Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039446
Article

An Efficient Cyber Security and Intrusion Detection System Using CRSR with PXORP-ECC and LTH-CNN

Nouf Saeed Alotaibi*

Department of Computer Science, College of Science and Humanities Al Dawadmi, Shaqra University, Shaqra, 11961, Saudi Arabia

*Corresponding Author: Nouf Saeed Alotaibi. Email: n.saeed@su.edu.sa

Received: 30 January 2023; Accepted: 01 June 2023; Published: 30 August 2023

Abstract: Intrusion Detection System (IDS) is a network security mechanism that analyses all users' and applications' traffic and detects malicious activities in real-time. The existing IDS methods suffer from lower accuracy and lack the required level of security to prevent sophisticated attacks. This problem can result in the system being vulnerable to attacks, which can lead to the loss of sensitive data and potential system failure. Therefore, this paper proposes an Intrusion Detection System using Logistic Tanh-based Convolutional Neural Network Classification (LTH-CNN). Here, the Correlation Coefficient based Mayfly Optimization (CC-MA) algorithm is used to extract the input characteristics for the IDS from the input data. Then, the optimized features are utilized by the LTH-CNN, which returns the attacked and non-attacked data. After that, the attacked data is stored in the log file and non-attacked data is mapped to the cyber security and data security phases. To prevent the system from cyber-attack, the Source and Destination IP address is converted into a complex binary format named 1's Complement Reverse Shift Right (CRSR), where, in the data security phase the sensed data is converted into an encrypted format using Senders Public key Exclusive OR Receivers Public Key-Elliptic Curve Cryptography (PXORP-ECC) Algorithm to improve the data security. The Network Security Laboratory–Knowledge Discovery in Databases (NSL-KDD) dataset and real-time sensor are used to train and evaluate the proposed LTH-CNN. The suggested model is evaluated based on accuracy, sensitivity, and specificity, which outperformed the existing IDS methods, according to the results of the experiments.

Keywords: Intrusion detection system; logistic tanh-based convolutional neural network classification (LTH-CNN); correlation coefficient based mayfly optimization (CC-MA); cyber security

1 Introduction

With the phenomenal rise of network-based services and sensitive data on networks, network and cyber security are becoming more vital than ever [1]. Cyber security refers to the technologies, methods, and practices that secure internet-connected systems, such as networks, computers, programs, and data, against external cyberattacks such as Denial of Service (DoS), User-to-root (U2R), Remote-to-user (R2L), probing, and so on [2,3]. Thus, the Cyber security process safeguards the network from unauthorized users and protects the user's data against any illegitimate modifications, this ensures the confidentiality, and integrity of the internet-connected systems [4,5].

One component of a cyber security system is an intrusion detection system (IDS). IDSs are used to discover, determine, and identify intrusions by analyzing data generated by network devices, and hence play a critical part in network security [6]. The IDS can be divided into two groups based on their detection mechanisms: exploitation detection, which is signature-based, and anomaly detection, which is behavior-based [7]. Exploitation detection techniques rely on a database of predefined attack patterns to detect attacks [8]. As a result, they are highly effective at detecting known attacks and are preferred due to their low false-positive rate [9]. However, because unknown attacks are not included in the specified pattern lists, misuse detection systems are unable to protect the system against them. Furthermore, this method is incapable of detecting zero-day attacks [10,11]. Anomaly detection approaches, on the other hand, use regular system operations to identify anomalies as behaviors that differ from the norm [12]. Such methods are appealing because they can identify all known and unknown sorts of attacks, including zero-day attacks [13]. The fundamental problem of anomaly detection methods is that they require tweaking and have significant false-positive rates [14]. Many IDSs today are rule-based systems, meaning that their performance is heavily reliant on the rules defined by security experts. The process of encoding rules is expensive and lengthy due to the large amount of network traffic [15].

Hence, several existing research explains Machine Learning (ML) algorithms for the cyber intrusion detection system to avoid such constraints and improve detection [16]. However, the majority of the classification models failed to detect unknown threats, and they must be retrained regularly to maintain high detection rates [17]. This is not feasible because obtaining tagged data is challenging. Similarly, most present solutions aim to secure sensitive data while neglecting to optimize system performance [18]. Hence, to address such problems, the work has proposed an efficient cyber security system and intrusion detection using CRSR with PXORP-ECC and LTH-CNN, which efficiently detects cyber-attacks and preserves the security of the network.

The rest of the paper is organized as follows: Section 2 surveys the associated works regarding the proposed method, Section 3 explains the proposed methodology called an efficient cyber security system and intrusion detection using CRSR with PXORP-ECC and LTH-CNN, and Section 4 illustrates the results and discussion for the proposed method based on performance metrics. Finally, Section 5 concludes the paper with future work.

2 Previous Studies

Haghnegahdar et al. [19] introduced a new intrusion detection model that classified binary-class, triple-class, and multi-class cyber-attacks and power-system incidents efficiently. A whale optimization algorithm (WOA)-trained artificial neural network (ANN) served as the basis for the intrusion detection model. The WOA was used to initialize and update the weight vector of the ANN, resulting in the lowest mean square error. In a power system, the developed WOA-ANN model handled the difficulties of assaults, failure prediction, and detection. The created model was compared to other extensively used classifiers in the experimental investigation. The WOA-ANN model was shown to be superior in comparison. However, this scheme's privacy preservation rate was quite inadequate.

Another intelligent intrusion detection model is developed by Al-Omari et al. [20] which accurately predicted and identified cyber-attacks. The model was created using the Decision Trees idea and took into account the ranking of security aspects. An actual dataset for network intrusion detection systems was used to test the model. It was also validated using predetermined performance evaluation measures, such as accuracy, precision, recall, and F-score. Meanwhile, the experimental results demonstrated that, when compared to other standard machine learning techniques, the tree-based intrusion detection model efficiently recognized and forecasted cyber-attacks while also reducing the complexity of the computation process. The network throughput, on the other hand, was not taken into account in this method.

Suzen et al. [21] presented a hybrid deep belief network (DBN) cyber intrusion detection system for industrial control systems (ICS) that managed network traffic and improved network security. Contrastive divergence (CD) was used to update the DBN's hidden layers, and the output layer was integrated with the Softmax classifier in this model. As a result, numerous limitations, such as the complexity and size of training data, were overcome by the created model architecture. The hybrid DBN model exhibited 99.72% accuracy in intrusion detection and classification, according to the testing results. As a result, the model outperformed the existing intrusion detection system (IDS). It also improved accuracy by about 5% more with the hybrid model than with the older DBN-based systems. But the scheme was inefficient in handling intelligent attacks.

Kumar et al. [22] introduced two different Machine Learning approaches, both supervised and unsupervised. The techniques discussed were Naive Bayes (supervised learning) and Self-Organizing Maps (unsupervised learning). For feature extraction, Convolutional Neural Network (CNN), a deep learning algorithm, was applied. The two machine learning processes were executed on both types of altered datasets, and the results were compared to the accuracy of intrusion detection. The User to Root attack had the best Detection Rate (DR) at 93.0%, while Denial of Service attacks had the worst DR at 0.02%. But the scheme was not generalizable to the real world because of the insufficient features.

