer creates additional features by combining the feature points in the immediate neighbourhood. Network training is accelerated by pooling. Maximum and average pooling are often employed. In the last stage, feed data into the Softmax classifier over the FC (Fully Connected) layer to classify the incursion. Hyperparameters tweaking can be done faster and more accurately using metaheuristic methods. As a result, in this study, the DCNN-BiLSTM model's parameter selection is done using the HGFSO method.

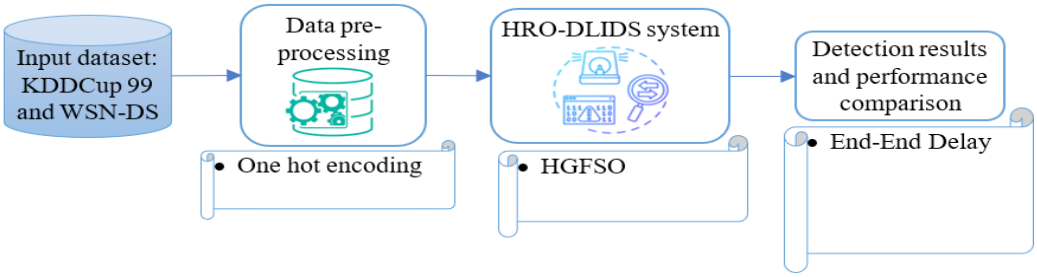

Figure 1. Architecture Diagram of Proposed Methodology

Dataset details

KDDCup 99: this dataset was derived from 1998 DARPA intrusion detection challenge datasets via tcpdump data processing. The MADMAID framework was utilized in order to extract appropriate data from tcpdump data. To compile the KDDCup 1998 dataset, the MIT Lincon laboratory utilized thousands of UNIX workstations and hundreds of users. For ten weeks, packets were intercepted and archived in tcpdump format. The initial seven weeks of data served as the training set, while the remaining information constituted the testing set. The KDDCup 99 dataset is available in two distinct formats. The two options are a complete dataset and a sample dataset. This dataset[29] comprises 41 characteristics and 5 classes ('U2R', 'Normal,' 'Probe,' 'DoS,' 'R2L,').

WSN-DS is a dataset developed for WSNs that is exclusive to IDS. Blackhole, Grayhole, Flooding, and Scheduling are some of these 'DoS' assaults. Using the LEACH (Low-energy Adaptive Clustering Hierarchy) approach, they preprocessed data from Network Simulator 2 (NS-2) and retrieved 23 features.[29]

Dataset Pre-processing

The initial dataset has to be pre-processed using methods like data purification and data transformation for constructing a classification algorithm for ID depends on the incorporation of CNN and BiLSTM. Dirty data is a term for missing, inaccurate, and incomplete information. As a result of violated internal rules in the raw data, data analysis and processing perform poorly. As a result, "dirty data" must be cleaned in order to be transformed into data that meets the needed requirements for data quality. The process of data cleansing is largely impeded by the presence of missing values, erroneous data, mismatches, and distortion. The proposed methodology involves the removal or substitution of existing special symbols and corrupted codes, followed by the use of a consistent constant to populate the invalid data.

One hot encoding

Since most ML and DL algorithms need numerical input, we utilized one hot encoding to convert category features to numeric values. With this method, each value is given a unique index after counting the unique values for each attribute.

Data normalization

It either normalizes feature values in [0,+1] or [1,+1] according to the DL model. Model convergence is accelerated and training time is decreased through data normalization, also known as standardization. Averaging, standard scaling, and minmax scaling are some of the methods for normalizing data. In this study, the data were normalized using a standard scalar. Standard scalar's mean and variance are 0 and 1, respectively, since it employed a standard normal distribution (SND). 37+43 features after one hot encoding; in addition, a standard scalar was utilized to normalize the feature space. Equation 1 may be used to describe it mathematically.

https://doi.org/10.56294/sctconf2024762

$$z = \frac{x-\mu}{\sigma} \qquad (1)$$

Here, the mean, σ standard deviation, and z stand for the standard feature space of the x input data samples. The mean is expressed mathematically as the following: $\mu=(1/N)\sum_{i=1}^{n} x_i$ and standard deviation is:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2} \qquad (2)$$

Here x_i is input sample.

Intrusion Detection using HGFSO with DCNN-BiLSTM

For classification problems, particularly in image recognition, the DCNN method in Algorithm 1 is a SL method. A picture or series of images is provided as input, and the algorithm classifies them according to their characteristics. The method is started by setting the weight settings for each layer. The weights are then adjusted repeatedly across a number of epochs. Convolutional, activation and max-pooling layers are used to remove features from the input image throughout each epoch's forward transit through the network. L1 and L2 regularization then regulate overfitting after processing the flattened feature map with a FC layer with ReLU activation. As the network's last layer, a softmax layer outputs class probabilities for the input image. After calculating loss using projected probabilities and actual labels, weight parameters are modified by backpropagation to decrease loss. Repeat this for each training image. After each epoch, the model is tested on the validation set for accuracy and overfitting. The method checks the algorithm's effectiveness on the test set at the conclusion of each epoch to gauge its overall correctness. The typical DNN algorithm shown in Algorithm 2 was designed to be improved upon by including L1 and L2 regularization approaches. The suggested DCNN Algorithm presented in Algorithm 1 was compared with this improved version in the following study. Each neuron in a layer of the DNN algorithm is linked to every neuron in the layer above it, forming a succession of completely connected layers. This algorithm's primary objective is to find the weight values and biases that will allow for the most accurate categorization of the input data for each layer.

