International Journal of Management, IT & Engineering
Vol. 14 Issue 04, April 2024
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

A GREY WOLF OPTIMIZATION-BASED FEED-FORWARD NEURAL NETWORK FOR DETECTING INTRUSIONS IN INDUSTRIAL IOT

Ajay Chandra MK
Technical ERP Program Manager

ABSTRACT

The rapid development of the Internet of Things (IoT) system can be attributed to the industrial sector's utilization of cloud-based technology and the Internet. Due to the growing volume of data and variety of devices, IoT technology utilized in the business has evolved into a large-scale network. Networks used for industrial IoT (IIoT) are inherently vulnerable to intrusions and cyberattacks. Thus, the development of Intrusion Detection Systems (IDS) is crucial to guaranteeing the security of IIoT networks. Create a Feed Forward Neural Network with Golden Eagle Optimization (FFNN-GEO) to defend the IIoT system from threats. The created model's primary goal is to improve the IIoT system's security by effectively identifying assaults. To reduce the dimensionality and eliminate noise, min-max normalization and Improved Principal Component Analysis (IPCA) are used. N-gram is used to extract pertinent features, while Correlation Feature Selection and Spider Monkey Optimization (CFS-SMO) are used to choose significant features. Ultimately, the developed FFNN-GEO model more accurately identified and detected the assaults that exist in the IIoT ecosystem. The created technique's experimental findings are verified in terms of accuracy, precision, execution time, and error rate against other methods. The generated model's reliability in identifying attacks is demonstrated by the suggested technique's 98.56% accuracy and 0.046% error rate.

Keywords: Attack Detection, Feed Forward Neural Network, Internet of Things, Industrial IoT, Hybrid Optimization, Security

1. INTRODUCTION

Communication and information technologies have advanced significantly with the Internet of Things (IoT) [1]. As a result, technology has been applied to create automated, intelligent, sustainable, and affordable solutions in several essential industries [2]. The Industrial Internet of Things (IIoT) is the outcome of the widespread IoT integration into the industrial and manufacturing sectors [3]. The activities of businesses and organizations run more reliably and efficiently using the IIoT. To improve industrial and manufacturing procedures, the IIoT is made up of actuators, sensors, and connected devices that interact with each other [4, 5]. It is intended to improve manufacturing sectors' operations by involving embedded systems. IIoT is essential for Industry 4.0 because it enables real-time decision-making for industrial infrastructure facilities and devices and sets the stage for the change of manufacturing processes and cyber-physical networks through the use of big data [6, 7].Yet, the accessibility and interconnectedness of these systems in smart manufacturing units make them vulnerable to attack and misuse by hostile parties, emphasizing the need for cyber security [8]. IIoT networks are intrinsically susceptible to intrusions and hacker attacks.

Even though the industrial sector has benefited greatly from the incorporation of IoT technologies, connected critical areas lack adequate security and privacy protections [9]. Security flaws including open ports, shoddy authentication procedures, and out-of-date software all contribute to the rise of dangers [10]. More possible cyber security threats are brought about by the combination of these elements and the network's direct internet connection.While IIoT offers several advantages to service providers and end users alike, security and privacy continue to be major obstacles [11, 12]. Cybercrime assaults put commercial IoT applications and industrial processes at risk by rendering the service inaccessible [13]. Cyberattacks cause major operational and financial damages and continue to be a major security and privacy problem in intelligent devices [14].Therefore, it's crucial to develop IDS to ensure the security of IIoT networks.

To reduce cyberattacks in a network, an IDS has been used. One of the main influences on creating an efficient IDS is its heterogeneous nature [15]. Traffic feature analysis and anonymous activity identification are difficult tasks for IDS. Current IDS are unable to detect zero-day vulnerabilities because ofa lack of sufficient feature mapping techniques. Nevertheless, fresh and undiscovered cyberattacks are not detected by this kind of IDS [16]. Techniques of deep learning in cybersecurity to identify and counteract various kinds of cyberattacks have shown outstanding cyberattack classification performance [17]. While numerous relevant efforts have addressed cyberattacks in such a networked environment using ML-based intrusion detection systems, certain drawbacks have been noted such as security issues, noise, high execution time, and less scalability [18]. With the increasing variety and sophistication of cyberattacks, the existing anomaly detection approaches in IIoT are becoming less effective [19]. Unfortunately, a lot of IIoT-based IDS solutions are currently developed but suffer from excessive feature dimensions, outdated datasets, imbalanced datasets, and a lack of comprehensiveness in attack types, which makes some of the earlier detection systems ineffective [20]. Furthermore, feature selection techniques have a significant impact on the IDS's efficacy yet are rarely employed in related studies.

To enhance the security of the IIoT system from attacks, design a Feed Forward Neural Network with Golden Eagle Optimization (FFNN-GEO). The main aim of the developed model is to enhance security and detect attacks in the IIoT environment using UNSW-NB15 dataset. The main key contribution of the developed model is detailed below,

- Design an effective optimization-based DL model for identifying and predicting attacks in the IIoT environment.
- The IPCA model is employed to enhance the data quality and overcome the dimensionality issue
- Correlation Feature Selection and Spider monkey optimization (CFS-SMO) are used to enhance the performance of the feature selection process
- GEO is employed to optimize the FFNN model, which enhances attack detection results.

The present study is structured as follows: section 2 provides an overview of related studies, section 3 addresses the problem statement, section 4 presents a proposed methodology for attack detection in anIIoT, section 5 presents results and discussion, and section 6 concludes with a conclusion.

2. RELATED WORKS

An intelligent detection method was created by Sahar and colleagues [21] to detect cyberattacks on IIoT systems. The suggested model reduces data characteristics and enhances detection performance by utilizing the Single Value Decomposition (SVD) technique. Moreover, the Synthetic Minority Over-Sampling (SMOTE) method is used to prevent biased classification caused by over- and under-fitting problems. Data analysis for single- and multi-classification has been accomplished through the application of several ML and DL methods. A lower mistake rate and higher accuracy rate were attained with the suggested strategy.

For IIoT security, Yousef et al. [22] create an IDS model that takes advantage of feature engineering and ML. To greatly reduce calculation costs and prediction times, the created model integrates Pearson's Correlation Coefficient (PCC) with Isolation Forest (IF). To find and eliminate outliers from datasets, IF is utilized. To determine which features are most suited, the designed model used PCC. Implementing the Random Forest (RF) classifier improves IDS performance. The results demonstrated better performance superior to analogous models.

To detect intrusions in the IIoT network, Hakan and colleagues [23] created three unique models using DL architectures, such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). Including both datasets, the hybrid CNN+LSTM approach performed better than the other suggested schemes in terms of malware detection accuracy. The accuracy of the proposed CNN+LSTM system was 93.21%.

Abbas [24] suggests utilizing the Auto Encoder (AE) framework and LSTM to create an ensemble DL model for cyber threat detection in the IIoT that can identify unusual behavior. To decrease the data dimension, AE selects the important data attributes, and the LSTM is used to construct a replica of a normal time series of data to understand normal data patterns. Higher accuracy performance is obtained when the results are compared to standard ML classifiers.

Friha [25] suggested a decentralized, Differentially Private (DP) Federated Learning (FL)-based Intrusion Detection System (2DF-IDS) to protect intelligent manufacturing plants. The DP gradient exchange system, the decentralized FL approach, and the key exchange protocol make up the three building pieces of the proposed 2DF-IDS. In terms of accuracy, the suggested system beats the FL-based method (93.91 percent) and achieves equivalent performance (94.37 percent) with the centralized learning technique.

