Discover Sustainability

Research

An adaptive hybrid framework for IIoT intrusion detection using neural networks and feature optimization using genetic algorithms

Mohammad Zubair Khan[1] · **Aijaz Ahmad Reshi**[2] · **Shabana Shafi**[2] · **Ibrahim Aljubayri**[1]

Received: 23 December 2024 / Accepted: 7 April 2025
Published online: 08 May 2025

Abstract
In Industrial Internet of Things (IIoT) networks, securing device connectivity through effective intrusion detection systems is essential for maintaining operational integrity. This paper presents an adaptive hybrid framework for IIoT intrusion detection that combines Artificial Neural Networks (ANNs) with Genetic Algorithms (GA) for feature optimization. This study utilizes the dataset, which is widely recognized benchmark for intrusion detection research. The dataset comprises 625783 network traffic samples, classified into five categories: Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), User-to-Root (U2R), and Normal traffic. Initially, an ANN model was developed, yielding high accuracy but exhibiting signs of overfitting. To enhance generalization, we introduced L2 regularization, adjusted dropout rates, and optimized the learning rate, ultimately achieving a 99.7% validation accuracy with an AUC score of 0.9969. Additionally, Genetic Algorithms were employed to optimize feature selection, further refining the ANN's input space to improve computational efficiency without sacrificing predictive power. After training over 50 epochs with early stopping, the model demonstrated exceptional robustness, achieving 99.5% accuracy on the test set, with precision and recall values of 0.97 and 0.98, respectively. This combination of ANN and GA yielded a highly efficient and sensitive framework, providing enhanced detection of anomalies in IIoT environments. The proposed hybrid model thus establishes a robust solution for real-time IIoT security, outperforming conventional detection systems through a strategic blend of neural learning and evolutionary optimization.

Keywords IIoT · ANN · Genetic Algorithm · Feature Optimization

1 Introduction

The growing adoption of Industrial Internet of Things (IIoT) technologies across diverse sectors has revolutionized traditional industrial frameworks, significantly improving efficiency and productivity. However, this swift integration has also led to considerable security vulnerabilities, making IIoT systems prime targets for cyberattacks. Intrusion detection is vital in protecting these systems by recognizing unauthorized access and anomalies within their network traffic. Conventional security measures fall short in addressing the dynamic and heterogeneous nature of IIoT environments, underscoring the need for sophisticated techniques capable of adaptingable to evolving threats.

✉ Aijaz Ahmad Reshi, aijazonnet@gmail.com; Mohammad Zubair Khan, mkhanb@taibahu.edu.sa; Shabana Shafi, bhatshabu@gmail.com; Ibrahim Aljubayri, ijubayri@taibahu.edu.sa | [1]Department of Computer Science and Information, Taibah University, Madinah 42353, Saudi Arabia. [2]Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia.

Machine learning methods and neural networks, have been identified as effective approaches for intrusion detection in IIoT contexts. The capability of the models to learn intricate patterns from extensive datasets enables the detection of subtle anomalies that may be overlooked by traditional methods. This research proposes a comprehensive workflow for IIoT intrusion detection that employs a neural network architecture aimed at enhancing detection efficiency.

The proposed methodology includes several preprocessing steps designed to maintain the quality and integrity of the input data. These steps involve handling missing values in both numeric and categorical columns, encoding categorical variables, and balancing the dataset using techniques such as FW-SMOTE. This systematic data preparation establishes a solid foundation for further analysis.

Optimization is essential for improving the performance of the model. Optimization algorithms, namely Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), have been employed to refine the feature selection process [1]. These algorithms not only optimize the model but also aid in reducing classification errors while enhancing accuracy.

The neural network architecture implemented in this study comprises an input layer, two hidden layers, and an output layer, each configured specifically to reduce overfitting and enhance learning efficiency. Increased L2 regularization and dropout techniques have been integrated to strengthen the model's resilience against overfitting, thereby enhancing generalization to new, unseen data.

To assess the effectiveness of the proposed approach, standard performance metrics such as accuracy, precision, and recall have been utilized. These metrics offer a thorough evaluation of the model's ability to detect intrusions both accurately and reliably.