Algorithm 1 DCNN Algorithm

Input: $I = [X1, X2, \ldots Xn]$

Output: Classify Attacks W

1. Initialize weight parameters W for each layer l;
2. for each epoch $e \in [1, E]$ do
3. Shuffle the training set
4. for each training (xi, yi) do
5. Forward propagation
6. Convolution layer: $z[l] = W[l] * a[l-1] + b[l]$
7. Activation layer: $a[l] = g(z[l])$ using ReLU
8. Max-pooling layer: $a[l] = maxpool(a[l-1])$
9. Flatten into a vector $a[L]$
10. Fully connected layer with ReLu as $z[L+1] = W[L+1]*a[L] + b[L+1]$
11. L1 Regularisation $L_1=\lambda\sum_{i=1}^{n}|w_i|//w_i$
12. L2 Regularisation $L_1=\lambda \sum_{j=1}^{p} \beta_{j2} // \beta_j$
13. $a[L+1] = sof\ tmax(z[L+1])$
14. Calculate Loss $Loss=-(1/OS)\sum_{i=21}^{OS} y_i*\log y_i+(1-y_i)*\log(1-y_i)$
15. Update weight parameters
16. end for
17. Test model performance on validation set
18. end for
19. Test model performance on test set

https://doi.org/10.56294/sctconf2024762

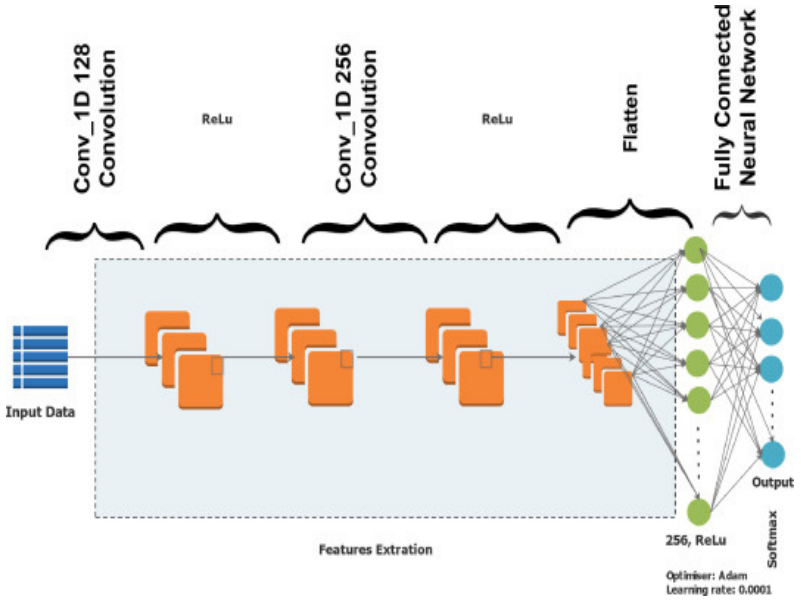

Figure 2. Architecture of DCNN

Figure 2 depicts the suggested method design for the DCNN algorithm, which includes an input layer that has a unit that is comparable to the feature of choice. ReLU activation is used in input layer applications. Accordingly, 128 and 256 filter units are constructed for the CNN layers. In contrast, the FC networks have a heightened level of connectivity, characterized by a larger number of units, namely 256, and a dropout rate of 0,1. These FC networks provide links to the top layer. The softmax activation function is employed in the output layer of the CNN whereas ReLU activation function is employed in each of the other layers. Equation 3 is used to calculate the loss of a sample, and the definite CEL (Cross-Entropy Loss) function is used to calculate the loss.

$$Loss = -\sum_{i=21}^{OS} y_i * log_i \qquad (3)$$

The input layer of the DCNN model shown in figure 2 contains a unit that is comparable to the feature that was selected. The input layer makes use of the relu activation function. In contrast to those with FC networks, which featured a smaller network with just 256 units and a 0,1 dropout rate linking to the last layer, the CNN employed layers with 128 and 256 filter units. Each of the CNN layer's RELU activation functions is used, and the output layer's soft max activation (Equation 4) is used. When determining the loss of a sample by equation 1 for the DCNN, the definite CEL function is generated.

$$SA(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \qquad (4)$$

The DNN approach makes use of the soft max activation function in the output layer to find the derivation of the loss function in relation to each weight and each value in the set being trained. This computation is performed using equation 4. The degree to which the extra hidden layers use the RELU activation function is demonstrated in equation 5.

$$ReLu = \max(0, x) \qquad (5)$$

If the supplied value is positive, the ReLu function returns that value; if not, it returns 0. This makes it easier for the model to understand complicated correlations between the input data and the target variable by adding nonlinearity to it. Adam, an algorithm that uses an adjustable learning rate, is the optimization procedure employed in this algorithm. Adam improves the training process by calculating separate LR (learning rates) for all parameter in the NN, in contrast to conventional gradient descent. By calculating the 1st and 2nd momentum of the gradient, it adjusts the learning rate for each weight. The Deep CNN model that is being suggested includes more substantial hidden layers, which makes computation costly. We may think about utilizing smaller filters in the convolutional layers or lowering the layers count to enhance performance while preserving processing resources. Weight regularization may be used to avoid overfitting and continue to improve the suggested model. Weight regularisation reduces overfitting by incrementing a penalty term to

https://doi.org/10.56294/sctconf2024762

the loss function depends on weight complication algorithm. L1 and L2 weight regularization are popular. L1 regularisation penalizes are proportional to weights' absolute value, whereas L2 penalizes are proportional to weights' square. Weight regularization allows for the development of a more accurate and general mode throughout training.