Without exchanging data, Amir [26] suggests using an ensemble-based deep FL cyber-threat tracking system to seek down assault samples. Two parallel federated-based components compensate for the suggested hunting paradigm; one examines the IIoT status based on the network's typical conditions, while the other analyses it while taking the danger scenario into account. Evaluations further reveal that the suggested model behaves steadily when dealing with varying client counts and that it trains more quickly than centralized models.

DL model anomaly detection in IIoT was created by Jayalaxmi, et al [27] (PIGNUS). Cascade Forward Back Propagation Neural Network (CFBPNN) is integrated with Auto Encoders (AE) for attack detection and classification, while AE is used to choose the best features. The cascade approach creates an ideal classification by identifying normal and abnormal behavior patterns through interwoven relationships from the starting layer to the output layer. According to the findings, the designed technique offers over 95% accuracy, which is approximately 25% better than the current models.

In an IoT-enabled industrial digital world, Sumit et al. [28] suggested a malware detection method using an ML technique called MADP-IIME. The proposed MADP-IIME successfully allows the detection of malware attacks using four distinct types of ML methods: Random Forest (RF), naive Bayes, Artificial Neural Networks (ANN), and logistic regression. Moreover, MADP-IIME attains a 99.5 percent detection and 0.5 percent false positive rate, outperforming other comparable existing schemes.

3. PROBLEM DEFINITION

The possibility of cyber threats and attacks on important systems increases as industries rapidly combine smart and connected equipment via IIoT to improve efficiency and automation. It is crucial to make sure IIoT systems are secure to avoid potential harm, illegal access, and disruptions [29]. Creating intelligent and efficient systems to identify and mitigate IIoT attacks is the problem. IIoT is susceptible to several attack vectors, including replay, malware injection, man-in-the-middle, confidential information leakage, and alteration of sensitive data. As a result, it's critical to stop such an environment from communicating to defend against various attack vectors [30]. Attacks, overfitting, imbalanced datasets, poor accuracy, misprediction, and lengthy execution times are the most difficult problems facing IIoT systems. These days, researchers are paying attention to botnet attacks, often known as malware attacks industrial multimedia infrastructure with IoT capabilities experience communication disruptions due to such attacks [31]. Furthermore, hackers can alter the functions of smart gadgets and take remote control of them. To identify the existence of malware attacks in the IIoT environment, certain strong procedures are necessary.

4. PROPOSED METHODOLOGY

To enhance the security of the IIoT system from attacks, design a Feed Forward Neural Network with Golden Eagle Optimization (FFNN-GEO). The main aim of the developed model is to enhance security and detect attacks in the IIoT environment using UNSW-NB15 dataset. Initially, datasets are collected and trained in the Python system. Moreover, data preprocessing techniques are employed to remove the noise and minimize the dimensionality using IPCA and min-max normalization. The relevant features are extracted using N-gram and important features are selected using Correlation Feature Selection and Spider monkey optimization (CFS-SMO). Finally, the developed FFNN-GEO model identified and detected the attacks present in the IIoT environment with better results. The architecture of the developed model is shown in Fig. 1.

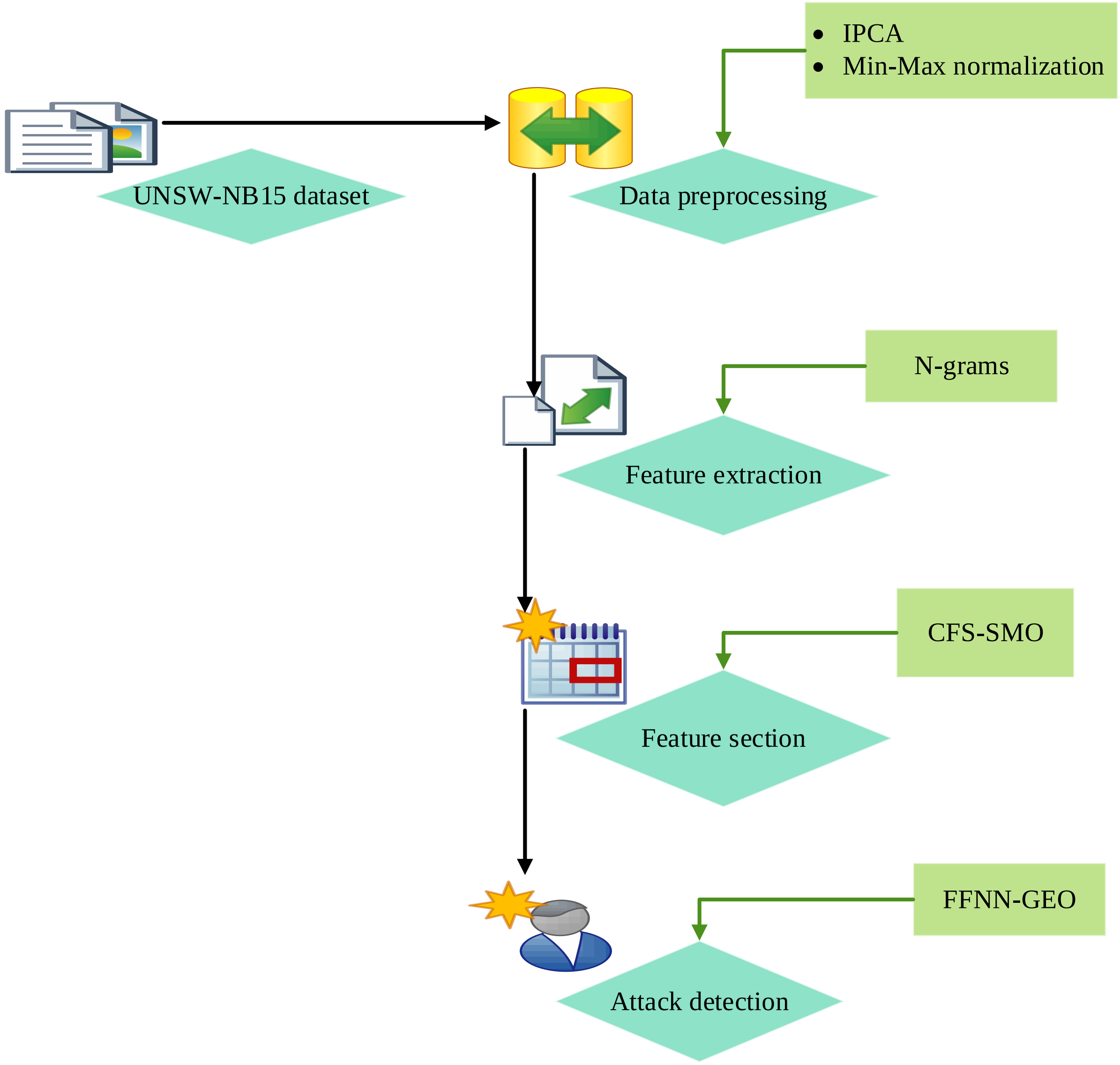

Fig.1 Proposed methodology

4.1 Dataset description

Raw network packets produced by the IXIA PerfectStorm software in the Australian Centre for Cyber Security's cyber range lab are included in the UNSW-NB15 dataset [32]. Within four distinct CSV files, the dataset comprises two million, 540,044 instances, 45 input features, and nine different types of cyberattacks. The experiment considers a division of the dataset with 82,332 testing cases and 175,341 training occurrences with multi-class assault variations. There are 119,341 entries with nine attack types and 56,000 normal values in the collection. 33,393 records total for exploits, and 40,000 records for the generic attack type. Table 1 provides the UNSW-NB15 dataset's comprehensive attack instance.