In Supervisory Control and Data Acquisition Systems (SCADA), Khan et al. [23] proposed a Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA systems (HML-IDS). Initially, preprocessing procedures that normalized and scaled the data were used. Second, dimensionality reduction algorithms were used, which improved the anomaly identification process. Third, the dataset was balanced using a modified nearest-neighbor rule technique. Fourth, the Bloom filter was employed to establish a signature database. Finally, new threats were discovered by merging package contents-level detection with another instance-based learner, resulting in a hybrid anomaly detection system. The HML-IDS technique exceeded the benchmark models with an accuracy rate of 97% when tested on a genuine large-scale dataset collected from a gas pipeline SCADA system. However, the scheme required a large amount of labeled data for the training process.

The need for Machine Learning based Intrusion Detection Systems arises from the fact that traditional rule-based or signature-based IDS are often not effective in detecting and preventing sophisticated and constantly evolving cyber-attacks. These attacks can exploit unknown vulnerabilities or use advanced evasion techniques that can bypass rule-based systems. Machine learning techniques offer a promising approach for building more effective IDS because they can automatically learn and adapt to changing attack patterns and network behavior. ML-based IDS can analyze substantial amounts of network traffic data and detect anomalies or patterns that may indicate malicious activity.

3 Proposed Intrusion Detection System

Because of the reliance on information technology, a significant quantity of sensitive user and business data is exchanged over the network, making it more vulnerable to cyberattacks that threaten data confidentiality, integrity, and availability [24,25]. Intrusion detection is an important part of the network defense process since it alerts security administrators to hostile behaviors including intrusions, attacks, and malware. An intrusion detection system (IDS) is a software program that monitors network activity and reports malicious activity related to a specific attack. In this paper, an efficient intrusion detection system using LTH-CNN is proposed. The proposed system detects the attacked and non-attacked data based on the features extracted from the input data. After that, Cyber Security and Data Security are carried out for the non-attacked data to improve network security. The proposed methodology is depicted as a block diagram in Fig. 1.

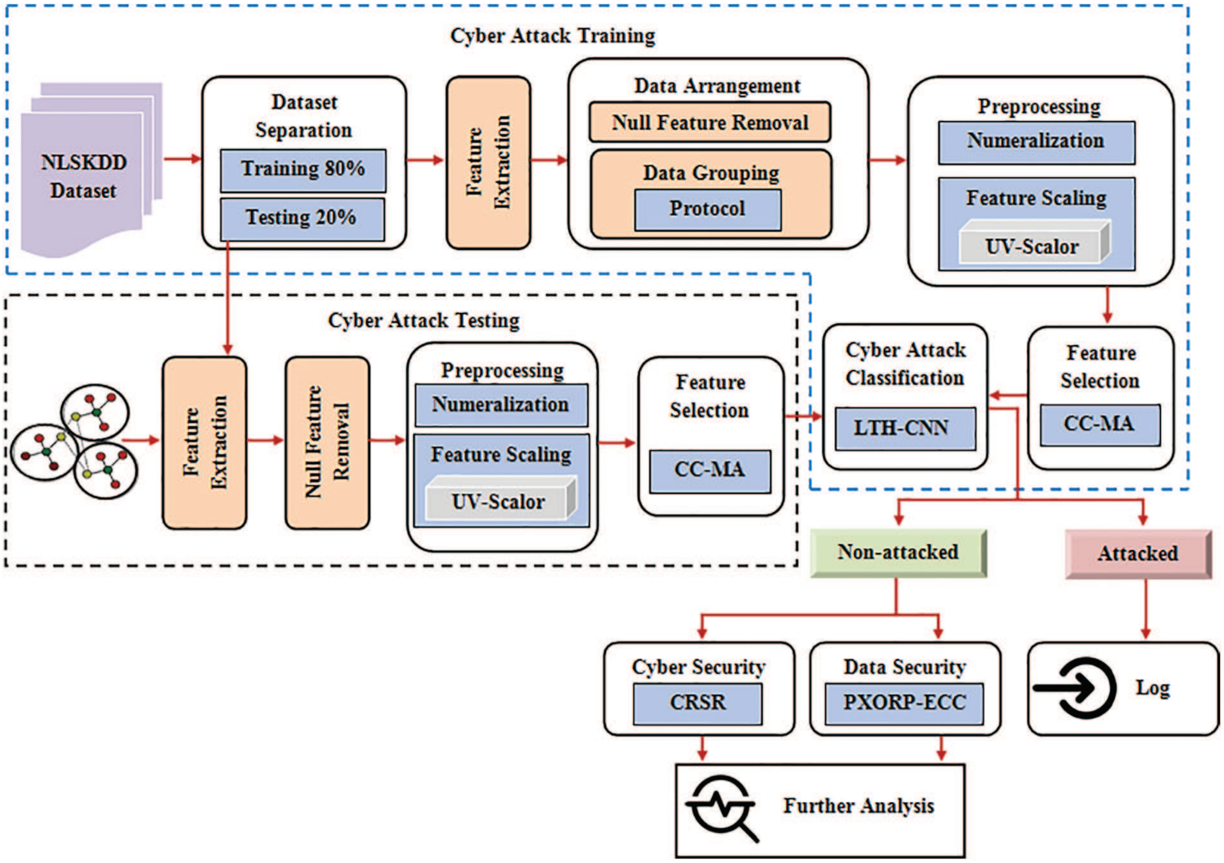

Figure 1: Block diagram of the proposed methodology

3.1 Data Acquisition

Initially, the input data is collected from the “NSL-KDD dataset” which contains the records of the internet traffic. The collected data is then split into training data and testing data. After that, the prominent features are extracted from the data. The extracted features are expressed as:

$$\wp_{fea(n)} = \{\wp_{fea(1)}, \wp_{fea(2)}, \wp_{fea(3)}, \ldots., \wp_{fea(N)}\} \tag{1}$$

where, $\wp_{fea(n)}$ denotes the extracted features, $\wp_{fea(N)}$ is the N^{th} number of features.

3.2 Data Arrangement

After feature extraction, the extracted features $\wp_{fea(n)}$ are arranged into a well-structured format to improve the data analysis and reduce the processing time. This phase contains two steps called null feature removal and data grouping.

(a) Null feature removal: As the presence of null values affects the performance and accuracy of the system, it is important to remove the null values before processing the data. Therefore, this step discards features with null values and selects more prominent features as the input data.

(b) Data grouping: This step is to improve the efficiency of estimation in which the data are divided into three groups using protocols such as Transmission Control Protocol (TCP), Uesr Datagram Protocol (UDP), and Internet Contorl Message Protocol (ICMP). It makes the system focus on the important subpopulation by neglecting the irrelevant ones.

Hence the structured features after certain arrangements can be expressed as:

$$\wp_{AF(n)} = \{\wp_{AF(1)}, \wp_{AF(2)}, \wp_{AF(3)}, \ldots., \wp_{AF(N)}\} \tag{2}$$

where, $\wp_{AF(n)}$ signifies the structured data.