BiLSTM: Due to their enhanced ability to retain this particular series of data, Bi-LSTM models have recently attracted a lot of attention since both prior, and forthcoming contexts are equally essential.[30] This overcomes the limitations of unidirectional LSTMs and classical RNNs, which could only store previous context and short-term memory, respectively. In order to forecast DDoS assaults based on previous traffic, this research constructed a BiLSTM model, which uses two unique hidden layers. Forward and backward pass LSTM make comprise the two sub-networks that makeup BiLSTM.[30] Given an input sequence of $x_1,x_2,\ldots,x_n$, of "n" features, BiLSTM calculates the subsequent (ahead) hidden vector."(hv) $\vec{}$ " and the preceding (backwards) hidden vector."(hv)$\vec{}$". By integrating the left and right contextual portrayals, the output sequence $h_1,h_2,\ldots,h_t$ It is created as an input to the final layer to predict each data stream: $h^{\leftrightarrow}=[(h^{\leftarrow}\&h^{\rightarrow})]$

SoftMax-Based Prediction Strategy: The final layer uses SoftMax to compute the chance of accurately predicting the target labels after receiving the output of the BiSLTM layer as an input (i.e., the court decisions). The aggregate input was calculated as traffic t, as given in equation (6):

$$\mathbb{t}_i = \phi(\mathbb{x}) \sum w_i \cdot \mathbb{i}_i + b \qquad (6)$$

Where $\phi(x)$ It is the high-dimensional feature space that is nonlinearly transcribed from the input space x, and "w" stands for the weight vector, i for the input vector, "b" for the bias. The calculation for SoftMax Sm is described in equation (7):

$$\mathcal{S}\mathfrak{m}(\mathbb{t}_i) = \frac{e^{\mathbb{t}_i}}{\sum_{n=1}^{m} e^{\mathbb{t}_n}} \qquad (7)$$

This calculation showed that the traffic (normal) had the highest likelihood. Based on the provided historical traffic statistics, the expected attack choice was "A ". The current DCNN-BiLSTM model provides the best results while tolerating low accuracy. Specific enhancements were developed utilizing HGFSO with the intention of solving the shortcomings of the traditional technique.

Hyperparameter Optimization using HGFSO

The HGFSO technique is used in the last step to optimize the hyperparameters of the DCNN-BiLSTM model. By combining GOA and FMSO, the HGFSO algorithm is created. A better position for an efficient result might be determined using this GOA method.[31] Because of the larger number of processing steps, FMSO simultaneously uses a higher level of computational complexity.[32] Use the HGFSO technique with the HGFSO to get an optimum global solution with quick performance and improved computational steps.

Grampus Optimization Algorithm (GOA)

The GOA is a revolutionary population-based stochastic optimization approach that was recently created and is inspired by nature. Search agents help the GOA find the best optimization solution. The GOA simulates humpback whale hunting using "bubble-net hunting". The GOA consists of three main steps: surrounding the prey, using a bubble net to assault, and looking for the best prey.[26] The Grampus uses a bubble net to hunt, and its basic method is to locate its target, build a bubble net along the spiral route, and then proceed upstream to the prey. Prey is surrounded, attacked with a bubble net, and hunted in three steps of this predation behavior. To identify the best method for histogram augmentation, the Grampuses surround their prey, such as fish, and then attempt to update their locations. Equation (8) illustrates the basic mathematical component of the GOA.

$$\mathbb{X}(time+1) = \begin{cases} \mathbb{X}^*(time) - \mathcal{u} \times |cv \times \mathbb{X}^*(time) - \mathbb{X}(time)| & if\ \mathcal{P} < 0.5 \\ |cv \times \mathbb{X}^*(time) - \mathbb{X}(time)| \times e^{\mathcal{b}\cdot\ell} \cdot cos(2\pi time) + \mathbb{X}^*(time) & if\ \mathcal{P} \geq 0.5 \end{cases} \qquad (8)$$

If time is the time or iteration index, X^* is the best solution so far, and X is a vector containing all the grampuses' locations; u=2a.(r-a); cv=2.r; rd is a random vector with values between 0 and 1; and is a coefficient vector that progressively falls from 2 to 0 over the course of repetitions; b is a constant value set to 1 in this work that, depending on the input image, determines the geometry of the logarithmic spiral; l the probability P in equation (8) are 50 % and 50 %, meaning that throughout the optimization process, the grampuses randomly choose either route with an equal chance. Is a random value between - 1 and 1 that is used to switch updating

https://doi.org/10.56294/sctconf2024762

the grampuses' locations. The random value for the bubble-net phase is u is [- 1,1], The random value of vector A during the searching phase, however, might be more or lower than 1. Equation (9) illustrates the search procedure.

$$\mathbb{X}(time+1) = \mathbb{X}_{rd} - u \times |cv \times \mathbb{X}^{*}(time) - \mathbb{X}(time)| \quad (9)$$

The GOA algorithm is enforced to do a global search using this random search mechanism with a value of |u| larger than one, which stresses the search process. Beginning with the GOA searching procedure, random solutions are generated. Then, using the procedure, update these solutions repeatedly. The search will end after a certain number of iterations.