Table 1 UNSW-NB 15 dataset description

Category	Samples
Generic	40000
Worms	130
Analysis	2000
Backdoors	1746
Exploits	33393
Shellcode	1133
Dos	12264
Fuzzers	18184
Reconnaissance	10491
Normal	560000

Numerous attack classes are available in the training dataset, however, new attack classes that weren't initialized in the training dataset are available in the validation dataset. Attack values are indicated as 1, while normal values are represented as 0. Thirty percent of the dataset was utilized during testing, and seventy percent was used during training. Through the use of data preprocessing techniques, the noise and errors in the gathered dataset are removed. Also, enhance the prediction results by minimizing the dimensionality of the dataset.

4.2 Data preprocessing

The data preprocessing step is essential to reduce the dimension and enhance the data quality. Also,reduces noise and increases the prediction results. Moreover, the IPCA method [33] is employed to remove noise, and error present in the dataset and reduce dimensionality. Then the dataset is normalized using min-max normalization [34].

- **Improved Principal Component Analysis**

The presence of noise and irrelevant information in cybersecurity data can make it difficult to spot trends linked to attacks. By highlighting the principal components that account for the majority of the variance in the data, Principal Component Analysis (PCA) can aid in the noise reduction process. This improves the signal-to-noise ratio and facilitates the identification of attack-related patterns. PCA looks for a new matrix B which approximates the given matrix eqn. (1) to reduce the dimension space of the input data.

$$\min_{R(B)} \|W-B\|_f^2 \tag{1}$$

Let, $W=\{w_1,.....,w_m\} \in T^{d\times n}$ is the centered matrix, d is the dimension of the sample data, and n is considered as the total number of samples. Then the projection matrix Z is satisfies using eqn. (2).

$$Z \in T^{d\times k}, C \in \overset{\min}{T^{n\times k}}, Z^t Z = I^{\|W-ZC^t\|_f^2} \tag{2}$$

The matrix Z is denoted as eigenvectors of $Z^t Z$, and C is considered as k large eigenvectors. The PCA model has the issues to measure the correct mean. To overcome this issue,a weight matrix is employed in the diagonal element of the PCA model using eqn. (3).

$$d, w, Z \in \overset{\min}{T^{n\times k}}, Z^t Z = \sum_{i=1}^{n} \left\|\left(I - Z^t Z\right)\right\| \left(d_i - w_i\right) \tag{3}$$

Let, d is denoted as a diagonal matrix, and w is represented as a weight matrix. The IPCA model overcomes the mean issues and enhances the dataset quality.

- **Min-Max normalization**

Normalization involves normalizing attributes, with values falling between 0.0 and 1.0. Techniques for classification can exclude properties with broad ranges from those with narrower ranges because of normalization. One of the most used techniques for normalizing data is min-max normalization. All features have their minimum value converted to a zero, their maximum value converted to a one, and all other values converted to a decimal within 0 and 1.To apply the min-max normalization, use eqn. (4).

$$\vec{N} = \frac{n - \mathrm{mi}_v}{\mathrm{ma}_v - \mathrm{mi}_v}\left(ne\_ma_v - ne\_mi_v\right) + ne\_mi_v \tag{4}$$

Let, n is denoted as an input value, v is considered as normalized value, mi_v is denoted as normalized minimum value, and ma_v is considered as normalized maximum value. Moreover, ne_ma_v and ne_mi_v are considered as newly generated normalized maximum and minimum values. After data normalization, normalized data are sent to the feature extraction phase for extracting relevant information from the dataset. It minimizes the execution time and enhances prediction accuracy.

4.3 Feature extraction using N-grams

By examining header data from network communications or by keeping a check on connection efforts and session activity, it is possible to identify several attacks that take advantage of bugs in interfaces and services. Examining the header data alone is insufficient for detecting attempts that try to infect susceptible services or apps with harmful codes or viruses. An analysis of the packets' payload data is required. Certain attack patterns can be identified from the payload by creating a collection of signatures with domain expertise.

To automatically extract features, $n-gram$ extraction process [35] is typically used. Moreover, $n-gram$is a following of n items based on a specified order. If a payload is regarded as a string for attack detection purposes, then an $n-gram$is a substring of i characters. Considering that the payloads of legitimate traffic and malicious traffic are distinct from one another. the automated technique for creating features based on $n-gram$ extraction for attack detection. Let us assume, $n-gram($ $i \geq 1)$, thus the space $\hat{S}$ of every possibility $n-gram$has the size of2^{8i} , as taking into account each character's 8-bit representation is detailed in Eqn (5).

$$\hat{S} = \left\{ n\_gram_k \,/\, k = 1.....2^{8i} \right\} \tag{5}$$

Considering a payload y, a feature vector of y can be built using eqn. (6)

$$g_y = \left(g_1, g_2, ...g_{2^{8i}} \right) \tag{6}$$

Let, g_i is denoted asthe quantity of appearances of n_gram_k in y. To improve the prediction outcomes and choose the most suitable features, the features are updated in the feature selection phase.

4.4 Feature section using CFS-SMO

Techniques for feature selection can be applied to choose the pertinent features that are most likely to be associated with the intended outcomes. A simple feature selection procedure must be used to create an efficient IIoT attack detection mechanism. The Correlation-Based Feature Selection (CFS) [36], a straightforward algorithm that assesses the corresponding connections between the outputs and correlated input characteristics, can be applied in conjunction with the filter-based approach.According to this approach, correlation between characteristics can be used to pick features for deep learning categorization.

The fundamental function of CFS is to heuristically assess feature subset values based on several presumptions: Features that are strongly linked with classes but not with each other make up the optimal feature subset. Equation (7) displays the evaluation criteria in the following manner.

$$S_m = \frac{n\overline{e_{fc}}}{\sqrt{n+n(n-1)\overline{e_{ff}}}} \tag{7}$$

Let, S_m is denoted as the probabilities for all feature subset m valuesthat containing n features, $\overline{e_{fc}}$ is denoted asa mean of the feature-class correlation, and $\overline{e_{ff}}$ is considered as the mean of the feature-feature intercorrelation. Moreover, $\overline{e_{xy}}$ can be determined using a correlation-measurement metric, such as Symmetric Uncertainty (SU), This is explained by Equation (8)

$$S_u(x,y) = 2\left[\frac{k(x)-k\left(x/y\right)}{k(x)+k(y)}\right] \tag{8}$$

Generally, x and y are denoted as independent variables. Information gain serves as the expression's numerator. In actuality, information gain is influenced in favor of features with higher values; however, Equation (8) normalizes the values of these features to fall between [0,1] and corrects for this bias using the denominator. A correlation coefficient of 1 signifies a high correlation between the two variables, whereas a coefficient of 0 denotes their independence from one another. Information gain's bias for features with higher values is counteracted by SU, which adjusts its value to a range of [0,1], where 1 denotes that knowledge of one matches the value of the other and 0 represents the other.Pairs of features are taken into account symmetrically. Although nominal features are necessary for entropy-based techniques to function, if continuous features are discretized appropriately, they are used to assess correlations among continuous characteristics.

Spider Monkey Optimization (SMO): Social interactions among spider monkeys have an impact on SMO [37], a population-level technique. It relies on spider monkeys' clever foraging strategies, which imitate the fission-fusion social structure. Members of FFSS form short-lived, petty organizations whose members are integrated into a larger, more stable society. The abundance and scarcity of food supplies caused the monkeys to divide into victim groups and smaller groups. The model that has been built produces an ideal key to improve the feature selection process by taking into account the foraging habits of spider monkeys.