3.3 Pre-Processing

After data arrangement, the preprocessing steps are properly applied to clean the data $\wp_{AF(n)}$ to make it useful. The proposed work comprises two preprocessing steps such as numeralization and feature scaling. These steps are as follows:

Numeralization: Numeralization is the process of converting the raw data from strings to numerical values as the strings cannot be used to predict the output.

Feature scaling: Feature scaling is one of the most critical steps to bind the feature values between two numbers, typically, between [0, 1] or [−1, 1]. In the proposed methodology, the feature values are scaled to range between the ranges 0 to 1 which is done by using the uv-Scalar technique. The feature scaling can be done as:

$$\wp_{sclFea(n)} = \frac{\wp_{AF(n)}}{||\wp_{AF(n)}||} \tag{3}$$

where, $\wp_{sclFea(n)}$ refers to the output features after preprocessing.

3.4 Feature Selection

In this phase, the prominent features are selected using the Correlation Coefficient based Mayfly Optimization (CC-MFO) Algorithm. The MFO algorithm is a meta-heuristic algorithm inspired by the social performance of the mayflies. In the proposed method, the mayflies are considered as the preprocessed features $\wp_{sclFea(n)}$. The algorithm works based on the assumption that mayflies are created as adults where the fittest ones survive. In the existing Mayfly algorithm, the Cartesian distance is unsatisfactory to identify the global best solution. So, to solve this problem and improve convergence speed toward the best solution identification, the Correlation Coefficient technique is used in the proposed CC-MA method. The algorithm steps are as follows:

Initially, two groups of mayflies that represent both the male and female populations are generated. It is expressed as:

$$ {}^{m}\wp_{sclFea(n)} = \{{}^{m}\wp_{sclFea(1)}, {}^{m}\wp_{sclFea(2)}, {}^{m}\wp_{sclFea(3)}, \ldots.., {}^{m}\wp_{sclFea(N)}\} \tag{4} $$

$$ {}^{fm}\wp_{sclFea(n)} = \{{}^{fm}\wp_{sclFea(1)}, {}^{fm}\wp_{sclFea(2)}, {}^{fm}\wp_{sclFea(3)}, \ldots.., {}^{fm}\wp_{sclFea(N)}\} \tag{5} $$

where, ${}^{m}\wp_{sclFea(n)}, {}^{fm}\wp_{sclFea(n)}$ are the male and female populations in N-dimensional vector. The change in position also known as the velocity of the male and female mayflies is considered as, ${}^{m}\varpi_{\wp_{sclFea(n)}} = \{{}^{m}\varpi_{\wp_{sclFea(1)}}, {}^{m}\varpi_{\wp_{sclFea(2)}}, \ldots, {}^{m}\varpi_{\wp_{sclFea(n)}}\}$ and ${}^{fm}\varpi_{\wp_{sclFea(n)}} = \{{}^{fm}\varpi_{\wp_{sclFea(1)}}, {}^{fm}\varpi_{\wp_{sclFea(2)}}, \ldots, {}^{fm}\varpi_{\wp_{sclFea(N)}}\}$ respectively.

The mayflies change their moving direction based on the interaction of the individual's best location and the best location obtained by all mayflies. Thus, the position of the mayflies can be adjusted by adding the velocity as follows:

$$ {}^{m(b+1)}ps_{\wp_{sclFea(n)}} = {}^{m(b)}ps_{\wp_{sclFea(n)}} + {}^{m(b+1)}\varpi_{\wp_{sclFea(n)}} \tag{6} $$

$$ {}^{fm(b+1)}ps_{\wp_{sclFea(n)}} = {}^{fm(b)}ps_{\wp_{sclFea(n)}} + {}^{fm(b+1)}\varpi_{\wp_{sclFea(n)}} \tag{7} $$

where, ${}^{m(b+1)}\varpi_{\wp_{sclFea(n)}}$ and ${}^{fm(b+1)}\varpi_{\wp_{sclFea(n)}}$ refers to the velocities of male and female mayflies at iteration $b+1$, ${}^{m(b+1)}ps_{\wp_{sclFea(n)}}$ and ${}^{m(b)}ps_{\wp_{sclFea(n)}}$ are the position of male mayflies at iteration $b+1$ and b, ${}^{fm(b+1)}ps_{\wp_{sclFea(n)}}$ and ${}^{fm(b)}ps_{\wp_{sclFea(n)}}$ are the position of female mayflies at iteration $b+1$ and b, respectively.

The male mayflies cannot develop great speed and move constantly as they are always a few meters above water performing the nuptial dance. Therefore, the velocity of male mayflies can be calculated as:

$$ {}^{m(b+1)}\varpi_{\wp_{sclFea(n)}} = {}^{m(b)}\varpi_{\wp_{sclFea(n)}} + \frac{g_1\left({}^{m(pbest)}ps_{\wp_{sclFea(n)}} + {}^{m(b)}ps_{\wp_{sclFea(n)}}\right)}{exp(\varepsilon c_{pb}^2)} + \frac{g_2\left({}^{m(gbest)}ps_{\wp_{sclFea(n)}} + {}^{m(b)}ps_{\wp_{sclFea(n)}}\right)}{exp(\varepsilon c_{gb}^2)} \tag{8} $$

$$ {}^{m(pbest)}ps_{\wp_{sclFea(n)}} = \begin{cases} {}^{m(b+1)}ps_{\wp_{sclFea(n)}} & if\,(fit({}^{m(b+1)}ps_{\wp_{sclFea(n)}}) < fit({}^{m(pbest)}ps_{\wp_{sclFea(n)}})) \\ {}^{m(b)}ps_{\wp_{fea(n)}} & otherwise \end{cases} \tag{9} $$

where, ${}^{m(pbest)}ps_{\wp_{sclFea(n)}}, {}^{m(gbest)}ps_{\wp_{sclFea(n)}}$ are the personal best solution and the global best solution of the swarms, c_{pb}, c_{gb} are the correlation coefficients of ${}^{m(pbest)}ps_{\wp_{sclFea(n)}}$ and ${}^{m(b)}ps_{\wp_{sclFea(n)}}$, ${}^{m(gbest)}ps_{\wp_{sclFea(n)}}$ and ${}^{m(b)}ps_{\wp_{sclFea(n)}}$, g_1, g_2 are the positive attraction constants of personal and global, ε is the visibility coefficient to balance the values, $fit(\cdot)$ is the objective function evaluated based on the classification accuracy. The correlation coefficient can be computed as:

$$ c = \frac{Cov({}^{m(b)}ps_{\wp_{sclFea(n)}}, {}^{m(pbest,gbest)}ps_{\wp_{sclFea(n)}})}{\delta_{{}^{m(b)}ps_{\wp_{sclFea(n)}}}\,\delta_{{}^{m(pbest,gbest)}ps_{\wp_{sclFea(n)}}}} \tag{10} $$

where, $\delta_{{}^{m(b)}ps_{\wp_{sclFea(n)}}}, \delta_{{}^{m(pbest,gbest)}ps_{\wp_{sclFea(n)}}}$ are the standard deviation of the solutions, $Cov(\cdot)$ is the covariance of the solutions.

The best mayflies in the swarm continue to perform an up-and-down nuptial dance which is the important functioning of the algorithm. Hence the changing velocities of the best mayflies are updated as:

$$ {}^{m(b+1)}\varpi_{\wp_{sclFea(n)}} = {}^{m(b)}\varpi_{\wp_{sclFea(n)}} + nd * ran \tag{11} $$

where, nd is the nuptial dance coefficient, ran is the random number.