Felis Margarita Swarm Optimization Algorithm (FMSO)

The feeding habits of Felis Margaritas in the desert are used as the basis for the FMSO algorithm. In order to seek prey above or below ground, the Felis Margarita can sense low-frequency sounds. Position updates let the search agent identify movies around the optimal value. In exploration space, the algorithm perceives the ideal value as prey. Prey search and assault mechanisms are included in the FMSO algorithm. Felis Margaritas' hunt for prey may be mimicked using the search prey mechanism. The population of Felis Margaritas has a search-prey equation that is as follows:

$$\vec{\mathbb{X}}(recitr+1) = \vec{\mathfrak{sr}} \cdot \left(\vec{\mathbb{X}}_{bp}(recitr) - rd(0,1) \cdot \vec{\mathbb{X}}_{cl}(recitr)\right) \quad (10)$$

$\vec{X}$ denotes the search agent, recitr reflects recent iterations, $\vec{X}_{bp}$ indicates the best candidate position, $\vec{X}_{cl}$ indicates the current location, rd represents a random number, and $\vec{(sr)}$ describes Felis Margaritas' sensitivity to low-frequency noise.

$$\vec{\mathfrak{sr}} = \vec{\mathfrak{sr}}_g \times rd(0,1) \quad (11)$$

As the $\vec{(sr)}_g$ decreases linearly from 2 to 0, the overall sensitivity range may be represented as:

$$\vec{\mathfrak{sr}}_g = \frac{sc_f \times recitr}{itr_m} \quad (12)$$

The current iteration is recitr, and the maximum iteration is itr_m. Further, the Felis Margaritas detects low frequencies sc_f of 2khz. After searching for prey, FMSO algorithm assaults prey, and Felis Margaritas population prey attack method is:

$$\vec{\mathbb{X}}_{rd} = \left|rd(0,1) \cdot \vec{\mathbb{X}}_{bp}(recitr) - \vec{\mathbb{X}}_{cl}(recitr)\right| \quad (13)$$

$$\vec{\mathbb{X}}(recitr+1) = \vec{\mathbb{X}}_{bp}(recitr) - \vec{\mathfrak{sr}} \cdot \vec{\mathbb{X}}_{rd} \cdot \cos\theta \quad (14)$$

Where θ is an arbitrary angle range from 0 and 360 degrees, hence cos θ has values form -1 to 1. When the best position and the present position are combined, $\vec{X}_{rd}$ shows the random position that results. The population may migrate in several circular directions with this technique. The angles are picked at random by each Felis Margarita. Felis Margaritas will be able to dodge any nearby ideal traps as she moves closer to the prey. The agent's direction of search and hunt is influenced by the random angle in equation (13).

https://doi.org/10.56294/sctconf2024762

1. Define the objective function F(X) of the problem
2. Set values for parameters of GOA and FMSO
3. Generate a block of storage and retrieval requests
4. Set the parameters of GOA and FMSO
5. Initialize the positions $\mathbb{X}_i$ of Grampuses
6. Initialize the positions $\vec{\mathbb{X}}_j$ of Felis Margaritas
7. Evaluate the F($\mathbb{X}_i$) for each Grampus i; Find the $\rightarrow \mathbb{X}_w^* \ (time)$
8. Evaluate the F($\mathbb{X}_j$) for each Felis Margarita j;
9. Set $itr = 1$ // itr is iteration counter
10. REPEAT1 until (t = T)
11. Call FMSO () and find the best $\rightarrow \mathbb{X}_{sr}^* (time)$; // call FMSO
12. If $\rightarrow |AIT| > 1$ then update the $\rightarrow \vec{\mathbb{X}} (recitr + 1)$
13. Set t= 1
14. Repeat 2 until (rit= $recitr$)
15. Update parameters $\vec{\mathrm{sr}}_g$, $\vec{\mathbb{X}}_{rd}$ and $\vec{\mathbb{X}} (recitr + 1)$
16. IF ($\mathcal{P} < 0.5$) // shrink encircling movement
17. IF ($|AIT| \leq 1$) // Exploitation
18. Update the $\vec{\mathbb{X}} (recitr + 1)$with Eq. (14)
19. Else // Exploration
20. Select a random $\vec{\mathbb{X}}_{rd}$ from the Grampus group
21. Update the $\mathbb{X}(recitr + 1)$with Eq. (10)
22. End If
23. Else $\vec{\mathbb{X}} (recitr + 1)$
24. Update the $\mathbb{X}_i$ with Eq. (12) and Eq. (14)
25. End If
26. $recitr = recitr + 1$
27. End Repeat 2
28. Check and amend the Grampuses out of the search space
29. Evaluate $fitness$ for each Grampus i as accuracy Eq.(18).
30. Update the global best $\rightarrow \vec{\mathbb{X}} (recitr + 1)$
31. $itr = itr + 1$
32. End Repeat 1
33. Output hyperparameters of optimal solution of DCNN-BiLSTM

Figure 3. The Pseudocode of GOA-FMSO Based Contrast Enhancement

Exploration and Exploitation process: An adaptive impact technique $(AIT)^{\rightarrow}$ used by FMSO to balance the exploration and exploitation stages is as follows:

$$\overrightarrow{AIT} = 2 \times \vec{\mathrm{sr}}_g \times rd(0,1) - \vec{\mathrm{sr}}_g \qquad (15)$$

Where, $(sr)^{\rightarrow}_g$ as the quantity of iterations is increased, drops linearly from 2 to 0. During the exploration and exploitation phase, the updated position of each Felis Margarita is as follows:

$$\vec{\mathbb{X}} (recitr + 1) = \begin{cases} \vec{\mathrm{sr}} \cdot \left(\vec{\mathbb{X}}_{bp}(recitr) - rd(0,1) \times \vec{\mathbb{X}}_{cl}(recitr) \right) & |AIT| > 1 \\ \vec{\mathbb{X}}_{bp}(recitr) - \vec{\mathrm{sr}} \cdot \vec{\mathbb{X}}_{rd} & |AIT| \leq 1 \end{cases} \qquad (16)$$

Where |AIT|≤1y initiates an assault by the FMSO search agent on the target prey, otherwise, the search agent looks for potential solutions broadly. To prevent the algorithm from settling on local optimum solutions, each Felis Margarita has a unique search radius throughout the exploration phase.