The rapid proliferation of the Industrial Internet of Things (IIoT) has introduced significant cybersecurity challenges, particularly in detecting and mitigating sophisticated cyber threats. Traditional Intrusion Detection Systems (IDS) rely heavily on signature-based and anomaly-based techniques, which often suffer from high false positive rates, poor adaptability to novel attacks, and computational inefficiencies when deployed in real-time IIoT environments. Existing IDS frameworks frequently lack effective feature selection mechanisms, leading to increased processing overhead and suboptimal classification performance. Additionally, deep learning-based IDS approaches, while promising, require optimization techniques to prevent overfitting and improve generalization. One of the key challenges in real-time intrusion detection for IIoT lies in achieving high accuracy while maintaining low latency and computational efficiency. IIoT networks generate vast amounts of data, making it imperative for IDS models to process network traffic efficiently without overwhelming resource-constrained devices. Moreover, evolving attack patterns necessitate adaptive models that can dynamically refine their feature selection and classification strategies.

To address these challenges, this study introduces a novel hybrid framework that integrates Artificial Neural Networks (ANNs) with Genetic Algorithms (GA) for optimized feature selection and performance enhancement. Unlike conventional IDS models that rely on static feature sets, our approach dynamically selects the most relevant features, reducing computational complexity while preserving high detection accuracy. The incorporation of L2 regularization and dropout mechanisms further enhances generalization, mitigating the risk of overfitting. By combining ANN's capability for deep feature learning with GA's optimization potential, our framework offers a scalable and high-performance solution for IIoT security, effectively bridging the gap in existing IDS methodologies.

2 Related work

The exponential growth of IIoT networks has led to an increasing number of potential security vulnerabilities [2], making intrusion detection a significant area of research. Traditional IDS methods, such as signature-based and anomaly-based detection, struggle to effectively secure IoT environments due to their dynamic nature and limited resources [3]. Consequently, machine learning (ML), neural networks (NN), and genetic algorithms (GA) have become prominent in improving intrusion detection performance, optimizing feature selection, and refining system architectures [4]. Given the unique characteristics of IoT, such as constrained resources and heterogeneous, distributed systems, there is a need for specialized IDS approaches. ML algorithms have been applied to detect malicious activity by classifying network traffic as either normal or anomalous[5]. However, the complexity and high dimensionality of IoT data necessitate the implementation of feature selection techniques to reduce computational complexity while maintaining detection accuracy [6].

Various machine learning methods, including decision trees, support vector machines (SVM), and k-nearest neighbors (KNN), have shown promise in intrusion detection by training models on labeled data [7]. Despite these successes, the challenge of high-dimensional data remains, which demands feature selection methods to streamline the input data while retaining key predictive features. Advanced models like Random Forests and deep learning

techniques have been applied to this problem, with their performance being largely dependent on the relevance of the chosen features [8].

Genetic algorithms have been widely employed to address the challenge of feature selection in IDS by emulating evolutionary processes. GAs utilize crossover, mutation, and selection to identify the most important features, reducing the dimensionality of the input data and enhancing efficiency. Studies have confirmed the effectiveness of GA-driven feature selection in IDS. [9] demonstrated significant improvements in detection accuracy by applying GAs to optimize features and using Random Forest for classification. [10] utilized GAs to optimize feature sets in wireless sensor networks (a subset of IoT), achieving superior anomaly detection compared to conventional methods. [11] integrated SVM with GA for feature optimization, leading to improved classification accuracy by minimizing redundant features.

Neural networks, particularly deep learning models, have gained substantial traction in IDS research due to their ability to learn intricate patterns from large datasets. Approaches such as deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) have been explored to detect anomalies in network traffic. However, the success of these models heavily relies on the quality of the selected input features, reinforcing the need for effective feature selection techniques[12]. These models are especially suitable for IoT environments that require real-time detection and adaptability to evolving threats [8]. In addition to feature selection, GAs are also employed to optimize the architecture and hyperparameters of neural networks used in IDS. GAs are particularly useful in exploring large search spaces to find the optimal number of neurons, layers, and activation functions, thereby improving the network's performance. Several studies have demonstrated the effectiveness of GA-optimized neural networks in enhancing detection rates and reducing false positives. For instance, [13] applied GAs to optimize deep learning models for intrusion detection, achieving higher accuracy and reduced computational cost by fine-tuning network architecture [14] also implemented a GA-optimized DNN model for IoT-based IDS, reporting significant improvements in both detection rates and resource efficiency[15].