Instead of assembling in a swarm, the female mayflies fly towards the male mayflies for breeding where the attraction process is performed based on the fitness function. Thus, the velocity of female mayflies is updated as:

$$^{fm(b+1)}\varpi_{\wp_{sclFea(n)}} = \begin{cases} ^{fm(b)}\varpi_{\wp_{sclFea(n)}} + \frac{g_2(^{m(b)}ps_{\wp_{sclFea(n)}} + ^{fm(b)}ps_{\wp_{sclFea(n)}})}{\exp(\varepsilon c_{m,fm}^2)} if\,(fit(^{fm}ps_{\wp_{sclFea(n)}} > {}^{m}ps_{\wp_{sclFea(n)}}) \\ ^{fm(b)}\varpi_{\wp_{sclFea(n)}} + rw * ranif\,(fit(^{fm}ps_{\wp_{sclFea(n)}} \leq {}^{m}ps_{\wp_{sclFea(n)}}) \end{cases} \tag{12}$$

where, rw is the random walk coefficient, $c_{m,fm}$ is the distance between the male and female mayflies. Based on the fitness function one male mayfly and one female mayfly are selected for the mating process which can be represented by the crossover operation. As a result, two offspring are generated from the crossover operation.

$$of\,sp(1) = ran * {}^{m}\wp_{sclFea(n)} + (1 - ran)^{fm}\wp_{sclFea(n)} \tag{13}$$

$$of\,sp(2) = ran * {}^{fm}\wp_{sclFea(n)} + (1 - ran)^{m}\wp_{sclFea(n)} \tag{14}$$

where, $^{m}\wp_{sclFea(n)}$ is the male parent, $^{fm}\wp_{sclFea(n)}$ is the female parent, $of\,sp(1)$, $of\,sp(2)$ are the generated offspring. The initial velocities of offspring are initialized as zero and the worst mayflies are replaced with the best mayflies. This process is continued until the stop criteria are met and finally, the best mayflies are returned. The pseudo-code of the proposed CC-MA algorithm is shown in Algorithm 1,

Algorithm 1: Feature selection using the CC-MA algorithm

```
Input: Extracted features $\wp_{sclFea(n)}$
Output: selected features $\wp_{uv}^{*}$
Begin
    Initialize population $^{m}\wp_{sclFea(n)}$, $^{fm}\wp_{sclFea(n)}$, velocities $^{m}\varpi_{\wp_{sclFea(n)}}$, $^{fm}\varpi_{\wp_{sclFea(n)}}$, maximum number of iteration $b_{max}$
    Evaluate fitness for each solution
    Set $b = 0$
    While $(b \leq b_{max})$ do
        Update solution of male mayflies by $^{m(b+1)}ps_{\wp_{sclFea(n)}}$ and with the help of correlation coefficient $c$
        Update velocities of male mayflies $^{m(b+1)}\varpi_{\wp_{sclFea(n)}}$
        Update solution of female mayflies using $^{fm(b+1)}ps_{\wp_{sclFea(n)}}$, and correlation coefficient $c$
            $c = \frac{Cov\left(^{m(b)}ps_{\wp_{sclFea(n)}}, {}^{m(pbest,gbest)}ps_{\wp_{sclFea(n)}}\right)}{\delta_{^{m(b)}ps_{\wp_{sclFea(n)}}}\delta_{^{m(pbest,gbest)}ps_{\wp_{sclFea(n)}}}}$
        Update velocities of female mayflies $^{fm(b+1)}\varpi_{\wp_{sclFea(n)}}$
        Evaluate fitness of the new position
            If $(fit(^{m(b+1)}ps_{\wp_{sclFea(n)}}) < fit(^{m(pbest)}ps_{\wp_{sclFea(n)}}))$
            {
                $^{m(pbest)}ps_{\wp_{sclFea(n)}} = {}^{m(b+1)}ps_{\wp_{sclFea(n)}}$
            }
            Else
            {
```

(Continued)

Algorithm 1 (continued)

```
                Keep the same solution
            }
        End if
        Rank the mayflies
        Mate the mayflies
        Perform CC
        Evaluate generated mayflies
        Replace the worst solutions with best solutions
        Set $b = b + 1$
        Return $\wp^*_{uv}$
    End while
End
```

As the best mayflies are selected, the best features are selected by using the CC-MA algorithm. Thus, the selected features are used to form a feature matrix which is for classification. The feature matrix of the selected features is as follows:

$$\wp^*_{uv} = \begin{bmatrix} \wp^*_{1\times1} & \wp^*_{1\times2} & \cdots & \wp^*_{1\times v} \\ \wp^*_{2\times1} & \wp^*_{2\times1} & \cdots & \wp^*_{2\times v} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_{u\times1} & \alpha_{u\times2} & \cdots & \wp^*_{u\times v} \end{bmatrix} \tag{15}$$

where, $\wp^*_{uv}$ denotes the selected features for classification.

3.5 Classification

The classification of attacked and non-attacked data is carried out by using the Logistic Tanh-based Convolutional Neural Network Classification (LTH-CNN) Algorithm. A convolution neural network is a multi-layer neural network containing multiple hidden layers between the input and output layers. Each layer consists of a convolution layer, activation layer, and pooling layer, and the output of the final convolution layer is fed to the fully connected layer. Finally, the output layer containing the softmax layer is used to get the output as the class labels. Here to improve the classification accuracy, the modified activation function called Logistic Tanh is included in the conventional Convolutional Neural Network Algorithm.

The proposed LTH-CNN works in two phases i.e., training and testing. The feature matrix is constructed by using the selected features $\wp^*_{uv}$ is fed to the network for training. The steps are as follows,

In the convolution layer, the input feature map obtained from the previous layer is convoluted with the convolution kernels (i.e., weights), and the output feature map is obtained through an activation function. It can be expressed as:

$$FM_{mn} = \mathfrak{R}_{af}\left(\sum_{x=1}^{X}\sum_{y=1}^{Y} \wp^*_{m-x,n-y} Wt_{xy} + \vartheta\right) \tag{16}$$

where, the output feature map FM_{mn} at the pixel coordinate of (m, n) is calculated by convoluting the kernel Wt_{xy} of size $X \times Y$ with input features in the region of $X \times Y$ same as the kernel, $\Re_{af}$ is the activation function, and ϑ denotes the bias value. The activation function can be expressed as:

$$\Re_{af}(X) = \frac{exp(-X) - 1}{1 + exp(-X)} Where, X = \sum_{x=1}^{X} \sum_{y=1}^{Y} \wp^{*}_{m-x,n-y} Wt_{xy} + \vartheta \tag{17}$$

This process is continued until all the regions of the input features starting at the pixel coordinate $(m - X, n - Y)$ and ending at the pixel coordinates (m, n) are scanned.

After that, the convolution layer's feature map is mapped to the pooling layer. The pooling layer aims to downsample the feature size with the max-pooling function. The output of the pooling layer is obtained as:

$$FM_{pool} = \partial mn_{max} \tag{18}$$

where, ∂_{max} is max pooling function reduces the dimension of the feature map by selecting the maximum value from the small patch of the feature map, FM_{pool} is the output of the pooling layer.