First of all, SMO produces an initial population that is uniformly distributed of x Spider Monkeys (SMs) where each monkey $m_i\left(i=1,2,...,x\right)$ is a D-dimensional vector and m_i signifies the i^{th} SM in the population. Each dimension j of m_i is adjusted using eqn. (9).

$$m_{ij}=m_{\min j}+rand\left[0,1\right]\left(m_{\max j}-m_{\min j}\right) \tag{9}$$

Let, $m_{\min j}$ and $m_{\max j}$ are limits of m_i in j^{th} direction and $rand\left[0,1\right]$ is a random number with uniform distribution throughout the interval [0,1].

Local Leader Phase (LLP): In this step, each SM modifies its present position in response to the information gathered from the leaders' and neighbourhood members' actions. The fitness value of the recently acquired position is assessed. If the new location's fitness rating is higher than the prior one's, the SM exchanges places with it. In this phase, updates the location of the i^{th} SM using eqn. (10).

$$m_{Newij}=m_{ij}+rand\left[0,1\right]\left(Lg_{nj}-m_{ij}\right)+rand\left[-1,1\right]\left(m_{lj}-m_{ij}\right) \tag{10}$$

Let, m_{ij} is denoted as the j^{th} dimension of the i^{th} SM, Lg_{nj} is represented as j^{th} dimension of the l^{th} position of the local group leader. m_{lj} shows the j^{th} measurement of the l^{th} SM and is randomly selected from the n^{th} group such that $n=i$, $rand\left[-1,1\right]$ is a random number with a uniform distribution between -1 and 1.

Global leader phase: The GLP begins when the LLP is finished. All of the SMs reassess their stances based on the experiences of the global leaders and members of local groups. Equation (11) is used to update the location of the GLP.

$$m_{Newij}=m_{ij}+rand\left[0,1\right]\left(Gl_j-m_{ij}\right)+rand\left[-1,1\right]\left(m_{lj}-m_{ij}\right) \tag{11}$$

Let, Gl_j is the j^{th} dimension of the position of the global leader and $j\in\left\{1,2,.....,d\right\}$ is an arbitrary index selection. SMO discovers the best answer by more capably striking a balance between exploration and exploitation. To find the search region, the LLP is used to modify their position with a large perturbation based on the measurements. Even though this promotes exploitation at this point, better features are chosen by shifting their placements along the GLP. To identify the attacks, the proposed model chooses the optimal features based on the LLP and GLP behavior of SM. This performance is updated to the update SU phase and is determined using eqn. (12).

$$S_u(x,y) = 2\left[\frac{k(x)-k\left(x/y\right)}{k(x)+k(y)}\right]\times m_{Newij} \tag{12}$$

The selected features are shown in Table 2. The optimal features are chosen, and the classifier is updated with these characteristics to accurately identify and detect attacks in the IIoT environment.

Table 2 Selected features

Feature name	Description	Type
dttl	final destination to the source of lifetime	Integer
sbytes	bytes from source to destination	Integer
ct_dst_sport_ltm	number of rows in 100 rows of identical dstip and sport	Integer
service	Such asftp, http, ssh, smtp, dns and ftp-data	Nominal
sload	Bits per second of the source	Float
sttl	Time to live from source to destination	Integer
ct_srv_dst	number of rows in 100 rows with the identical service and dstip	Integer

4.5 Attack detection using FFNN-GEO

A fully connected, FFNN classifier [38] is called a multi-layer perceptron. The suggested model has two hidden layers with 256 nodes each, and it accepts the normalized results of the chosen features as inputs. For the 14 different forms of attacks and benign traffic, the classifier produces 15 outputs. By randomly removing units from the MLP with a possibility prb, dropout is a regularization approach for hidden layers that assists in preventing over-adjustment on training data.Neural network output $d^{(3)}$ is determined by using equations (13), (14), and (15) to connect the outputs of the several layers. Let, $a, w_t^{(i)}$ and $b_t^{(i)}$ are denoted asthe input vector, weights matrix, and the bias vector for layer i.

$$d^{(1)} = h_1\left(w_t^{(1)T}.a + v^{(1)}\right) \tag{13}$$

$$d^{(2)} = h_1\left(w_t^{(2)T}.d^{(1)} + v^{(2)}\right) \tag{14}$$

$$d^{(3)} = h_2\left(w_t^{(3)T}.d^{(2)} + v^{(3)}\right)' \tag{15}$$

Activation functions h_1 and h_2 provide the neural network uncertainty. Since MLP does not automatically normalize its outputs, hidden layers employ the Scaled Exponential Linear Unit (SELU) with their activation function using equation (16). A softmax activation function is used in the output layer h_2 and isgiven in equation (17). softmax activation function is employed to enable the interpretation of each output and predict a specific class. The predicted label $\hat{b}$ is given by $\hat{b} = \arg\max d^{(3)}$.

$$h_1(x) = \lambda \begin{Bmatrix} \alpha\left(e^x - 1\right) & for x < 0 \\ x & for x \geq 0 \end{Bmatrix} \tag{16}$$

$$h_2(x)_j = \frac{e^{x_j}}{\sum_{l=1}^{n} e^{x_l}} \; for j \in [1;15] \tag{17}$$

The training set is split up into mini-batch sizes of 32 occurrences for training and implementing the model. MLP learns categorization by adjusting the weights w_t among neural network nodes to minimize the cross-entropy loss function $L(w_t)$ using eqn. (18).

$$L(w_t) = -\left[b.\log\left(\hat{b}\right) + (1-b).\log\left(1-\hat{b}\right)\right] \tag{18}$$

Let, b is denoted asthe ground truth label and $\hat{b}$ is considered apredicted class. By assessing the validation data set, the neural network's risk of over-fitting is managed. To maintain a minimal performance disparity among the training and verification sets, the dropout parameter prb is modified. The difficult effort of determining the ideal collection of variables to attain the best performances is known as hyperparameter tuning.A Golden Eagle Optimization (GEO) was applied to tune weight ($w_t^{(i)}$), bias ($b_t^{(i)}$), and entropy loss function ($L(w_t)$). It improves performance and the outcome of the predictions. The process of FFNN-GEO is shown in Fig. 2.