Hybrid DCNN-BiLSTM: in order to achieve the required outcomes, the DCNN-BiLSTM architecture uses the DCNN layer for FE (Feature Extraction) on input data,[29] BiLSTMs for sequence prediction, and the DCNN to boost optimizing the error and loss.[30] The CNN BiLSTM architecture-based hybrid DCNN and BiLSTM model was originally known as the Long-term Recurrent Convolutional Network, or LRCN model. Initially, it was referred to as the DCNN BiLSTM architecture. By initially adding CNN layers to the front end, then BiLSTM levels, a DCNN layer, and then the output layer, it is feasible to construct a Deep CNN-BiLSTM.

RESULTS AND DISCUSSION

DoS, Probe, R2L, and U2R are the four kinds of threats that may be made against the NSL-KDD dataset. A total

of 37 different attack types are also conceivable for each of these categories. Due to this, we have gathered information from the dataset that relates to DDoS assaults (WSN-DS), which includes Back, Land, Neptune, Pod; Smurf; Mailbomb; Processtable; Udpstorm; Apache2; and Worm. The extracted datasets for KDDTrain and KDDDTest include 148517 records and 43 feature columns. The tests were conducted on a computer running Windows 10 64-bit with 16 GB of RAM and an Intel(R) Core-i7 processor. A cloud server with several virtual machines was built using VMware Workstation. How accurately the detecting system can categorize incoming signals determines how effective it is. Comparative studies were conducted between the suggested strategy HGFSO-DLIDS and COA-GS-IDNN and popular techniques as AdaBoost-RBFSVM, GWO LSTM and DNN.[6,7,16] The following describes these actions.

Recall: based on successful instances in the database, estimates the number of useful class predictions.

$$Recall = \frac{TP}{TP+FN} \qquad (17)$$

Precision: determines the proportion of accurate positive class predictions.

$$Precision = \frac{TP}{TP+FP} \qquad (18)$$

F-measure: generates a single score while taking accuracy and recall problems into consideration:

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \qquad (19)$$

Accuracy: It is a proportion between the total number of samples and those that were properly categorized.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (20)$$

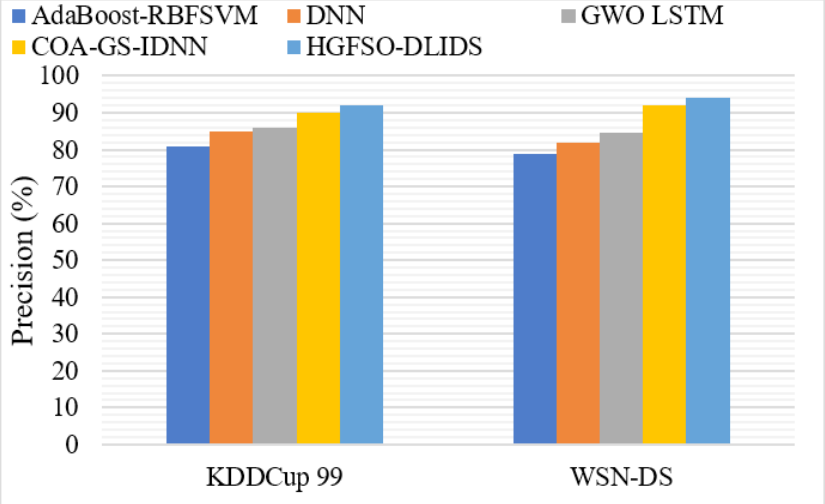

Figure 5. Precision performance comparison

The precision comparison results between suggested COA-GS-IDNN, HGFSO-DLIDS and conventional AdaBoost-RBFSVM, DNN, GWO LSTM and COA-GS-IDNN classifiers are presented in figure 5. When compared to current approaches, the suggested method, as shown by the graph, has a high accuracy rate. When evaluating the accuracy of current methods, AdaBoost-RBFSVM, DNN, GWO LSTM and COA-GS-IDNN, HGFSO-DLIDS provide good precision rates of 81 %, 85 %, 86 %, 90 %, 92 % respectively for KDDCup, attained 79 %, 82 %, 84,5 %, 92 %, 94 % for WSN-IDS. The capacity and size of the dataset, the FS, the LR and activation, as well as other factors, all affect how well the HGFSO-DLIDS algorithm performs. In developing a classifier depends on DL, which is still an exciting area of research, it is illogical to adopt that a given algorithm is acceptable for all datasets. Since DCNN-BiLSTM and HGFSO's assertiveness rates were greater than those of other metrics, the suggestion to apply this method was validated.

