Optimized Deep Neural Network for Attack Detection in Cyber-Physical Systems for Smart Healthcare using Modified Ant Lion Optimization

Ali Ahmadian[1], Ashok Kumar Yadav[2] and Massimiliano Ferrara[3,*]

[1]Faculty of Engineering and Natural Sciences, Istanbul Okan University, Istanbul, Turkey
[2]Department of Information Technology, Rajkiya Engineeing College Azamgarh, Uttar Pradesh, India
[3]Decisions Lab, Mediterranea University of Reggio Calabria, Reggio Calabria, Italy

Abstract

The widespread implementation of innovative healthcare systems brings about notable security risks, especially in cyber-physical systems (CPS). Ensuring patient safety and system performance is crucial in CPS, particularly when detecting and preventing attacks. This paper discusses smart healthcare systems and presents a modified deep neural network (DNN) model that can effectively classify various types of attacks on CPS. In addition, we present a modified Ant Lion Optimization (ALO) algorithm that enhances the model's accuracy and reliability when combined with ensemble methods. By incorporating multiple feature selection techniques, the voting-based ensemble selection method improves the ability to detect attacks by leveraging the importance of the rankings of each feature assessed in those approaches. This enhances the recovery of vital data while minimizing the number of characteristics utilized for identification. Our optimized DNN model outperforms traditional approaches regarding real-time attack detection in smart healthcare system networks. From a theoretical standpoint, the methods outlined in the paper have the potential to enhance the security measures implemented in the construction of CPS and significantly bolster the resilience of smart healthcare systems against the latest cyber threats. The optimized DNN, which was further optimized with the help of the modified ALO algorithm, returned excellent results, with a carpet accuracy of 99.5%, a precision of 99.3%, a recall of 99.4%, an F1-score of 99.35%, and an ROCAUC of 0.995. Such metrics illustrate the model's effectiveness in detecting and classifying different cyberattack forms with a high accuracy rate.

Keywords

Smart Healthcare, Cyber-Physical Systems (CPS), Deep Neural Networks (DNN), Attack Detection, Modified Ant Lion Optimization (ALO), Ensemble Feature Selection.

1. Introduction

Integrating cyber-physical systems (CPS) into competent healthcare has improved the effectiveness of patient monitoring, treatment, and care delivery. However, these advances could result in significant security vulnerabilities. Such systems can become vulnerable to attacks, which could jeopardize the patient's safety and the system's overall functionality. The urgent need to protect healthcare systems from unauthorized intrusion, data and services assault, and illicit content embedding is becoming increasingly apparent [1]. Using computer-based algorithms, CPS facilitates the seamless merging of the digital and physical domains. A CPS ensures that a process is well-managed and regulated [2]. The CPS is built to resist several types of data attacks, such as man-in-the-middle attacks, medical data manipulation, and ransomware attacks like WannaCry. By utilizing the blockchain to store medical data and employing advanced techniques such as convolutional neural networks for analysis, this system can potentially improve the privacy and security of this data [3]. CPS aims to combine physical methods with data processing and communications. Interdependent computational entities interacting with the cosmos and its processes make up a CPS [4], [5].

2nd Workshop "New frontiers in Big Data and Artificial Intelligence" (BDAI 2025), May 29-30, 2025, Aosta, Italy
*Corresponding author.
ahmadian.hosseini@gmail.com (A. Ahmadian); ashok@gecazamgarh.ac.in (A. K. Yadav); massimiliano.ferrara@unirc.it (M. Ferrara)

This work enhances DNN models for CPS healthcare security using a modified ALO algorithm and ensemble feature selection. The ALO algorithm optimizes hyperparameters, while feature selection removes redundancies. Key contributions include ALO's role in security-focused DNN optimization and showing improved attack detection. Experimental results validate its real-time effectiveness.