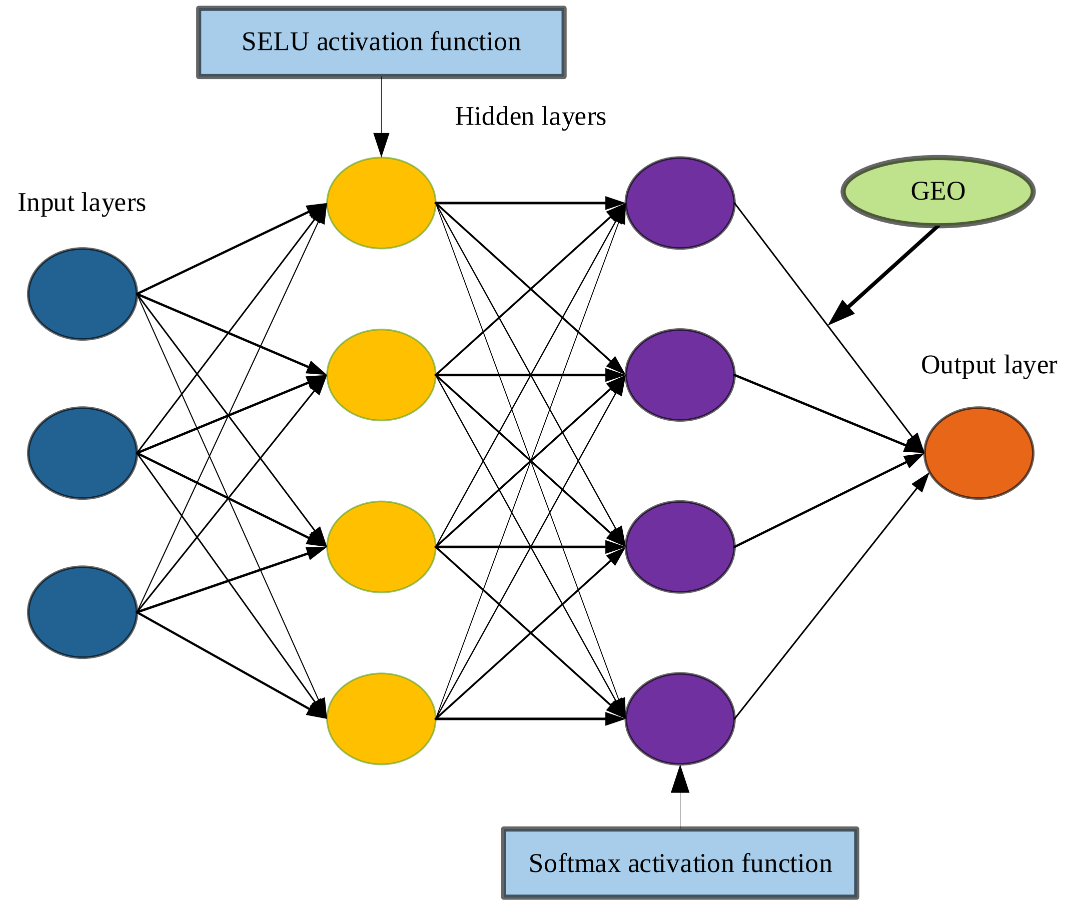

Fig.2 Process of developed FFNN-GEO model

- **Golden Eagle Optimization**

To improve the FFNN classifier and determine the ideal parameters, this study uses the GEO [39] technique. GEO is utilized to adjust the FFNN parameters such as $w_t^{(i)}, b_t^{(i)}$, and $L(w_t)$. The brains of golden eagles, which change their speed at different places along their spiral path to hunt, served as the model for a GEO algorithm. Problems involving global optimization are solved using this method. It exhibits a greater interest in exploring new areas and searching for prey in the early stages of the hunt, and a greater urge to attack in the later stages. The golden eagle likewise tinkers with these two gadgets to quickly seize its perfect meal. This behavior is emphasized to demonstrate the efficacy of the GEO strategy's study and application. It also establishes the hyperparameter's ideal value. The GEO technique consists of the following phases. Fig. 3 displays the GEO algorithm's flowchart.

Initialization: The GE population is first determined by its spiral motion. Every GE maintains a record of the places they have already visited. Every GE has a predetermined population and memory. The population size in this work is represented by the total number of data points (n)

Random generation: For every iteration (n), a distinct golden eagle (x) chooses to aim for a random GE. It focuses on a more advantageous location that the golden eagle visits. Additionally, GE (x) chooses the circles that stand in for memory.

Fitness Function: A random solution should result from the initialized values. The setting of the parameter for the optimization is then displayed in the objective function and fitness function solutions such as $w_t^{(i)}, b_t^{(i)}$, and $L(w_t)$ for tuningthe FFNN classifier. The fitness function is evaluated using eqn. (19).

$$F(t) = opt\left[w_t^{(i)}, b_t^{(i)} \text{ and } L(w_t)\right] \tag{19}$$

Optimize $\left(w_t^{(i)}, b_t^{(i)}\right)$ **usingGE exploitation behavior:** Depending on the prey's current location, this attack ends with the prey's position being stored in the golden eagle's memory. The GE's vector of exploitation $(E_x(v))$ is in eqn. (20).

$$(E_x(v)) = C_l^{\bullet} - C_p + \left(w_t^{(i)}, b_t^{(i)}\right) \tag{20}$$

Let $C_l^{\bullet}$ is denoted as the greatest place that GE visited, and C_p is denoted as the GE current position.

Optimize $L(w_t)$ **usingGE explorationbehavior:** The exploitation vector is used to locate the cruise vector. It is tangent to the circle and then diagonal to the vector of exploitation. Moreover, the cruise is converted into quadratic speed by GE, which is connected to prey. Thus, the cruise vector in n-dimensions, which is calculated using eqn. (21), is contained in the tangent hyperplane of a circle.

$$\sum_{p=1}^{x} E_p(v) h_p = \sum_{p=1}^{x} E^i{}_p(v) h_p^{\bullet} + L(w_t) \tag{21}$$

The vector is denoted as $E^i{}_p(v) = \left[E^1{}_p(v), E^2{}_p(v), \ldots. E^n{}_p(v)\right]$ and the variable decision vector is denoted as $h_p = \left[h_1, h_2 \ldots. h_n\right]$, selected prey position signifies $h_p^{\bullet} = \left[h_1^{\bullet}, h_2^{\bullet} \ldots. h_n^{\bullet}\right]$

that is active to improve the parameters of FFNN. An investigation of GEO's exploitation behavior led to an adjustment in the fitness function's value. The best categorization parameters are found using GEO and attain better prediction results.

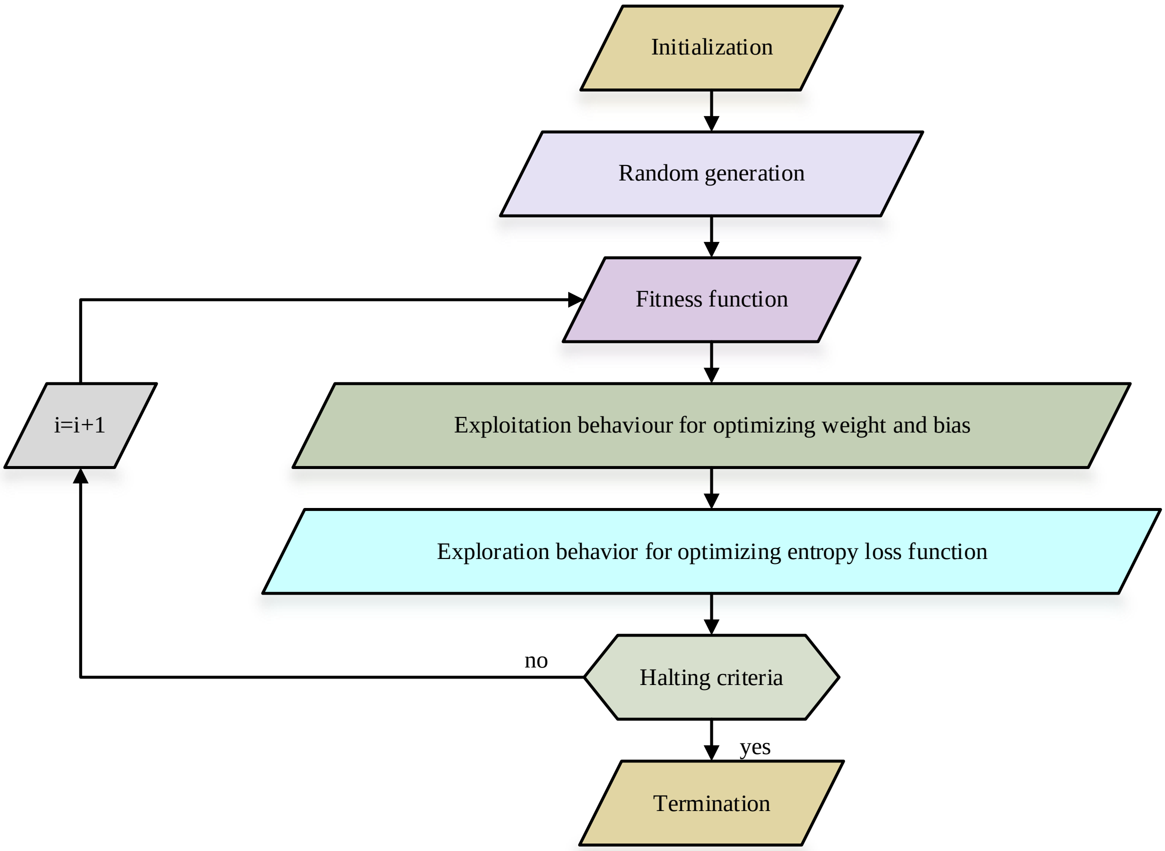
Initialization
Random generation
Fitness function
i=i+1
Exploitation behaviour for optimizing weight and bias
Exploration behavior for optimizing entropy loss function
no
Halting criteria
yes
Termination

Fig.3 Flowchart representation of the GEO algorithm

Termination: The enhanced exploitation incorporates GEO's exploring behavior and uses it to refine the parameters of the FFNN classifier. The developed objective function improves accuracy and reduces computation time while accounting for errors. The third stage of the process is repeated until the conditions for termination are met $i = i + 1$.