https://doi.org/10.56294/sctconf2024762

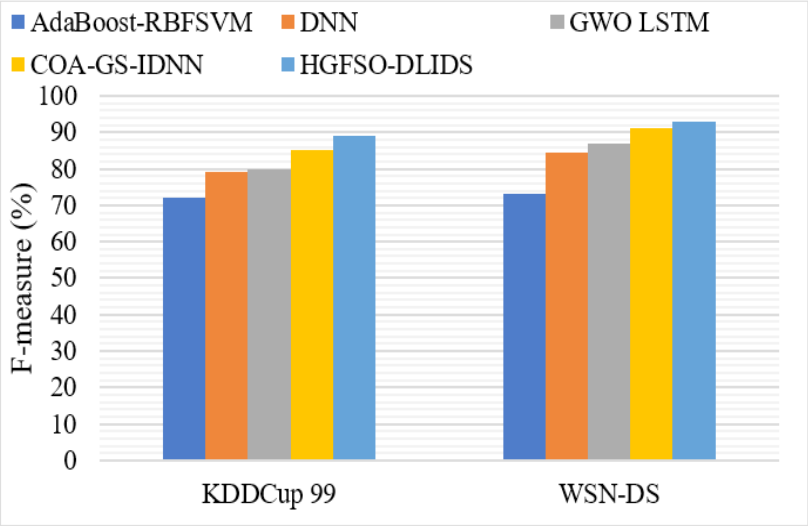

Figure 6. F-measure performance comparison

Figure 6 illustrates the outcomes of the F-measure comparison of the recommended classifiers for AdaBoost-RBFSVM, DNN, GWO LSTM, COA-GS-IDNN, and HGFSO-DLIDS. When contrasting the F-measure ratio among the current methods, AdaBoost-RBFSVM, DNN, GWO LSTM and COA-GS-IDNN, HGFSO-DLIDS provide lower rates of 72 %, 79 %, 80 %, 854 %, 89 % respectively for KDDCup 99, for WSN-DS dataset attained 73,15 %, 84,5 %, 87 %, 91 %, 93 % respectively, proving that the recommended plan can provide better results for attack detection than the earlier methods. A fully connected DNN is coupled to the BiLSTM algorithm, is the study's main contribution. They were coupled because BiLSTM can learn bidirectional long-term and short-term dependencies, increase model resilience, and handle the sequence of input depending on the structure. The combinational model beats all conventional current IDS models, according to the experiment results for the suggested strategy.

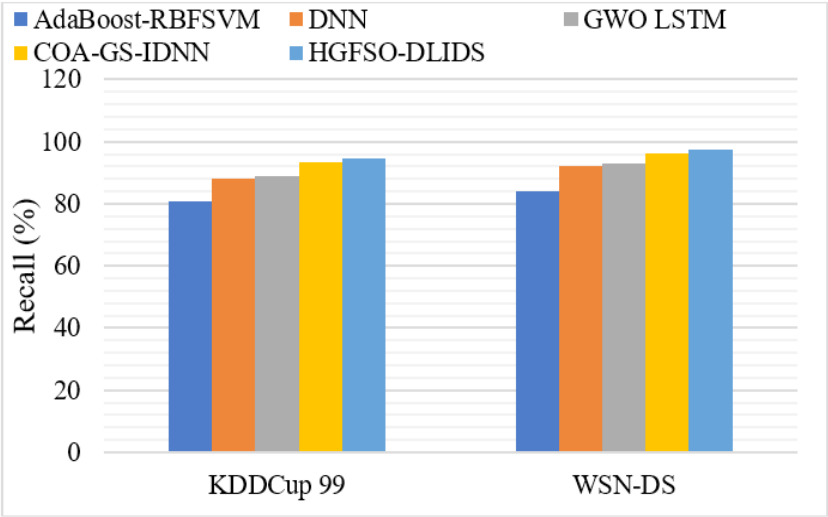

Figure 7. Recall performance comparison

Figure 7 displays the recall comparison results for the classifiers COA-GS-IDNN, HGFSO-DLIDS, DNN, GWO LSTM, and AdaBoost-RBFSVM that have been recommended. The suggested method delivers a very high recall rate. The recommended COA-GS-IDNN, HGFSO-DLIDS has a high recall rate value, demonstrating a strong attack detection proportion, according to the data. In comparing the recall rates of the current methods, AdaBoost-RBFSVM, DNN, GWO LSTM and COA-GS-IDNN, HGFSO-DLIDS provide recall rates of 81 %, 88 %, 88,95 %,93,2 %, 94,5 % respectively for KDDCup 99, for WSN-DS dataset attained 84 %, 92 %, 93 %, 96,31 %, 97,52 % respectively, proving that the recommended plan can provide better attack identification results than the earlier methods. The main benefits of the HGFSO-DLIDS approach are the robustness in the achievement when providing a huge amount of data.