2. Related Works

Deep learning (DL) consistently outperforms traditional machine learning techniques, as demonstrated in studies such as [6], [7],[8]. Given sufficient data, DL models typically deliver superior results [9]. However, the use of DL models to address the CPS information security problem has been very gradual compared to their application in other areas such as natural language processing, software fragility, and image processing [10], [11]. Furthermore, other DL models have been suggested in recent articles to identify CPS cyberattacks. It is often believed that the difficulties in superimposing privacy and security on top of CPSs are to blame for the difficulty in detecting cyberattacks on these systems. The authors detail their work on a tree classifier-based model for detecting network intrusions in a paper cited as [12]. Achieving an accuracy of 94.23%, the system aims to accelerate anomaly detection by reducing the dimensionality of incoming data. In the context of the Internet of Medical Things (IoMT), malicious actors could jeopardize patient safety by remotely altering device configurations. To address such threats, SMDAps, a specification-based misbehavior detection system, has been proposed [13]. To detect assaults on personal medical devices, the authors[14] have proposed an intrusion detection system (IDS) called HEKA. Using the SVM classifier, HEKA can identify assaults on personal medical supplies with an accuracy of 98.4% and an F1 score of 98%. They reportedly built an intrusion detection system using the KDDCup-'99 dataset. The system used a combination of different classifiers to predict network attacks and applied principal component analysis (PCA) to simplify the data, as mentioned in [15]. Using the bagging algorithm's categorized decision trees, the system had a 93.2% accuracy rate. The Dew-Cloud-based model, which incorporates an organizational long-term memory (HLSTM) model, is a hierarchical federated learning (HFL) system that the researchers suggested in their study [16]. Cyberattacks have developed into an asymmetrical kind of warfare, which is worrisome for computer scientists and the world at large [17], [18]. Research by [19] Data analysis factors like accuracy, speed, delay, ability to handle errors, amount of data, growth potential, convergence, and overall performance guided the suggestion of using feed-forward and feedback propagation ANN models for research. To mentally load and fool the adversary ([20]), provide a cognitive deception model (CDM) based on a neural network. The CDM takes an input message and produces decoy messages that are separate, believable, persuasive, and syntactically and semantically coherent. Their approach centered on investigating the performance of various techniques across diverse datasets with varying characteristics and determining the optimal parameters for these algorithms to function effectively. In a study conducted by researchers [21], they utilized a recurrent neural network (RNN)-based AE. They employed data segmentation and aggregation techniques to enhance the model's performance, creating segments of varying lengths while maintaining a similar total variation. Similarly, [22] utilized RNN AEs to effectively reconstruct multi-dimensional time series data, giving researchers helpful information about the operating state of specific sensors in the system without requiring complete reconstruction.

3. Methods and Materials

In this study, we develop an optimized DNN model to detect different cyberattacks on intelligent healthcare CPS. We have utilized the TON_IoT dataset, which includes a range of attacks, and addressed the challenges of data imbalance and selection through a voting-based ensemble approach. In addition, the ALO algorithm, modified explicitly for ant lion optimization, is employed to optimize the hyper-parameter of the DNN to attain the utmost detection rate for different attack scenarios. The working flow diagram is shown in Figure 1. The selected features are assessed and selected using techniques such as mutual information, lasso regression, and chi-square tests before they are integrated using a

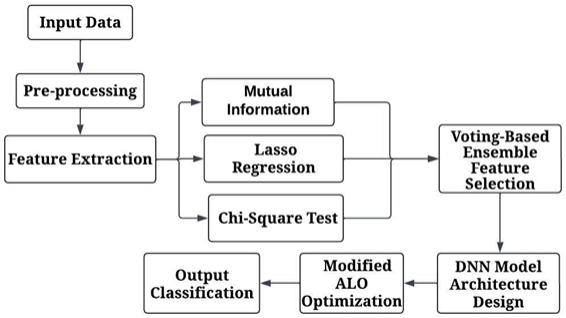

Figure 1: Overall Processing Flow Diagram of Attack Detection in Cyber-Physical Systems for Smart Healthcare

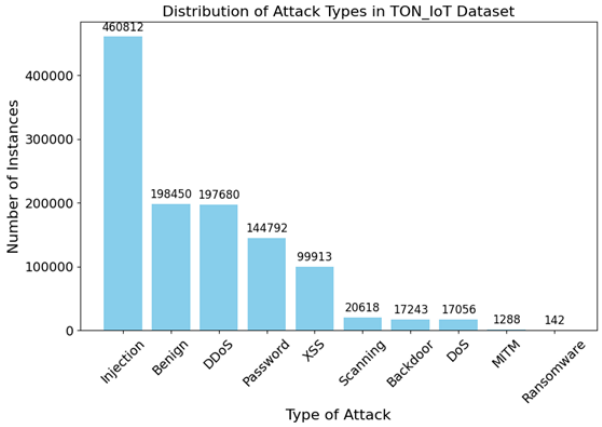

Figure 2: Data Distribution from the TON_IoT Dataset

voting-based ensemble feature selection. The extracted features also guide the DNN model architecture design, which the modified ALO algorithm enhances. The last step involves classifying input data into normal and attack classes using the optimized deep neural network, further improving the security of smart healthcare systems.