The fully connected layer is then employed, which receives input from the previous layer in a flattened manner. A softmax layer with a given loss function is offered for classification after the fully connected layer. The softmax layer produces the output of the network as the probability distribution indicating the likelihood of each class to which the object belongs using the softmax activation function as:

$$\partial_{sft}(FM_{flat}) = \frac{exp(FM_i)}{\sum_{j=1}^{k} exp(FM_j)} \tag{19}$$

where, ∂_{sft} denotes the activation function, FM_i denotes each element of flattened feature vector FM_{flat}, the term $\sum_{j=1}^{k} \exp(FM_j)$ is the normalization term constituting the probability distribution, k is the number of classes.

After training the real-time sensor data and the testing data are given to the trained system for testing. During testing, the steps as performed at the time of training are repeated for the test data except for the step data grouping. The proposed LTH-CNN accurately classifies the data into attacked data D_{att} and non-attacked data $D_{non-att}$. From the classifier output, the attacked data is stored in the log file for future reference on the other hand; the non-attacked data is preceded by the Cyber Security and Data Security phases. The time complexity of an LTH-CNN is typically proportional to the number of floating-point operations (FLOPs) required for each layer's forward pass, multiplied by the number of layers in the network. Assuming that the input size is n, and the network has L layers, the time complexity can be expressed as $O(FLOPs \times L)$. The space complexity of an LTH-CNN is determined by the amount of memory required to store the model's parameters and intermediate activations during the forward and backward passes. Assuming that the input size is n, and the network has L layers, the space complexity can be expressed as $O(L \times M)$, where M is the maximum number of activations that need to be stored at any given time during the forward and backward passes.

3.6 Cyber Security

In this phase, the packet's source and destination IP address is concealed by using a Complex Binary Format named 1's Compliment Reverse Shift Right (CRSR). Here the source and destination IP address are separated into '4' sections at every dot. Next, the binary conversion of those split four values is performed. Then, the 1's complement is taken. After that complement numbers are reversed and performed the shift right operation for securing the IP address. The 1's complement of the binarized IP address can be expressed as:

$$\Theta_{IP}^{rev} = Rev\left(\overline{\Theta}_{IP}^{Bin}\right) \tag{20}$$

$$\Theta_{IP}^{sec} = \overset{\Rightarrow}{SR}{}_{IP}^{rev} \tag{21}$$

where, $Rev(\cdot)$ represents the reverse function to obtain the reversed data Θ_{IP}^{rev}, $\overline{\Theta}_{IP}^{Bin}$ denotes 1's complement of the binary format, $\overset{\Rightarrow}{SR}(\cdot)$ denotes the shift right operation, Θ_{IP}^{sec} denotes the concealed IP address.

3.7 Data Security

Next, the sensed data are converted into an encrypted format using Senders Public key XOR Receivers Public Key based Elliptic Curve Cryptography (PXORP-ECC) Algorithm. Elliptic Curve Cryptography (ECC) is a key-based technique for secure data transmission that focuses on pairs of public and private keys for encrypting and decrypting the data. It uses elliptic curve mathematics to generate security between key pairs for public-key encryption. The conventional ECC method has a poor design of systems and procedures they are more sensitive to vulnerabilities. To increase the security of the ECC technique, the XOR of the sender and receiver public keys is computed to generate a secret key, which is added to the encryption formula during encryption time and subtracted from the decryption formula during decryption time in the proposed method.

4 Results and Discussion

The proposed method's experimental analysis is carried out in this part. In addition, a performance analysis, as well as a comparison study of the proposed technique, are conducted to determine its efficacy. The suggested system is implemented in the JAVA working platform, with data from the NSL-KDD dataset. The analysis has been done in Python 3.7 and used the Matplotlib library to plot the analysis results. An Apple MacBook Pro with Intel Core i7 Quad-core 2.9 GHz CPU and 16.0 GB RAM has been used to perform all experiments [26]. Although the proposed LTH-CNN has been shown to be effective in various applications still there are several limitations associated with it observed while experiment, some of them are as follows:

Vanishing Gradient Problem: The use of Tanh activation functions in deep neural networks can result in the vanishing gradient problem, where the gradients become ridiculously small as they propagate through the network. This can make it difficult to optimize the network parameters during training and can lead to slower convergence and lower accuracy.

Overfitting: Logistic Tanh-based CNNs can be prone to overfitting, where the model performs well on the training data but poorly on new, unseen data. This can occur when the model is too complex or when the training dataset is too small or not diverse enough.

Sensitivity to Initial Conditions: LTH-CNN can be sensitive to the initial conditions of the model's parameters, which can affect the model's performance and convergence. This can make it difficult to reproduce the same results consistently, especially when training the model multiple times.

4.1 Dataset Description

For the performance analysis, the proposed work gathers the data from the NSL-KDD dataset for the training as well as a testing phase. The NSL-KDD dataset is the extension of the original KDD cup dataset and was downloaded from the University of New Brunswick website. The NSL-KDD dataset contains the Normal data as well as the 22 types of attacked data in four categories as DoS, U2R, R2L, and Probe. The NSL-KDD dataset is a useful benchmark data set for scientists and researchers to compare various intrusion detection systems. The dataset consists of 42 attributes out of which 41 attributes are features and one attribute is a class label as normal traffic or anomaly traffic. The total number of instances in the NSL KDD dataset is 177435 which are divided into training data 123505 and test data 53930. The train data set is used to train the model and the test dataset is given as inputs to the trained model for validation purposes. The advantage of this dataset is that it does not include any redundant records in the training data, so the classifiers are accurate and not biased toward more frequent records.

4.2 Performance Analysis of the Proposed CC-MA

Performance analysis of the proposed feature selection technique called CC-MA is validated with respect to fitness *vs*. iteration and feature selection time, and the final outcomes are compared with various existing techniques, such as Mayfly Optimization Algorithm (MA), *Glow-worm swarm optimization* (*GSO*), cat swarm optimization (*CSO*), and BAT algorithms to state its effectiveness.

From Fig. 2, it can be graphically analyzed and illustrated that the proposed CC-MA algorithm tends to achieve a better fitness outcome for the respective iteration than the existing Correlation Coefficient (CC), GSO, CSO, and BAT algorithms. The better fitness value within a low iteration may help to decrease the time complexity and may also help to a good accuracy without further iteration proceeding process. As per the statement, the proposed CC-MA provides an accurate result with minimal iteration. For instance, the 5th iteration of the proposed technique gains a 6273 fitness value and at the 25th iteration, it gains a 7054 fitness value. But the existing techniques gain an average of 4067 fitness values at the 5th iteration and 5823 fitness values at the 25th iteration. Thus, the comparison clearly stated that the proposed CC-MA achieves the best features within the limited number of iterations.