https://doi.org/10.56294/sctconf2024762

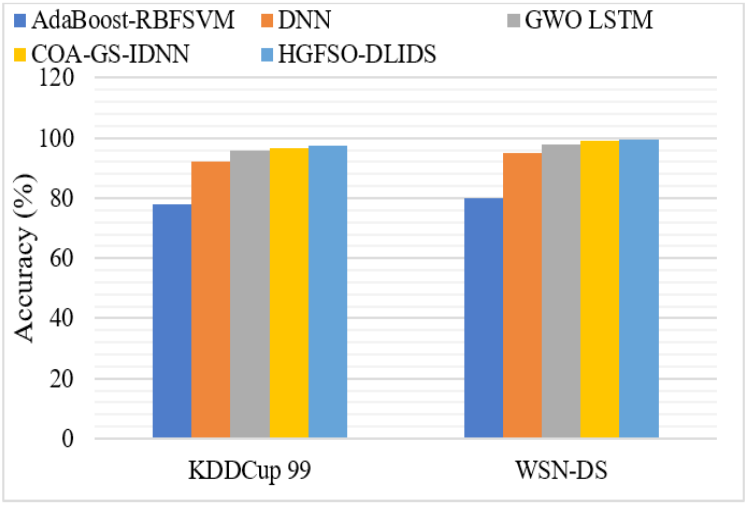

Figure 8. Accuracy performance comparison

The accuracy comparison for attack detection is shown in the graph in figure 8 above. Methods such as AdaBoost-RBFSVM, DNN, GWO LSTM and COA-GS-IDNN, HGFSO-DLIDS multiclass classifiers are used. Obtaining precise forecasts with a high accuracy rate is made possible by the COA-GS-IDNN, HGFSO-DLIDS approach, which is outstanding. In evaluating the efficacy of earlier methods such AdaBoost-RBFSVM, DNN, GWO LSTM and COA-GS-IDNN, HGFSO-DLIDS the rates are as follows: of 78 %, 92 %, 96 %, 96,55 %, 97,5 % respectively for KDDCup 99, for WSN-DS dataset attained 80 %, 95 %, 98 %, 99,05 %, 99,56 respectively. The result is huge, particularly in a situation when it's crucial to see assaults soon away in order to minimize damage. Because the HGFSO has a higher generalization performance and a quicker learning speed for building the model via batch normalization, which speeds up the deep network training, the time complexity while employing the HGFSO activation function was low when compared to the other techniques.

CONCLUSION

This study introduces a unique hyperparameter tuning technique combining HGFSO and a DCNN with BiLSTM for IDS. The suggested method improves the security of computer networks by providing a practical and trustworthy technique for identifying and categorizing network intrusions. The suggested framework increases the accuracy and efficiency of detecting network intrusions by making use of DCNN-BiLSTM technology. The suggested framework's usefulness in correctly recognizing and categorizing different sorts of intrusions is shown by the performance assessment. The evaluation's findings show that modern approaches can recognize targets more accurately than older ones, with higher detection rates and fewer false positives. The suggested approach regularly outperforms several other models in the table, achieving excellent accuracy across a variety of datasets. It may be concluded from this that the model is successful in correctly recognizing network breaches. The suggested technique has reasonably quick training duration, showing that it can be effectively trained on big datasets. The suggested method's efficient inference time allows for speedy predictions on fresh, unforeseen data. It may be challenging to install DCNN-BiLSTM-based IDS on high-speed networks due to the computing resources required to handle enormous volumes of data in real-time. Although HGFSO-DLIDSs have promising IDS capabilities, there are several restrictions and issues that need to be resolved. When compared to a balanced dataset, the model's loss rate is more likely to be unbalanced. Despite the aforementioned drawback, DCNN-BiLSTM permits end-to-end learning, which means they do so without the need for manually created feature engineering and instead learn from the raw input data. As a result, the hard and time-consuming process of creating particular characteristics for ID is reduced by doing away with the necessity for manual feature extraction. Future research will focus on developing hybrid metaheuristics-ensemble DL approach is for enhancing ID in WSN systems.

REFERENCES

1. Marriwala N, and Rathee P. An approach to increase the wireless sensor network lifetime. In World congress on information and communication technologies, pp. 495-499. https://doi.org/10.1109/WICT.2012.6409128.

2. Gungor VC, Lu B, and Hancke GP. Opportunities and challenges of wireless sensor networks in smart grid. IEEE transactions on industrial electronics, 57(10), pp. 3557-3564. https://doi.org/10.1109/TIE.2009.2039455.

3. Rassam MA, Maarof MA, and Zainal A. A survey of intrusion detection schemes in wireless sensor networks.

American Journal of Applied Sciences, 9(10), pp. 1636-1652.

4. Butun I, Morgera SD, and Sankar R. A survey of intrusion detection systems in wireless sensor networks. IEEE communications surveys & tutorials, 16(1), pp. 266-282. https://doi.org/10.1109/SURV.2013.050113.00191.

5. Modares H, Salleh R, and Moravejosharieh A. Overview of security issues in wireless sensor networks. In third international conference on computational intelligence, modelling & simulation, pp. 308-311. https://doi.org/10.1109/CIMSim.2011.62.

6. Sen, J. Security in wireless sensor networks. Wireless sensor networks: current status and future trends, 407, pp. 1-51.

7. Farooq N, Zahoor I, Mandal S, and Gulzar T. Systematic analysis of DoS attacks in wireless sensor networks with wormhole injection. International Journal of Information and Computation Technology, 4(2), pp. 173-182.

8. Ghosal A, and Halder S. Intrusion detection in wireless sensor networks: Issues, challenges and approaches. Wireless Networks and Security: Issues, Challenges and Research Trends, pp. 329-367. https://doi.org/10.1007/978-3-642-36169-2_10.

9. Apruzzese G, Colajanni M, Ferretti L, Guido A, and Marchetti M. On the effectiveness of machine and deep learning for cyber security. In 2018 10th international conference on cyber Conflict (CyCon), pp. 371-390. https://doi.org/10.23919/CYCON.2018.8405026.

10. Xin Y, Kong L, Liu Z, Chen Y, Li Y, Zhu H, Gao M, Hou H, and Wang C. Machine learning and deep learning methods for cybersecurity. Ieee access, 6, pp. 35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950.