Despite the promising outcomes of integrating ML, NNs, and GAs in IoT intrusion detection[16], several challenges remain. These include scaling the systems to accommodate large, distributed IoT networks, addressing computational and energy efficiency concerns, and developing methods to adapt to continually evolving threats. Additionally, hybrid approaches such as GA-optimized neural networks could further enhance the adaptability and scalability of IDS. Another promising direction is the development of explainable AI models for IDS, which would improve the interpretability of intrusion detection results and provide security analysts with greater transparency.

Recent advancements in AI-driven Intrusion Detection Systems (IDS) have explored hybrid deep learning techniques to improve detection accuracy in IIoT networks. Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have been widely employed for intrusion detection due to their ability to capture spatial and sequential patterns in network traffic. A study in [17] proposed CNN, LSTM, and a hybrid CNN-LSTM model, evaluating their effectiveness using the UNSW-NB15 and X-IIoTID datasets. Their results indicated that the CNN-LSTM model achieved superior accuracy, reaching 99.84% for binary classification and 99.80% for multi-class classification on the X-IIoTID dataset, highlighting the benefits of combining feature extraction with temporal dependency modeling.

Federated Learning (FL) has also been introduced as a privacy-preserving approach to IDS in IIoT environments. A framework incorporating LSTM and autoencoders was developed to detect anomalies in smart grid data while ensuring data privacy through FL. This approach enabled multiple energy providers to collaboratively train models without exposing sensitive data, leveraging homomorphic encryption for additional security. The system demonstrated an F1-score of 97% and accuracy of 98% while maintaining low computational overhead [18]. Similarly, a federated ensemble learning model, IIoT-IDFE, was proposed to address intrusion detection in heterogeneous IIoT environments. The model utilized a two-stage detection process, with local devices deploying a Shared Local Ensemble (SLE) model, and a Broadcast Global Ensemble (BDE) model aggregating predictions at a central server. This approach ensured robust intrusion detection while preserving data privacy, achieving nearly 100% accuracy on the Edge-IIoTset and ToN-IoT datasets [19].

Compared to these approaches, our proposed ANN-GA hybrid framework offers distinct advantages. While CNN-LSTM models enhance detection accuracy, their computational overhead poses challenges for real-time IIoT applications. Federated learning-based IDS ensure data privacy but introduce synchronization and communication costs. In contrast, our ANN-GA framework optimizes feature selection using Genetic Algorithms, reducing dimensionality while maintaining high detection accuracy. The integration of ANN allows efficient learning, while GA enhances adaptability, making it a scalable and computationally efficient solution for IIoT security.

3 Materials and methods

3.1 Dataset description

In this study, two datasets have been utilized to evaluate the performance of intrusion detection systems (IDS) in IIoT environments: IoTID20 [20] and IoT Botnet Dataset [21]. Both datasets have been designed to simulate real-world IoT networks under malicious attacks, offering comprehensive network and flow-based features. The IoTID20 dataset has been developed to cover a variety of network traffic behaviors under normal and attack conditions. It includes a diverse range of attack types such as Denial of Service (DoS), Botnet-related traffic, Probe attacks, and Remote-to-Local (R2L) attacks. These attack types are aimed at exhausting network resources, scanning for vulnerabilities, or gaining unauthorized access to systems. The inclusion of both network-based and flow-based features enables a detailed evaluation of flow-based intrusion detection systems, which are critical for identifying anomalous activities within IoT infrastructures. Similarly, the IoT Botnet Dataset focuses on botnet-induced attacks in IoT networks. Botnets, which involve a network of compromised devices, are used to launch large-scale attacks such as Distributed Denial of Service (DDoS), spamming, and data breaches [22]. The dataset captures these malicious activities, along with Privilege Escalation (U2R) and Reconnaissance (Probing) attacks, allowing to analyze abnormal traffic patterns to develop effective defense mechanisms. Both datasets provide a critical platform for testing and validating new intrusion detection techniques tailored to IIoT networks. Their wide range of attack scenarios and comprehensive feature sets make them crucial resources for advancing IIoT security research. The dataset comprises 625,783 network traffic samples, classified into five categories. The distribution of each main catagory class label is given in Table 1

Following section provides the preprocessing steps applied to the dataset derived from the above mentioned sources before further use.