3.1. Data Collection

The study uses the TON_IoT dataset, which is highly suitable for cybersecurity research in Internet of Things (IoT) environments and includes applications in intelligent healthcare systems. The dataset consists of various attack types, making it a solid basis for training and evaluating the proposed DNN model. The dataset contains a wide range of instances, including injection attacks, benign traffic, Distributed Denial of Service (DDoS) attacks, password attacks, scanning attacks, Cross-Site Scripting (XSS) attacks, backdoor attacks, Man-in-the-Middle (MITM) attacks, Denial of Service (DoS) attacks, and ransomware attacks. Figure 2 shows the data distribution from the TON_IoT dataset. In addition, the data underwent preprocessing, including normalization and encoding of categorical features. It was then split into a training set, which accounted for 70% of the data, and a testing set, which accounted for the remaining 30%. The processing methodology in this study consists of several basic steps: data normalization, feature extraction, and data splitting. The normalization of a feature x_i is performed using the following equation:

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Here, x_i represents the original feature value, x_{min} and x_{max} denote the minimum and maximum values of that feature in the dataset, respectively, and x_i' is the resulting normalized value.

3.2. Feature Selection

Extraction of features entails detecting and extracting specific essential characteristics found in the raw information to differentiate between the normal and attack states. Information concerning the features is obtained from network traffic, sensor data, and system event logs. In this case, an ensemble feature selection method is adopted, where the voting technique is utilized to rank the features while several feature selection methods are employed. The DNN then receives the selected features.

3.2.1. Mutual Information

The degree to which each characteristic depends on the target variable may be determined via mutual information. Features are deemed more relevant to the job if they have excellent mutual knowledge of the goal. Mathematically, mutual information between a feature x_i and the target variable y is calculated as

$$I(x_i; y) = \sum_{x_i, y} p(x_i, y) \log \left(\frac{p(x_i, y)}{p(x_i)p(y)} \right) \tag{2}$$

Where $p(x_i, y)$ is the joint probability distribution of x_i and y, and $p(x_i)$ and $p(y)$ are the marginal probability distributions of x_i and y, respectively. Features with the highest mutual information scores are selected for further analysis.

3.2.2. Lasso Regression (L1 Regularization)

By applying an L1 penalty to the coefficients, the linear model known as Lasso Regression selects features. Some coefficients become zero due to this penalty, eliminating aspects that aren't crucial to the model. The Lasso objective function is:

$$Minimize \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{m} |\omega_j| \right\} \tag{3}$$

Where ω_j is the coefficients, λ is the regularization parameter, and n and m represent the number of samples and features, respectively. Features with non-zero coefficients are selected as they contribute to predicting the target variable.

3.2.3. Chi-Square Test

To determine if categorical traits are independent of the dependent variable, statisticians employ the chi-square test. In terms of categorical variables, it quantifies the discordance between actual and predicted frequencies. The Chi-square statistic for a feature x_i concerning the target y is calculated as:

$$\chi^2(x_i, y) = \sum_{k} \frac{(O_k - E_k)^2}{E_k} \tag{4}$$

where O_k is the observed frequency and E_k is the expected frequency under the independence assumption. Features with the highest Chi-Square scores are considered the most relevant.

3.2.4. Voting-Based Ensemble Feature Selection Method

The voting-based ensemble method summarizes the different feature selection outcomes and credits repeated feature selections in the various techniques. This avoids the problem of bias resulting from excessive dependence on one method of selecting the best features. The dataset X, consisting of m features, is represented as:$X = \{x_1, x_2, \ldots, x_m\}$. We apply n different feature selection methods to this dataset, resulting in n subsets of selected features $S_1, S_2, \ldots, S_n$, where $S_j \subseteq X$ is the subset of features selected by the j-th feature selection method. For each feature x_i in the dataset X, a vote

is assigned based on whether it was selected using a particular feature selection method. The total number of votes for the feature x_i across all methods is calculated as

$$V(x_i) = \sum_{j=1}^{n} \mathbf{1}(x_i \in S_j) \tag{5}$$

where $\mathbf{1}(x_i \in S_j)$ is an indicator function that equals 1 if x_i is present in S_j, and 0 otherwise. A threshold k is set to determine the minimum number of votes required for a feature to be included in the final selected feature set F. The final selected feature set F is given by $F = \{x_i \in X \mid \text{votes}(x_i) \geq k\}$ Where, k is the minimum number of votes a feature must receive to be considered essential and included in the final feature set. The final feature set F consists of features that have received at least k votes, reflecting a consensus among the various feature selection methods. The choice of k can be adjusted depending on the desired strictness of feature selection. For example, setting $k = n$ would require a feature to be selected by all methods, while $k = \frac{n}{2}$ would require selection by at least half of the methods.