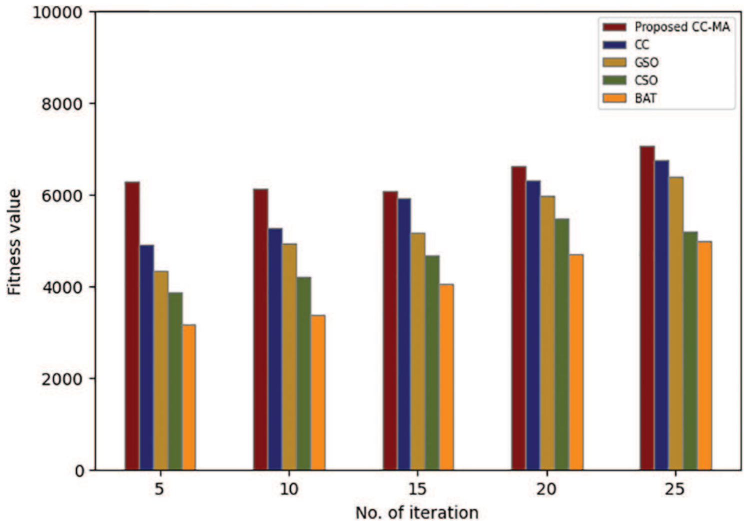

Figure 2: Comparative analysis of the proposed feature selection technique in terms of fitness *vs.* iteration

Table 1 evaluates the proposed CC-MA and the other existing works like CC, GSO, CSO, and BAT with respect to feature selection time. Feature selection time is the amount of time taken by the optimization algorithm to select the optimal features. The proposed CC-MA technique requires 4517 ms to select the ideal features whereas the existing techniques like CC, GSO, CSO, and BAT select the optimal features with the time of 6329, 7682, 8763, and 9524 ms, respectively. From the graphical analysis, it is evident that the proposed method selects the optimal features with limited time. Thus, the proposed method outperforms the other state of art methods and remains to be more reliable.

Table 1: Performance analysis of the proposed CC-MA with respect to feature selection time

S. No.	Techniques	Feature selection time
1	Proposed CC-MA	4517
2	CC	6329
3	GSO	7682
4	CSO	8763
5	BAT	9524

4.3 Performance Analysis of the Proposed LTH-CNN

The proposed LTH-CNN is compared to existing techniques such as Convolutional Neural Network (CNN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), and Deep Belief Network (DBN) in terms of sensitivity, specificity, accuracy, precision, recall, and F-measure (DBN). The comparative analysis is also done with the existing techniques in order to state the effectiveness of the model.

Table 2 and Fig. 3 demonstrate the performance analysis of the proposed LTH-CNN with various existing techniques such as CNN, DNN, RNN, and DBN in terms of accuracy, sensitivity, and specificity. From the analysis, it is understood that the proposed technique achieves higher metrics rates, such as 98.25% of accuracy, 97.29% of sensitivity, and 97.46% of specificity. But the existing works obtain an accuracy rate that overall ranges between 87.92%–95.83%, a sensitivity rate that overall ranges between 86.56%–94.51%, and specificity rates that range between 86.43%–95.72%. Hence, the proposed LTH-CNN withstands better performance metrics rates; therefore, the proposed method detects cyber-attacks accurately and preserves the security of the network more efficiently.

Table 2: Performance analysis of proposed LTH-CNN

S. No.	Techniques	Performance metrics (%)		
		Accuracy	Sensitivity	Specificity
1	Proposed LTH-CNN	98.25	97.29	97.46
2	CNN	95.83	94.51	95.72
3	DNN	92.74	92.67	91.52
4	RNN	90.41	90.13	89.2
5	DBN	87.92	86.56	86.43

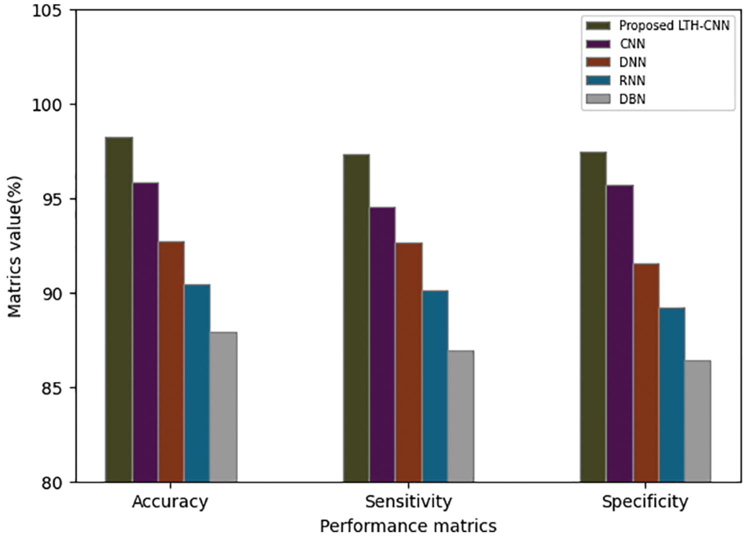

Figure 3: Graphical representation of the proposed LTH-CNN

Fig. 4 compares the precision, recall, and F-measure of the proposed LTH-CNN and the existing works like CNN, DNN, RNN, and DBN. The worthiness of the model is determined by the higher rate of precision, recall, and F-measure. As per the statement, the proposed method achieves 97.63% of precision, 96.75% of recall, and 98.54% of F-measure, whereas the existing techniques such as CNN, DNN, RNN, and DBN obtain precision at an average of 91.72%, recall at the average of 90.42%, and F-measure at the average of 90.93%. This is low as compared to the proposed work. Thus, the proposed LTH-CNN mitigates various complications and renders more prominent results under various complex circumstances.

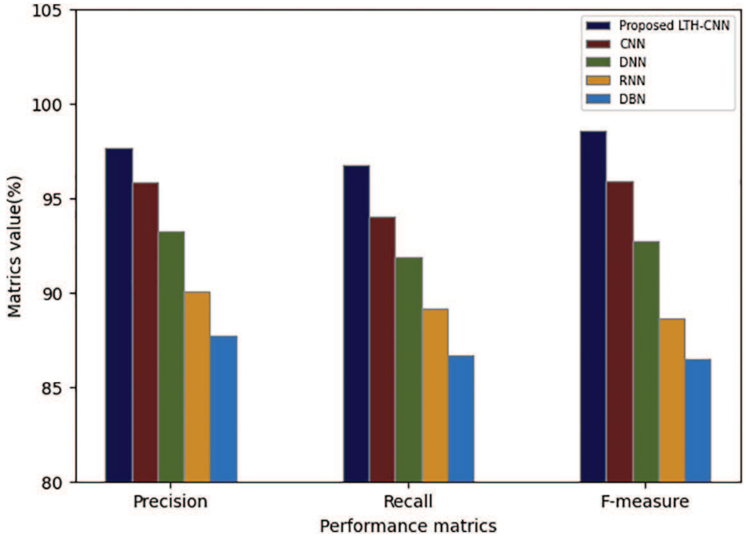

Figure 4: Graphical representation of the proposed LTH-CNN based on Precision, Recall, and F-Measure

Fig. 5 compares the training time of the proposed LTH-CNN with various existing techniques like CNN, DNN, RNN, and DBN. The training time is nothing, but the amount of time taken by the classifier to train the data. From the comparative study, it is clearly known that the proposed technique takes a minimum amount of training time, such that 71352 ms are taken by the classifier to complete the training process, whereas the existing classifiers like CNN, DNN, RNN, and DBN requires 84821, 88327, 93117, and 98723 ms respectively to complete the training process. Hence, the training time of the existing works is high, which increases the overall time of the entire model, but the proposed method completes the entire task quickly as possible, thereby the time complexity of the work can be alleviated.