5. RESULTS AND DISCUSSION

The developed technique is implemented in the Python tool and the performance of the developed technique is validated with existing models in terms of accuracy, precision, F-measure, sensitivity, specificity, execution time, and error rate. Moreover, GEO is employed to optimize the FFNN parameters for enhancing the detection performance. The designed technique effectively detects the attacks in the IIoT environment with less error rate and high detection accuracy.

5.1 Performance analysis

The developed model performance is analyzed using the UNSW-NB15 dataset and the existing techniques used for the validation process are RF [22], CNN+LSTM [23], 2DF-IDS [25], and CFBPNN [27]. To validate the performance of the developed model by generating better results in accuracy, sensitivity, specificity, precision, F-measure, error rate, and execution time.

5.1.1Accuracy

A measure of agreement between the detected true value and the detection result is called detection accuracy. The comparison of the accuracy with other existing models is shown in Fig. 4.

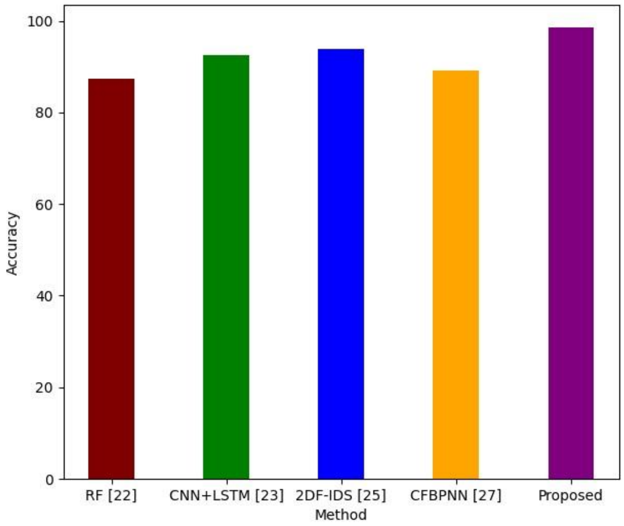

Fig.4 Accuracy comparison

The accuracy performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained 87.36% accuracy, the CNN-LSTM technique gained 92.46% accuracy, the 2DF-IDS model gained 93.87%, accuracy, and the CFBPNN model gained 89.16% accuracy. The designed technique attained a high rate of accuracy in detecting attacks of 98.56%, which is high when comparing other models.

5.1.2 Precision

The ratio of genuine positives to all positive predictions is known as precision. a classification framework's ability to pinpoint only the pertinent data points. The comparison of the precision with other existing models is shown in fig.5.

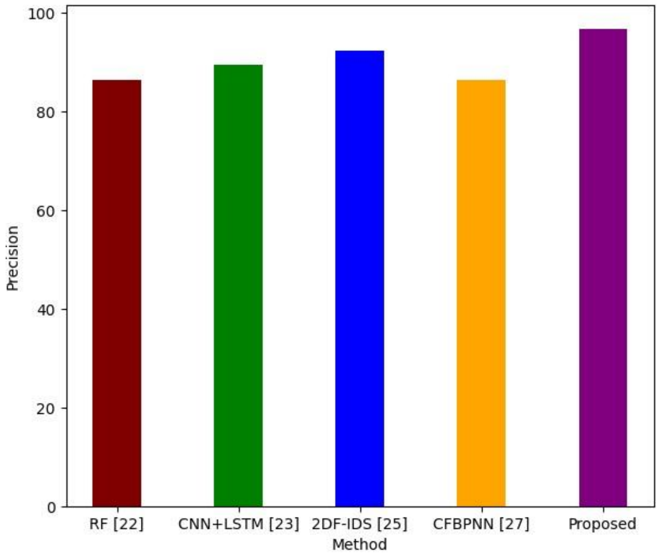

Fig.5 Precision comparison

The precision performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained 86.46% precision, the CNN-LSTM technique gained 98.47% precision, the 2DF-IDS model gained 92.45%, precision, and the CFBPNN model gained 86.35% precision. The designed technique attained a high rate of precision to detect attacks at 96.77%, which is high when comparing other models.

5.1.3 Sensitivity

Sensitivity (SN) is determined by dividing the overall number of positives by the quantity of correctly predicted positives. The comparison of the sensitivity with other existing models is shown in Fig. 6.

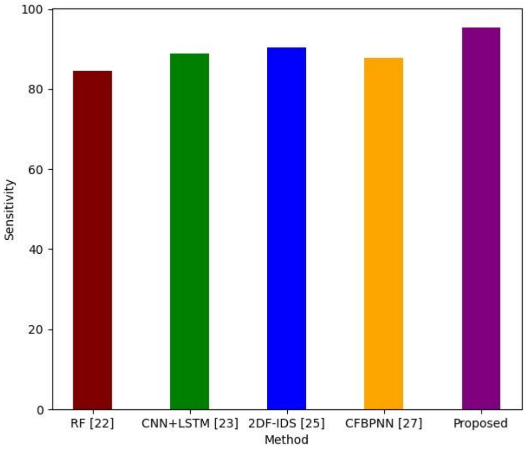

Fig.6 Sensitivity comparison

The sensitivity performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained 84.56% sensitivity, the CNN-LSTM technique gained 88.81% sensitivity, the 2DF-IDS model gained 90.46%, sensitivity, and the CFBPNN model gained 87.68% sensitivity. The designed technique attained a high rate of sensitivity to detect attacks at 95.34%, which is high when comparing other models.

5.1.4 Specificity

The ratio of accurate negative predictions to total negatives is used to compute specificity. The comparison of the specificity with other existing models is shown in Fig. 7.

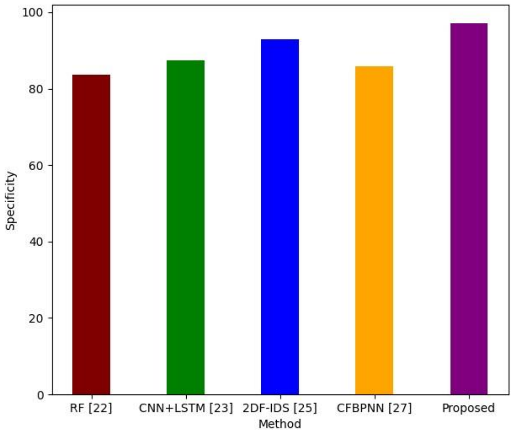

Fig.7 Specificity comparison

The specificity performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained 83.56% specificity, the CNN-LSTM technique gained 87.34% specificity, the 2DF-IDS model gained 92.99%, specificity, and the CFBPNN model gained 85.90% specificity. The designed technique attained a high rate of specificity to detect attacks at97.12%, which is high when comparing other models.