Algorithm: Voting-Based Ensemble Feature Selection
Input: Dataset X with m features $\{x_1, x_2, \ldots, x_m\}$ and target variable y
Output: Selected feature set F

1. Initialize empty lists for selected features: $S_{MI}, S_{Lasso}, S_{ChiSquare}$
2. Set voting threshold k (e.g., $k = 2$)
3. **Step 1: Feature Selection using Mutual Information**
   a) For each feature x_i in X:
      i. Compute Mutual Information $I(x_i; y)$
   b) Select the top n features based on the highest $I(x_i; y)$ values and add to S_{MI}
4. **Step 2: Feature Selection using Lasso Regression (L1 Regularization)**
   a) Fit Lasso Regression model on X with target y
   b) For each feature x_i in X:
      i. If Lasso coefficient $w_i \neq 0$, add x_i to S_{Lasso}
5. **Step 3: Feature Selection using Chi-Square Test**
   a) For each feature x_i in X:
      i. Compute Chi-Square statistic $\chi^2(x_i, y)$
   b) Select top n features based on highest $\chi^2(x_i, y)$ values and add to $S_{ChiSquare}$
6. **Step 4: Voting Mechanism**
   a) Initialize an empty dictionary VoteCount to store vote counts for each feature
   b) For each feature x_i in X:
      i. VoteCount$[x_i] = 0$
      ii. If $x_i \in S_{MI}$, then VoteCount$[x_i]$ = VoteCount$[x_i] + 1$
      iii. If $x_i \in S_{Lasso}$, then VoteCount$[x_i]$ = VoteCount$[x_i] + 1$
      iv. If $x_i \in S_{ChiSquare}$, then VoteCount$[x_i]$ = VoteCount$[x_i] + 1$
7. **Step 5: Select Final Feature Set** F
   a) Initialize empty set F
   b) For each feature x_i in X:
      i. If VoteCount$[x_i] \geq k$, add x_i to F
   c) Return F