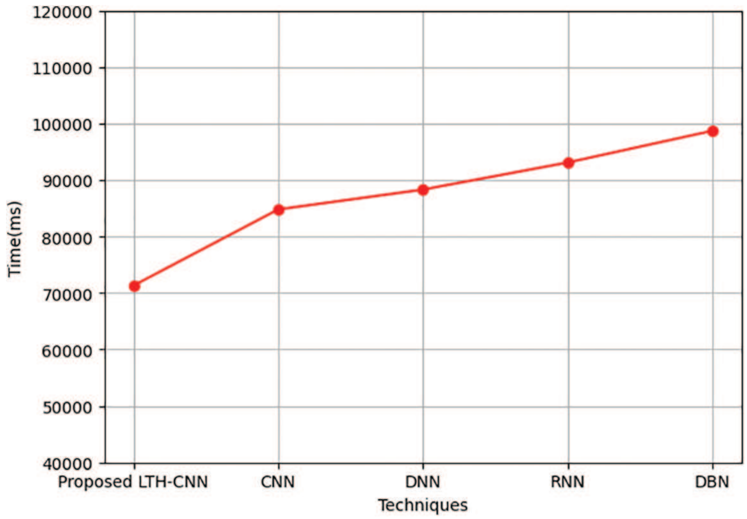

Figure 5: Comparative analysis of proposed LTH-CNN in terms of training time

Table 3 depicts the performance analysis of the proposed LTH-CNN and the existing works like CNN, DNN, RNN, and DBN with respect to memory usage. If the model consumes less amount of memory, then the model is said to be efficient. According to this, the proposed LTH-CNN uses 3512647 kb of memory space, whereas the existing works like CNN, DNN, RNN, and DBN use the memory of 4728375, 5462682, 6681924, and 7372481 kb, respectively. Thus, the memory consumption rate of the proposed method is low as compared to that of existing works. Hence, this shows that the proposed system has noteworthy efficiency in the attack detection system and ensures the security of the network.

Table 3: Performance analysis of proposed LTH-CNN with respect to memory usage

S. No.	Techniques	Memory usage
1	Proposed LTH-CNN	3512647
2	CNN	4728375
3	DNN	5462682
4	RNN	6681924
5	DBN	7372481

4.4 Performance Analysis of the Proposed PXORP-ECC

The proposed PXORP-ECC is evaluated in terms of various performance metrics, such as encryption time, decryption time, security level, and memory usage on encryption and decryption, and the results are compared with various existing works like Elliptic Curve Cryptography (ECC), Rivest Shamir Adleman (RSA), and Elgamal in order to state the worthiness of the model.

Fig. 6 compares the encryption and decryption time of the proposed PXORP-ECC and the existing techniques like ECC, RSA, and Elgamal. The robustness of the model is determined by the limited amount of encryption and decryption time. According to this statement, the proposed method encrypts and decrypts the data at the time of 32128 and 35176 ms, respectively. But the average encryption and average decryption time of the existing works like ECC, RSA, and Elgamal are 45181, and 46228 ms, respectively. Thus, the proposed framework ensures the authenticity of the network with limited time and cost.

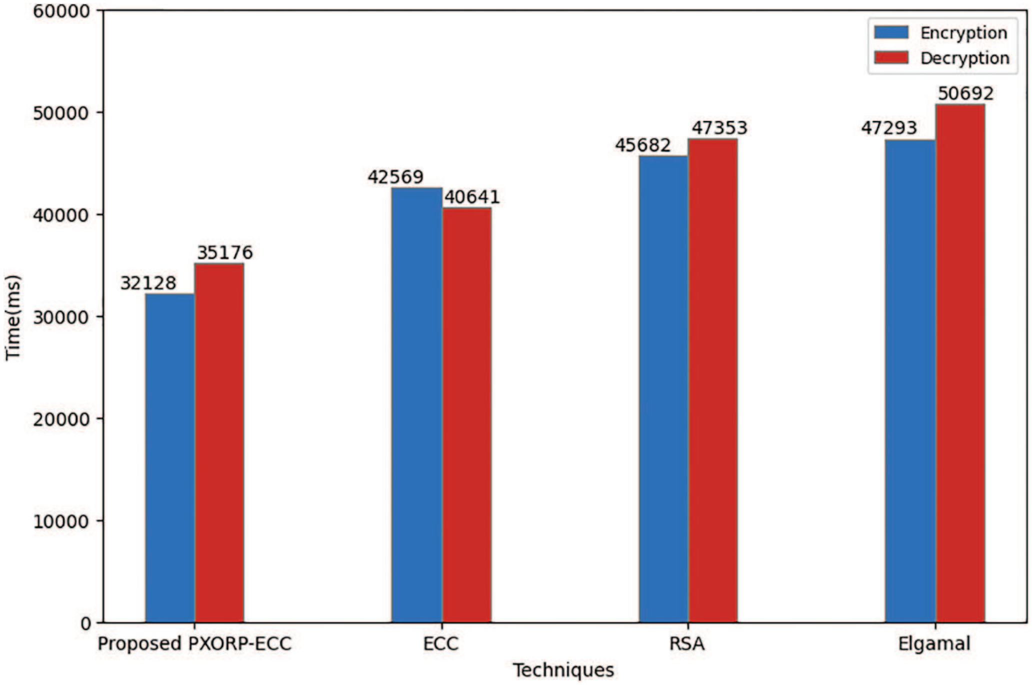

Figure 6: Comparative analysis of the proposed PXORP-ECC in terms of encryption and decryption time

Fig. 7 compares the security rates achieved by the proposed PXORP-ECC method and the existing works like ECC, RSA, and Elgamal. From the graphical analysis, it is known that the proposed method accomplishes a high-security rate of 97%. But the existing works exhibit a security level of an average of 88.33%. This is relatively low when compared to the proposed work. Hence it is concluded that the proposed work efficiently performs the high-secure level encryption and decryption process and mitigates the external attack. Thus, the proposed PXORP-ECC safeguards the network against intruders.