5.1.5 F-measure

A deep learning algorithm's efficiency is assessed using a statistic called F-measure. The average weight of precision and recall is F-measure. It generates a single score by combining recall and precision. The comparison of the F-measure with other existing models is shown in Fig. 8.

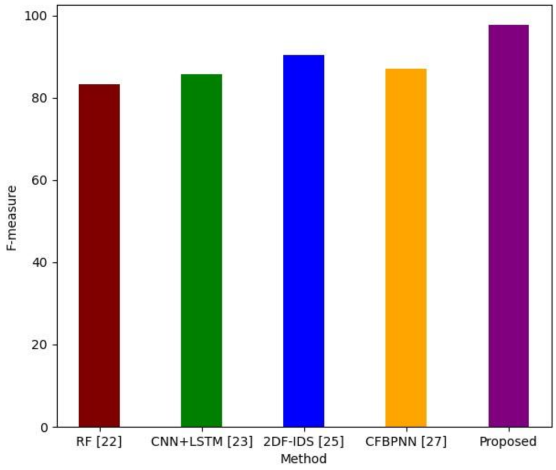

Fig.8 F-measure comparison

The F-measure performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained 83.23% F-measure, the CNN-LSTM technique gained 85.66% F-measure, the 2DF-IDS model gained 90.36%, F-measure, and the CFBPNN model gained 87.00% F-measure. The designed technique attained a high rate of F-measure to detect attacks of 97.76 %, which is high when comparing other models.

5.1.6 Error rate

The calculation of the error rate involves dividing the total number of inaccurate predictions from the total amount of datasets. A model's amount of prediction error about the genuine model is measured by its error rate. The comparison of the error rate with other existing models is shown in Fig. 9.

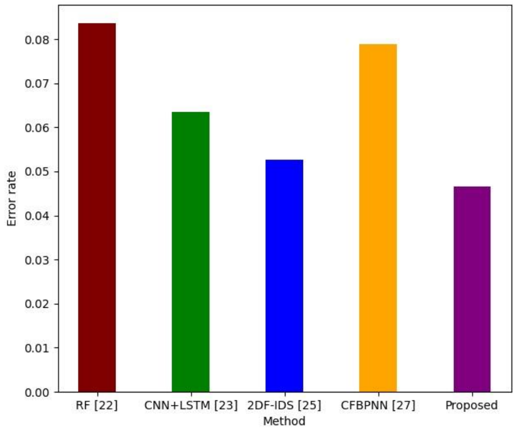

Fig.9 Error rate comparison

The error rate performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained a 0.083% error rate, the CNN-LSTM technique gained a 0.063% error rate, the 2DF-IDS model gained a 0.052%, error rate, and the CFBPNN model gained a 0.078% error rate. The designed technique attained a low rate of error rate to detect attacks of 0.046%, which is low when comparing other models.

5.1.7 Execution time

Overall execution time is the amount of time the system needs to do all of the tasks, measured from the moment the request is received to the point at which the final process is finished. The comparison of the execution time with other existing models is shown in Fig. 10.

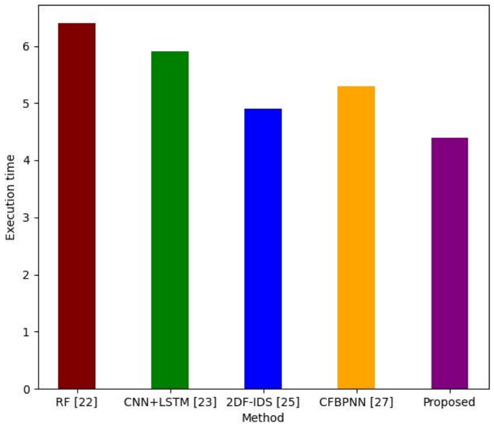

Fig.10 Execution time comparison

The execution time performance of the proposed technique is validated with existing models such as RF, CNN-LSTM, 2DF-IDS, and CFBPNN. The RF model gained 6.4sexecution time, the CNN-LSTM technique gained 5.9sexecution time, the 2DF-IDS model gained 4.9sexecution time, and the CFBPNN model gained 5.3sexecution time. The designed technique attained a high rate of execution time to detect attacks of 4.4s which is low when comparing other models.

6. CONCLUSION

The FFNN-GEO technique for Industrial IoT security against assaults is explored in this article. A Python system is initially used to gather and train the UNSW-NB15 dataset. IPCA and min-max normalization are two further data preprocessing techniques used to reduce dimensionality and eliminate noise. To improve the detection performance of the IDS, the suggested mode used the CFS-SMO technique for feature selection. Finally, the improved FFNN-GEO model prevented over-fitting problems and successfully identified and detected the attacks. Comparing the suggested deep learning models' trial outcomes to earlier techniques created using the same dataset, they performed better.These findings show that deep learning techniques are the most effective at identifying anomalous occurrences in big, intricate datasets. In an IIoT network, several ML and DL-based intrusion detection systems are employed to detect cyberattacks. Along with integrating a decision-making unit into the planned system to take appropriate action to halt the detecting attacks. Additionally, look at additional feature selection strategies to speed up the detection systems' training process.

REFERENCES

1. Vargas, H., Lozano-Garzon, C., Montoya, G.A. and Donoso, Y., 2021. Detection of security attacks in industrial IoT networks: A blockchain and machine learning approach. Electronics, 10(21), p.2662.
2. Khan, I.A., Keshk, M., Pi, D., Khan, N., Hussain, Y. and Soliman, H., 2022. Enhancing IIoT networks protection: A robust security model for attack detection in Internet Industrial Control Systems. Ad Hoc Networks, 134, p.102930.
3. Aboelwafa, M.M., Seddik, K.G., Eldefrawy, M.H., Gadallah, Y. and Gidlund, M., 2020. A machine-learning-based technique for false data injection attacks detection in industrial IoT. IEEE Internet of Things Journal, 7(9), pp.8462-8471.
4. Li, X., Xu, M., Vijayakumar, P., Kumar, N. and Liu, X., 2020. Detection of low-frequency and multi-stage attacks in industrial internet of things. IEEE Transactions on Vehicular Technology, 69(8), pp.8820-8831.
5. Taheri, R., Shojafar, M., Alazab, M. and Tafazolli, R., 2020. FED-IIoT: A robust federated malware detection architecture in industrial IoT. IEEE transactions on industrial informatics, 17(12), pp.8442-8452.
6. Yu, K., Tan, L., Mumtaz, S., Al-Rubaye, S., Al-Dulaimi, A., Bashir, A.K. and Khan, F.A., 2021. Securing critical infrastructures: deep-learning-based threat detection in IIoT. IEEE Communications Magazine, 59(10), pp.76-82.
7. Qureshi, K.N., Rana, S.S., Ahmed, A. and Jeon, G., 2020. A novel and secure attacks detection framework for smart cities industrial internet of things. Sustainable Cities and Society, 61, p.102343.
8. Latif, S., Zou, Z., Idrees, Z. and Ahmad, J., 2020. A novel attack detection scheme for the industrial internet of things using a lightweight random neural network. IEEE access, 8, pp.89337-89350.
9. Jahromi, A.N., Karimipour, H., Dehghantanha, A. and Parizi, R.M., 2021. Deep representation learning for cyber-attack detection in industrial iot. AI-Enabled Threat Detection and Security Analysis for Industrial IoT, pp.139-162.
10. Berger, S., Bürger, O. and Röglinger, M., 2020. Attacks on the Industrial Internet of Things–Development of a multi-layer Taxonomy. Computers & Security, 93, p.101790.
11. Borgiani, V., Moratori, P., Kazienko, J.F., Tubino, E.R. and Quincozes, S.E., 2020. Toward a distributed approach for detection and mitigation of denial-of-service

attacks within industrial Internet of Things. IEEE Internet of Things Journal, 8(6), pp.4569-4578.