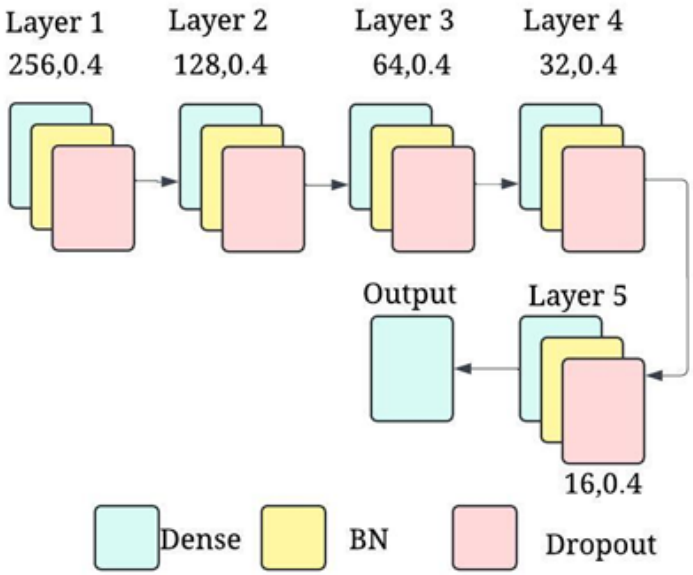

Figure 3: Proposed Deep DNN network

3.3. Model Architecture

The number of features, F, selected in the feature selection process corresponds to the number of input features, which is the input layer. Let $x \in \mathbb{R}^F$ be the vector of the selected input features. The DNN model for CPS intrusion detection proposes an attack. It is embedded in an intelligent health care system of five hidden BUS fully connected dense layers. In each hidden layer l, the input from the previous layer$h_{(l-1)}$ undergoes a linear transformation followed by applying the rectified linear unit (ReLU) activation function. The mathematical operation for the l^{th} hidden layer is given by $h_l = \text{ReLU}(W_l h_{l-1} + b_l)$ where W_l is the weight matrix that connects the neurons of layer l-1 to the neurons of layer l, b_l is the bias vector added to the linear transformation, and ReLU is the activation function defined as $\text{ReLU}(z) = \max(0, z)$. This activation function introduces non-linearity into the model, allowing it to learn more complex functions. The first hidden layer consists of 256 neurons, transforming the input vector $h_0 = x$ is the input features vector) using the weight matrix W_{l1} and bias b_l. The second hidden layer reduces the dimensionality further by using 128 neurons and applying a new set of weights W_2 and biases b_2. The third hidden layer has 64 neurons, continuing to distill the most relevant information through the weight matrix W_3 and bias b_3. The fourth hidden layer comprises 32 neurons, used W_4 to refine the feature representation further. The fifth and final hidden layer contains 16 neurons, producing the final intermediate output h_5 before the model's predictions are computed in the output layer. Batch normalization is used after each hidden layer to standardize the input to each layer to improve training stability and speed. Dropout is another measure used after each hidden layer, which aims to avoid overfitting by training a random percentage of neurons with zero outputs. Stacking these hidden layers enhances the model's capacity to identify different kinds of cyberattacks in smart healthcare CPS by allowing it to learn hierarchical feature representations gradually. Figure 3 shows the proposed deep DNN network. The batch normalization operation for a given layer is defined as:

$$h_l^{\text{norm}} = \frac{h_l - \mu}{\sigma} \cdot \gamma + \beta \tag{6}$$

Where, μ and σ are the mean and standard deviation of the mini-batch. γ and β are learnable parameters that scale and shift the normalized output. To prevent overfitting, dropout layers are applied after each hidden layer. Dropout randomly sets a fraction p of input units to zero during training. The dropout operation is defined as $h_l^{\text{drop}} = h_l \cdot r, \quad \text{where } r \sim \text{Bernoulli}(p)$ The variable r represents a binary mask vector that is drawn from a Bernoulli distribution, which has a probability p of retaining a unit.

The output layer consists of 2 neurons, corresponding to the two classes: 'normal' and 'attack.' The softmax activation function is applied to the output logits to convert them into class probabilities. The mathematical operation for the output layer is $o = \text{Softmax}(W_6 h_5 + b_6)$. Where, W_6 and b_6 are the output layer's weight matrix and bias vector. The softmax function is defined as:

$$\text{Softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)} \tag{7}$$

This function ensures that the outputs are non-negative and sum to 1, representing valid probabilities. The model learns to reduce the categorical cross-entropy loss, which looks at how different the predicted probabilities are from the actual class labels. The cross-entropy loss is defined as:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \tag{8}$$

Here, N denotes the number of training examples, C is the number of classes, $y_{(i,c)}$ represents the true label (1 for the correct class, 0 otherwise), and $\hat{y}_{i,c}$ is the predicted probability for class c for the i-th example.

Algorithm: Modified ALO for DNN Hyperparameter Tuning

Input: Population size P, Maximum number of iterations MaxIter, Search space for hyperparameters and DNN model architecture
Output: Optimal hyperparameters for the DNN

1. Initialize a population of Ant lions with random hyperparameters within the defined search space.
2. Evaluate the fitness of each Ant lion by training the DNN and measuring validation accuracy and loss.
3. Select the best-performing ant lions as elites.
4. **while** (termination criteria not met) **do**
   a) **for** each ant (candidate solution) **do**
      i. Perform a random walk in the hyperparameter space based on the position of the nearest Ant lion.
      ii. Update the ant's position using:

$$X_{\text{ant}}(t+1) = \frac{X_{\text{ant}}(t) + X_{\text{lion}}(t)}{2}$$

      iii. Train the DNN with the current hyperparameters of the ant.
      iv. Evaluate the fitness of the ant using the fitness function:

$$F(X) = \text{Validation Accuracy} - \alpha \times \text{Validation Loss}$$

      v. If the ant's fitness is better than the corresponding Ant lion's fitness, update the Ant lion's position.
   b) **end for**
   c) Apply elitism: retain the best-performing Ant lions as elites.
5. **end while**
6. Return the best set of hyperparameters found.