11. Milosevic N, Dehghantanha A, and Choo KKR. Machine learning aided Android malware classification. Computers & Electrical Engineering, 61, pp. 266-274. https://doi.org/10.1016/j.compeleceng.2017.02.013.

12. Aslan ÖA, and Samet R. A comprehensive review on malware detection approaches. IEEE access, 8, pp. 6249-6271. https://doi.org/10.1109/ACCESS.2019.2963724.

13. Ye Y, Li T, Adjeroh D, and Iyengar SS. A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3), pp. 1-40. https://doi.org/10.1145/3073559.

14. Kim S, Park KJ, and Lu C. A survey on network security for cyber–physical systems: From threats to resilient design. IEEE Communications Surveys & Tutorials, 24(3), pp. 1534-1573. https://doi.org/10.1109/COMST.2022.3187531.

15. Jiang J, Han G, Wang H, and Guizani M. A survey on location privacy protection in wireless sensor networks. Journal of Network and Computer Applications, 125, pp. 93-114. https://doi.org/10.1016/j.jnca.2018.10.008.

16. Kumar DP, Amgoth T, and Annavarapu CSR. Machine learning algorithms for wireless sensor networks: A survey. Information Fusion, 49, pp. 1-25. https://doi.org/10.1016/j.inffus.2018.09.013.

17. Dwivedi RK, Rai AK, and Kumar R. A study on machine learning based anomaly detection approaches in wireless sensor network. In 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 194-199. https://doi.org/10.1109/Confluence47617.2020.9058311.

18. Salmi S, and Oughdir L. Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network. Journal of Big Data, 10(1), pp. 1-25. https://doi.org/10.1186/s40537-023-00692-w.

19. Sharma HS, Sarkar A, and Singh MM. An efficient deep learning-based solution for network intrusion detection in wireless sensor network. International Journal of System Assurance Engineering and Management, 14(6), pp. 2423-2446. https://doi.org/10.1007/s13198-023-02090-0.

20. Gowdhaman V, and Dhanapal R. An intrusion detection system for wireless sensor networks using deep neural network. Soft Computing, 26(23), pp. 13059-13067. https://doi.org/10.1007/s00500-021-06473-y.

21. Abhale AB, and Manivannan SS. Supervised machine learning classification algorithmic approach for finding anomaly type of intrusion detection in wireless sensor network. Optical Memory and Neural Networks, 29(3), pp. 244-256. https://doi.org/10.3103/S1060992X20030029.

22. Sood T, Prakash S, Sharma S, Singh A, and Choubey H. Intrusion detection system in wireless sensor network using conditional generative adversarial network. Wireless Personal Communications, 126(1), pp. 911-931. https://doi.org/10.1007/s11277-022-09776-x.

23. Biswas P, Samanta T, and Sanyal J. Intrusion detection using graph neural network and Lyapunov optimization in wireless sensor network. Multimedia Tools and Applications, 82(9), pp. 14123-14134. https://doi.org/10.1007/s11042-022-13992-9.

24. Otoum S, Kantarci B, and Mouftah HT. On the feasibility of deep learning in sensor network intrusion detection. IEEE networking letters, 1(2), pp. 68-71. https://doi.org/10.1109/LNET.2019.2901792.

25. Zhao R, Yin J, Xue Z, Gui G, Adebisi B, Ohtsuki T, Gacanin H, and Sari H. An efficient intrusion detection method based on dynamic autoencoder. IEEE Wireless Communications Letters, 10(8), pp. 1707-1711. https://doi.org/10.1109/LWC.2021.3077946.

26. Halbouni A, Gunawan TS, Habaebi MH, Halbouni M, Kartiwi M, and Ahmad R. CNN-LSTM: hybrid deep neural network for network intrusion detection system. IEEE Access, 10, pp. 99837-99849. https://doi.org/10.1109/ACCESS.2022.3206425.

27. Yao C, Yang Y, Yin K, and Yang J. Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network. IEEE Access, 10, pp. 103136-103149. https://doi.org/10.1109/ACCESS.2022.3210189.

28. Dener M, Al S, and Orman A. Stlgbm-dds: An efficient data balanced dos detection system for wireless sensor networks on big data environment. IEEE Access, 10, pp. 92931-92945. https://doi.org/10.1109/ACCESS.2022.3202807.

29. Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, and Venkatraman S. Deep learning approach for intelligent intrusion detection system. Ieee Access, 7, pp. 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334.

30. Jin J. Intrusion detection algorithm and simulation of wireless sensor network under Internet environment. Journal of Sensors, pp. 1-10. https://doi.org/10.1155/2021/9089370.

31. Kaveh A, and Farhoudi N. A new optimization method: Dolphin echolocation. Advances in Engineering Software, 59, pp. 53-70. https://doi.org/10.1016/j.advengsoft.2013.03.004.

32. Seyyedabbasi A, and Kiani F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. Engineering with Computers, 39(4), pp. 2627-2651. https://doi.org/10.1007/s00366-022-01604-x.

FINANCING

"The authors did not receive financing for the development of this research".

CONFLICT OF INTEREST

"The authors declare that there is no conflict of interest".

AUTHORSHIP CONTRIBUTION

Conceptualization: K Sedhuramalingam.
Data curation: N Saravana Kumar.
Formal analysis: N Saravana Kumar.
Research: K Sedhuramalingam.
Methodology: K Sedhuramalingam.
Drafting - original draft: N Saravana Kumar.
Writing - proofreading and editing: K Sedhuramalingam.