12. Aouedi, O., Piamrat, K., Muller, G. and Singh, K., 2022. Federated semisupervised learning for attack detection in industrial Internet of Things. IEEE Transactions on Industrial Informatics, 19(1), pp.286-295.
13. Sen, S. and Song, L., 2021, November. An IIoT-Based Networked Industrial Control System Architecture to Secure Industrial Applications. In 2021 IEEE Industrial Electronics and Applications Conference (IEACon) (pp. 280-285). IEEE.
14. Chakraborty, S., Onuchowska, A., Samtani, S., Jank, W. and Wolfram, B., 2021. Machine learning for automated industrial IoT attack detection: An efficiency-complexity trade-off. ACM Transactions on Management Information System (TMIS), 12(4), pp.1-28.
15. Rouzbahani, H.M., Bahrami, A.H. and Karimipour, H., 2021. A snapshot ensemble deep neural network model for attack detection in industrial internet of things. AI-Enabled Threat Detection and Security Analysis for Industrial IoT, pp.181-194.
16. Hassan, M.M., Gumaei, A., Huda, S. and Almogren, A., 2020. Increasing the trustworthiness in the industrial IoT networks through a reliable cyberattack detection model. IEEE Transactions on Industrial Informatics, 16(9), pp.6154-6162.
17. Khan, I.A., Pi, D., Abbas, M.Z., Zia, U., Hussain, Y. and Soliman, H., 2022. Federated-SRUs: A federated simple recurrent units-based IDS for accurate detection of cyber attacks against IoT-augmented industrial control systems. IEEE Internet of Things Journal.
18. Mendonca, R.V., Silva, J.C., Rosa, R.L., Saadi, M., Rodriguez, D.Z. and Farouk, A., 2022. A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms. Expert Systems, 39(5), p.e12917.
19. Khan, F., Alturki, R., Rahman, M.A., Mastorakis, S., Razzak, I. and Shah, S.T., 2022. Trustworthy and Reliable Deep-Learning-Based Cyberattack Detection in Industrial IoT. IEEE Transactions on Industrial Informatics, 19(1), pp.1030-1038.
20. Mosteiro-Sanchez, A., Barcelo, M., Astorga, J. and Urbieta, A., 2020. Securing IIoT using defence-in-depth: towards an end-to-end secure industry 4.0. Journal of Manufacturing Systems, 57, pp.367-378.

21. Soliman, S., Oudah, W. and Aljuhani, A., 2023. Deep learning-based intrusion detection approach for securing industrial Internet of Things. Alexandria Engineering Journal, 81, pp.371-383.
22. Mohy-Eddine, M., Guezzaz, A., Benkirane, S., Azrour, M. and Farhaoui, Y., 2023. An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security. Big Data Mining and Analytics, 6(3), pp.273-287.
23. Altunay, H.C. and Albayrak, Z., 2023. A hybrid CNN+ LSTM-based intrusion detection system for industrial IoT networks. Engineering Science and Technology, an International Journal, 38, p.101322.
24. Yazdinejad, A., Kazemi, M., Parizi, R.M., Dehghantanha, A. and Karimipour, H., 2023. An ensemble deep learning model for cyber threat hunting in the industrial internet of things. Digital Communications and Networks, 9(1), pp.101-110.
25. Friha, O., Ferrag, M.A., Benbouzid, M., Berghout, T., Kantarci, B. and Choo, K.K.R., 2023. 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT. Computers & Security, 127, p.103097.
26. Jahromi, A.N., Karimipour, H. and Dehghantanha, A., 2023. An ensemble deep federated learning cyber-threat hunting model for Industrial Internet of Things. Computer Communications, 198, pp.108-116.
27. Jayalaxmi, P.L.S., Saha, R., Kumar, G., Alazab, M., Conti, M. and Cheng, X., 2023. PIGNUS: A Deep Learning Model for IDS in Industrial Internet-of-Things. Computers & Security, p.103315.
28. Pundir, S., Obaidat, M.S., Wazid, M., Das, A.K., Singh, D.P. and Rodrigues, J.J., 2023. MADP-IIME: malware attack detection protocol in IoT-enabled industrial multimedia environment using machine learning approach. Multimedia Systems, 29(3), pp.1785-1797.
29. Serror, M., Hack, S., Henze, M., Schuba, M. and Wehrle, K., 2020. Challenges and opportunities in securing the industrial internet of things. IEEE Transactions on Industrial Informatics, 17(5), pp.2985-2996.
30. Eyeleko, A.H. and Feng, T., 2023. A Critical Overview of Industrial Internet of Things Security and Privacy Issues Using a Layer-Based Hacking Scenario. IEEE Internet of Things Journal.

31. Tyagi, A.K. and Nair, M.M., 2020. Internet of Everything (IoE) and Internet of Things (IoTs): Threat Analyses, Possible Opportunities for Future. Journal of Information Assurance & Security, 15(5).
32. https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15
33. Elkhadir, Z., Chougdali, K. and Benattou, M., 2017, November. An effective cyber attack detection system based on an improved OMPCA. In 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE.
34. Kamani, G.J., Parmar, R.S. and Ghodasara, Y.R., 2019. Data normalization in data mining using graphical user interface: A pre-processing stage. Gujarat Journal of Extension Education, 30(2), pp.106-109.
35. Devi, G.R., Kumar, M.A. and Soman, K.P., 2020. Extraction of named entities from social media text in tamil language using N-gram embedding for disaster management. Nature-Inspired Computation in Data Mining and Machine Learning, pp.207-223.
36. Mohamad, M., Selamat, A., Krejcar, O., Crespo, R.G., Herrera-Viedma, E. and Fujita, H., 2021. Enhancing big data feature selection using a hybrid correlation-based feature selection. Electronics, 10(23), p.2984.
37. Sharma, H., Hazrati, G. and Bansal, J.C., 2019. Spider monkey optimization algorithm. Evolutionary and swarm intelligence algorithms, pp.43-59.
38. Ezhilarasi, M., Gnanaprasanambikai, L., Kousalya, A. and Shanmugapriya, M., 2023. A novel implementation of routing attack detection scheme by using fuzzy and feed-forward neural networks. Soft Computing, 27(7), pp.4157-4168.
39. Mohammadi-Balani, A., Nayeri, M.D., Azar, A. and Taghizadeh-Yazdi, M., 2021. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. Computers & Industrial Engineering, 152, p.107050.

Author Biography

Ajay Chandra MK(Ajay Chandra Manukondakrupa) is a seasoned professional with a master's degree in computer science and a remarkable career spanning over 24 years in the technology space. His journey through various domains, including Automobile, Pharma, Retail, Banking, and Telecom, has established him as a versatile and accomplished leader. Throughout his career, Ajay has honed his expertise in automation and the implementation of Enterprise Applications and ERP systems, playing a pivotal role in digital transformation projects across industries. His ability to conceptualize and execute comprehensive tech strategies, coupled with a strong aptitude for communication and response management, has been a hallmark of his success. Ajay has held influential positions at renowned organizations, including the World Bank, AbbVie, Ford Motors, Sephora, and currently working as Global Technical Program Manager for a telecommunications and satellite Organization. In these roles, he served as a technical program manager, overseeing cloud computing and SAP S4 HANA implementations. His leadership and technical acumen have contributed significantly to these companies' growth and digital evolution. Ajay Manukonda's career is a testament to his dedication to the ever-evolving world of technology and his profound impact on digital transformation in diverse sectors. His commitment to innovation and excellence continues to drive his career forward, making him a true luminary in the field of technology and enterprise solutions.