Table 1
The Performance Analysis of the Proposed Model

Metric	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Injection	99.7	99.6	99.8	99.7	0.997
Benign	99.4	99.2	99.3	99.25	0.994
DDoS	99.6	99.5	99.7	99.6	0.996
Password	99.4	99.2	99.3	99.25	0.994
XSS	99.3	99.1	99.2	99.15	0.993
Scanning	99.2	99	99.1	99.05	0.992
Backdoor	99.1	98.9	99	98.95	0.991
DoS	99.3	99.1	99.2	99.15	0.993
MITM	99	98.8	98.9	98.85	0.99
Ransomware	98.9	98.7	98.8	98.75	0.989
Overall	99.5	99.3	99.4	99.35	0.995

4. Result and Discussion

The evaluation focused on a DNN model enhanced with a modified ALO algorithm, tested on a healthcare CPS dataset containing various attacks. Performance was measured using accuracy, precision, recall, F1-score, and ROC-AUC metrics. Table 1 defines the performance parameters showcased by the deep neural network, or the DNN model, to identify the various forms of cyberattacks in smart healthcare cyber-physical systems. The model delivers appreciable performance in terms of the multiple forms of attacks with the weakness's injection attacks (accuracy of 99.7%, precision of 99.6%, recall of 99.8%, F1 score of 99.7%, and ROC AUC of 0.997) and DDoS attacks (accuracy of 99.6%, precision 99.5%, recalled 99.7%, F1 score of 99.6% and an AUC score of 0.996). The model performs similarly in classifying benign traffic accuracy at 99.4% and ROC AUC of 0.994, but at a lower level of precision and recall than the attack categories. However, while the model does well in terms of detection of XSS, scanning, backdoor, DoS, MITM, and ransomware attacks, the associated performance scores for these categories are less, with accurate values between 98.9% and 99.3%, AUC values nearing 0.99. Judging from these results, it can be inferred that the model is generally efficient, especially when detecting common and more devastating types of attacks such as injections and DDoS attacks. Results show the model performs optimally across most attack types, indicating its practical applicability, particularly in smart healthcare systems. Figure 4 shows the training and testing accuracy of the proposed DNN model optimized with the modified ALO technique, which has been carried out for 100 epochs. The graph implies that improvement and understanding of the model have taken place over time, both in the training data and testing data, since there has been a gradual increase in the accuracy percentages on both datasets. The training accuracy graph is slow to rise in the first few epochs, indicating that the model can harness a fast learning rate from the training set. However, after additional training, the graph begins to resemble a straight line, slightly below the optimal level of 100% accuracy. This means that the model has learned almost every feature of the data. The dashed line in the graph refers to the accuracy obtained during the testing phase, which follows almost the same trend but is lower than the training phase. There is a tiny margin between training and test accuracy. This evidence suggests the model can fit the unseen data reasonably well and does not overfit considerably. The high accuracy for both training and testing data is because the model's hyperparameters have been effectively adjusted using modified ALO for optimizing hyperparameters in detecting harmful cyberattacks in smart healthcare systems. Figure 5 illustrates the loss curves for training and test datasets over 100 epochs. The losses converge as constants remain stable, with an initial sharp dip indicating the model's learning phase. As training progresses, the curves stabilize, signifying optimal performance. Training loss is typically lower than test loss since both datasets share a similar distribution. The small gap between the curves suggests minimal overfitting, demonstrating that the model generalizes well. This balance ensures reliable predictions without excessive bias toward training data. The impact of feature selection on the performance of the