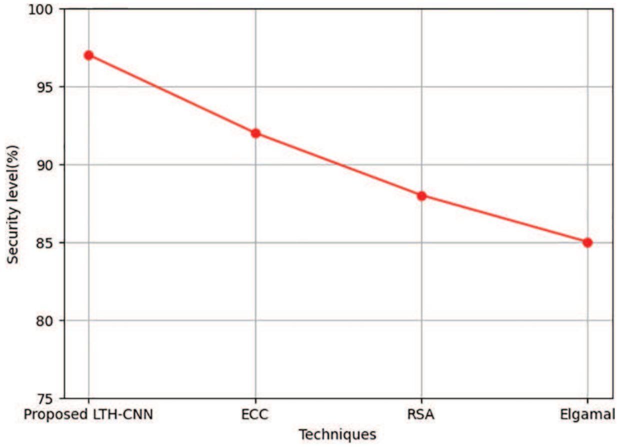

Figure 7: Comparative analysis of the proposed PXORP-ECC in terms of security level

Table 4 evaluates the performance of the proposed PXORP-ECC and the existing works like ECC, RSA, and Elgamal with respect to the memory usage of the encryption and decryption process. The efficiency of the model is determined by the less consumption of memory for the encryption and decryption process. According to this, the proposed PXORP-ECC uses 6326269 kb of memory usage on encryption and 6483954 kb of memory usage on decryption whereas the existing works like ECC, RSA, and Elgamal use an average of 7479624, and 5685632 kb memory on encryption and decryption process, respectively. Thus, the memory consumption rate of the proposed method is low as compared to that of existing works. Hence, this shows that the proposed system has noteworthy efficiency in the privacy-preserving framework.

Table 4: Performance analysis of the proposed PXORP-ECC with respect to memory usage on encryption and decryption

S. No.	Techniques	Memory usage on encryption	Memory usage on decryption
1	Proposed PXORP-ECC	6326269	6483954
2	ECC	6927631	726149
3	RSA	7492584	7983351
4	Elgamal	8018657	8347397

5 Conclusion

The work has proposed an efficient cyber security system and intrusion detection using CRSR with PXORP-ECC and LTH-CNN. The proposed methodology undergoes various steps and if the data is attacked, then it is stored in a log file. If the data is non-attacked, then further steps like cyber security and data security are proceeded to ensure the security. The experimentation analysis is then carried out, which includes a performance analysis and a comparative study of the offered methodologies in terms of some performance metrics to validate the proposed algorithm's effectiveness. For the analysis, the suggested method obtains 98.25% accuracy, 97.29% sensitivity, and 97.46% specificity using publicly available datasets named the NSL-KDD dataset. Furthermore, the approach ensures an elevated level of security, with a 97% success rate. Overall, the suggested framework outperforms current IDS approaches while also remaining more reliable and robust. The work will be expanded with advanced neural networks in the future, with a focus on preserving the privacy of complex datasets and evaluating the performance of the system with other datasets than NSL-KDD. While the proposed PXORP-ECC algorithm is used to encrypt the sensed data, there is still room for improvement in terms of the level of security and efficiency of the approach. Future research can focus on developing more advanced encryption techniques to ensure that the data is secure from any cyber-attacks.

Acknowledgement: The author would like to express deepest gratitude to the Deanship of Scientific Research and Department of Computer Science, Shaqra University, Saudi Arabia for providing research facilities to conduct the research experiments.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declares that she has no conflicts of interest regarding the present study.

References

[1] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Akhtar *et al.,* "A hybrid dl-based detection mechanism for cyber threats in secure networks," *IEEE Access*, vol. 4, pp. 73938–73947, 2021.

[2] G. Pu, L. Wang, J. Shen and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2021.

[3] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 66–79, 2019.

[4] J. Lee, J. Kim, I. Kim and K. Han, "Cyber threat detection based on artificial neural networks using event profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019.

[5] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat *et al.,* "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[6] H. Sadreazami, A. Mohammadi, A. Asif and K. N. Plataniotis, "Distributed graph-based statistical approach for intrusion detection in cyber-physical systems," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 137–147, 2017.

[7] R. Mitchell and R. Chen, "Effect of intrusion detection and response on reliability of cyber physical systems," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 199–210, 2013.

[8] K. D. Lu, G. Q. Zeng, X. Luo, J. Weng, W. Luo *et al.,* "Evolutionary deep belief network for cyber-attack detection in industrial automation and control system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7618–7627, 2021.

[9] R. Gifty, R. Bharathi and P. Krishnakumar, "Privacy and security of big data in cyber physical systems using Weibull distribution-based intrusion detection," *Neural Computing and Applications*, vol. 31, no. 4, pp. 23–34, 2018.

[10] C. Kim, M. Jang, S. Seo, K. Park and P. Kang, "Intrusion detection based on sequential information preserving log embedding methods and anomaly detection algorithms," *IEEE Access*, vol. 9, pp. 58088–58101, 2021.

[11] P. F. de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. G. Santos, D. Macedo *et al.,* "Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247–6256, 2020.

[12] S. Han, M. Xie, H. H. Chen and Y. Ling, "Intrusion detection in cyber-physical systems techniques and challenges," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1049–1059, 2014.

[13] P. V. S. Alpano, J. R. I. Pedrasa and R. Atienza, "Multilayer perceptron with binary weights and activations for intrusion detection of cyber-physical systems," in *Proc. IEEE Region 10 Conf. (TENCON)*, Penang, Malaysia, pp. 2825–2829, 2017.

[14] C. Fang, Y. Qi, P. Cheng and W. X. Zheng, "Optimal periodic watermarking schedule for replay attack detection in cyber-physical systems," *Automatica*, vol. 112, pp. 108698, 2020.

[15] J. Zhang, M. Zulkernine and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[16] R. Fu, X. Huang, Y. Xue, Y. Wu and Y. Tang, "Security assessment for cyber physical distribution power system under intrusion attacks," *IEEE Access*, vol. 7, pp. 75615–75628, 2018.

[17] C. C. Chan, C. Z. Yang and C. F. Fan, "Security verification for cyber-physical systems using model checking," *IEEE Access*, vol. 9, pp. 75169–75186, 2021.

[18] A. N. Jahromi, H. Karimipour, A. Dehghantanha and K. K. R. Choo, "Toward detection and attribution of cyber-attacks in IoT-enabled cyber-physical systems," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13712–13722, 2021.

[19] L. Haghnegahdar and Y. Wang, "A whale optimization algorithm-trained artificial neural network for smart grid cyber intrusion detection," *Neural Computing and Applications*, vol. 32, no. 1, pp. 9427–9441, 2019.

[20] M. Al-Omari, M. Rawashdeh, F. Qutaishat, M. Alshira'H and N. Ababneh, "An intelligent tree-based intrusion detection model for cyber security," *Journal of Network and Systems Management*, vol. 29, no. 2, pp. 1–18, 2021.

[21] A. A. Suzen, "Developing a multi-level intrusion detection system using hybrid-DBN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 1913–1923, 2020.

[22] P. Kumar, A. A. Kumar, C. Sahayakingsly and A. Udayakumar, "Analysis of intrusion detection in cyber-attacks using DEEP learning neural networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2565–2584, 2020.

[23] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain and A. Nawaz, "HML-IDS a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, vol. 7, pp. 89507–89521, 2017.

[24] M. Shoab and S. A. Jubayrin, "Intelligent neighbor selection for efficient query routing in unstructured P2P network using Q-learning," *Applied Intelligence*, vol. 52, no. 6, pp. 6306–6315, 2022.

[25] M. Imran, T. Arif and M. Shoab, "A statistical and theoretical analysis of cyberthreats and its impact on industries," *International Journal of Scientific Research in Computer Science Applications and Management Studies*, vol. 7, no. 5, pp. 1–7, 2018.

[26] M. Shoab and A. S. Alotaibi, "Deep Q-Learning based optimal query routing approach for unstructured P2P Network," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 5765–8581, 2022.