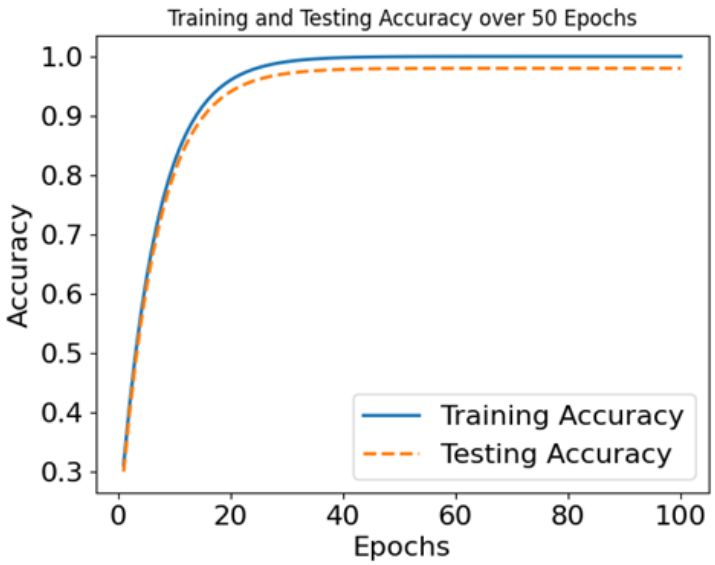

Figure 4: Accuracy of the Proposed Deep DNN network with Modified ALO algorithm

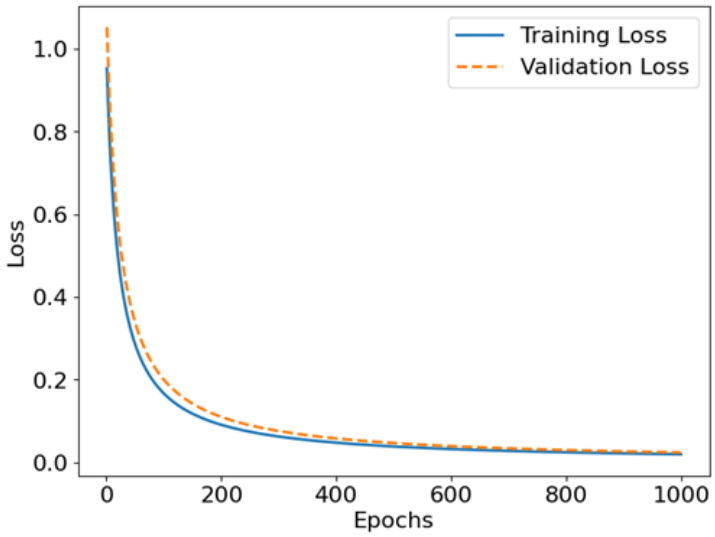

Figure 5: Loss of the Proposed Deep DNN Network with ALO algorithm

proposed DNN model is shown in Table 2. Applying feature selection techniques significantly improved all evaluated metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Accuracy increased from 98.5% to 99.5%, enhancing prediction precision. Similarly, precision rose from 98.2% to 99.3%, while recall improved from 98.3% to 99.4%, demonstrating better target identification. The F1-score, which balances precision and recall, increased from 98.25% to 99.35%. ROC-AUC also showed performance gains, reflecting improved class discrimination. These enhancements highlight the importance of feature selection in optimizing input data, ultimately strengthening the model's accuracy and effectiveness in detecting cyberattacks in smart health CPS. Table 3 presents the results of the modified ALO algorithm on the proposed DNN model performance metrics. The results indicate that using the ALO algorithm enhances the model's performance across all evaluated metrics. For instance, the accuracy of the model terms without ALO is 98.8%, while with ALO, it is 99.5%. This means ALO helps enhance the model's performance in classifying positive and negative classes. The same, the precision in positive predictions made by the model, i.e., true positives, is 98.5% in the absence of the ALO algorithm. In contrast, it is 99.3% in the presence of ALO, with even better performance. This improvement indicates that the

Table 2
Impact of Feature Selection

Metric	Without Feature Selection	With Feature Selection
Accuracy	98.5	99.5
Precision	98.2	99.3
Recall	98.3	99.4
F1-Score	98.25	99.35
ROC-AUC	0.985	0.995

Table 3
Effectiveness of Modified ALO

Metric	Without ALO	With ALO
Accuracy	98.8	99.5
Precision	98.5	99.3
Recall	98.6	99.4
F1-Score	98.55	99.35
ROC-AUC	0.988	0.995

model is better at decreasing false positives with ALO optimization. The F1-score, calculated as the harmonic mean of precision and recall, improves significantly from 98.55% to 99.35% percentile given the application of ALO techniques, emphasizing proportional advancement of precision and recall. Finally, the ROC-AUC score, which helps in understanding all the distinct classes in this data set, increases from 0.988 to 0.995, showing that the model performs better after ALO optimization. This study demonstrates that modified ALO for hyperparameter optimization enhances cyberattack detection in healthcare CPS.

5. Conclusion and Future Work

The proposed approach to detecting cyber threats within smart healthcare CPS demonstrates impressive efficiency by harmonizing advanced feature selection methods with DNN architecture carried out with the modified ALO algorithm. The enhanced optimal DNN, which was refined with the modified ALO algorithm, performed well, as evidenced by an accuracy level of 99.5%, precision of 99.3%, recall of 99.4%, F1-score of 99.35% and ROC-AUC of 0.995. These statistics suggest that the model effectively detects and classifies multiple categories of cyber threats, hence injection attacks, DDoS, and XSS, among many others. The findings provide evidence that the considered strategy effectively increases the safety and reliability of intelligent healthcare CPS, which is essential for accurately and quickly mitigating potential risks. Several avenues for future study might be explored in light of the successes of this work to deepen and expand the suggested technique. One such exciting possibility is determining how the modified ALO is extended to handle more extensive and complex datasets, as in the case of healthcare CPS, which are becoming very common. Expanding the model with real-time data enables continuous monitoring and quicker detection, enhancing the system's agility against cyber threats. Further work could also be conducted by integrating other optimization techniques, such as genetic algorithms, to develop new strategies for hyperparameter tuning. Furthermore, testing this approach in particular domains like industrial IoT or autonomous cars would be valuable since it will help demonstrate and assess its flexibility and generality. Finally, subsequent work can enhance model interpretation and clarify how DNN makes decisions and the strategies carried out to enhance better acceptance of AI-based security control systems, especially in critical areas such as healthcare.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] I. Priyadarshini, R. Kumar, L. M. Tuan, L. H. Son, H. V. Long, R. Sharma, S. Rai, A new enhanced cyber security framework for medical cyber physical systems, SICS Software-Intensive Cyber-Physical Systems (2021) 1–25.

[2] C. V. Kumar, et al., A real time health care cyber attack detection using ensemble classifier, Computers and Electrical Engineering 101 (2022) 108043.

[3] R. Ch, G. Srivastava, Y. L. V. Nagasree, A. Ponugumati, S. Ramachandran, Robust cyber-physical system enabled smart healthcare unit using blockchain technology, Electronics 11 (2022) 3070.

[4] M. Alowaidi, S. K. Sharma, A. AlEnizi, S. Bhardwaj, Integrating artificial intelligence in cyber security for cyber-physical systems., Electronic Research Archive 31 (2023).

[5] S. Izadi, M. Ahmadi, R. Nikbazm, Network traffic classification using convolutional neural network and ant-lion optimization, Computers and Electrical Engineering 101 (2022) 108024.

[6] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, volume 1, MIT press Cambridge, 2016.

[7] P. K. Srivastava, A. K. Yadav, Machine Learning Techniques and Industry Applications, IGI Global, 2024.

[8] H. Huang, P. Wang, J. Pei, J. Wang, S. Alexanian, D. Niyato, Deep learning advancements in anomaly detection: A comprehensive survey, arXiv preprint arXiv:2503.13195 (2025).

[9] P. K. Srivastava, A. K. Yadav, Methodologies, Frameworks, and Applications of Machine Learning, IGI Global, 2024.

[10] G. Lin, S. Wen, Q.-L. Han, J. Zhang, Y. Xiang, Software vulnerability detection using deep neural networks: a survey, Proceedings of the IEEE 108 (2020) 1825–1848.

[11] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, Y. Xiang, Code analysis for intelligent cyber systems: A data-driven approach, Information sciences 524 (2020) 46–58.

[12] K. Gupta, D. K. Sharma, K. D. Gupta, A. Kumar, A tree classifier based network intrusion detection model for internet of medical things, Computers and Electrical Engineering 102 (2022) 108158.

[13] P. V. Astillo, G. Choudhary, D. G. Duguma, J. Kim, I. You, Trmaps: Trust management in specification-based misbehavior detection system for imd-enabled artificial pancreas system, IEEE Journal of Biomedical and Health Informatics 25 (2021) 3763–3775.

[14] A. I. Newaz, A. K. Sikder, L. Babun, A. S. Uluagac, Heka: A novel intrusion detection system for attacks to personal medical devices, in: 2020 IEEE Conference on Communications and Network Security (CNS), IEEE, 2020, pp. 1–9.

[15] T. Saba, Intrusion detection in smart city hospitals using ensemble classifiers, in: 2020 13th International Conference on Developments in eSystems Engineering (DeSE), IEEE, 2020, pp. 418–422.

[16] P. Singh, G. S. Gaba, A. Kaur, M. Hedabou, A. Gurtov, Dew-cloud-based hierarchical federated learning for intrusion detection in iomt, IEEE journal of biomedical and health informatics 27 (2022) 722–731.

[17] O. I. Abiodun, M. Alawida, A. E. Omolara, A. Alabdulatif, Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey, Journal of King Saud University-Computer and Information Sciences 34 (2022) 10217–10245.

[18] A. E. Omolara, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, W. Alshoura, H. Arshad, The internet of things security: A survey encompassing unexplored areas and new insights, Computers & Security 112 (2022) 102494.

[19] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (2018).

[20] O. T. Taofeek, M. Alawida, A. Alabdulatif, A. E. Omolara, O. I. Abiodun, A cognitive deception model for generating fake documents to curb data exfiltration in networks during cyber-attacks, IEEE Access 10 (2022) 41457–41476.
[21] D. Hsu, Time series compression based on adaptive piecewise recurrent autoencoder, arXiv preprint arXiv:1707.07961 (2017).
[22] T. Wong, Z. Luo, Recurrent auto-encoder model for multidimensional time series representation (2018).