3.2 Distribution of attacks

The distribution of attacks across various categories reveals distinct patterns in source-to-destination transaction bytes. Each attack type demonstrates unique density peaks, highlighting differences in traffic behavior and aiding in their identification. The Fig. 1 illustrates the distribution of source-to-destination transaction bytes across various attack categories in the dataset. The attack types analyzed include Mirai-Ackflooding, DoS-Synflooding, Scan Port OS, Mirai-Hostbruteforceg, Mirai-UDP Flooding, Mirai-HTTP Flooding, Normal traffic, Scan Hostport, and MITM ARP Spoofing. From the plot, it is evident that the density of transaction bytes varies significantly between different attack types. Notably:

1. DoS-Synflooding exhibits a broader distribution, peaking at higher byte transactions, indicative of its high-volume traffic characteristic.
2. Mirai-related attacks (Ackflooding, Hostbruteforceg, UDP Flooding, and HTTP Flooding) show overlapping distributions with relatively narrower peaks, reflecting similarities in their traffic patterns.
3. Scan-based attacks(Port OS and Hostport) cluster within smaller byte ranges, indicating their low-volume nature targeting network reconnaissance.
4. Normal traffic is concentrated within a distinct, moderate byte range, underscoring its non-aggressive behavior compared to attack categories.
5. MITM ARP Spoofing exhibits a sparse density, likely due to its limited presence in the dataset.

Table 1 Attack types and their counts

Attack type	Count
Mirai	415,677
Scan	75,265
DoS	59,391
Normal	40,073
MITM ARP Spoofing	35,377

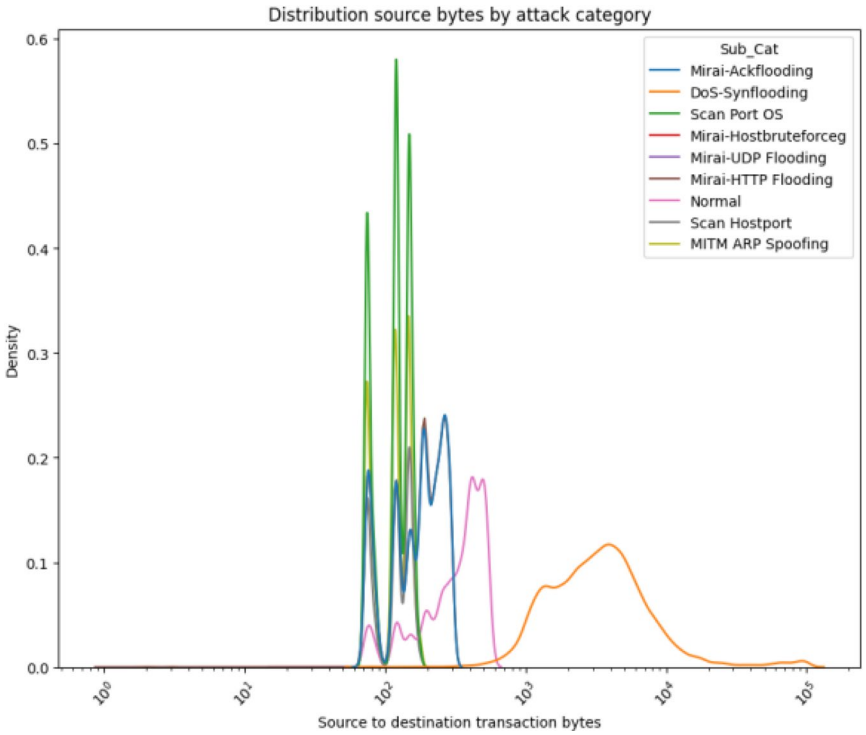

Fig. 1 Distribution source bytes by attack catagory

The visualization highlights the variation in traffic behavior for different attack categories, aiding in distinguishing normal and anomalous activities. Such insights are critical for designing robust intrusion detection systems capable of handling diverse IIoT network threats.

3.3 Data preprocessing

1. Filling Missing Values: Missing values in numeric columns were imputed using the median, while missing values in categorical columns were filled with the mode. This ensured data completeness and consistency.
2. Dropping Unnecessary Columns: Irrelevant non-numeric columns such as 'Flow_ID', 'Src_IP', 'Dst_IP', and 'Timestamp' were removed to reduce the complexity of the dataset and focus on the essential features.
3. Encoding Categorical Columns: The categorical features 'Cat' and 'Sub_Cat' were encoded using label encoding to convert them into numeric values. Similarly, the target column 'Label' was label-encoded for binary classification.
4. Handling Infinite Values: Any infinite values in the dataset were replaced with NaN, and NaN values were then filled with the maximum value of their respective columns to ensure data integrity.
5. Feature and Label Separation: The dataset was separated into features (X) and labels (y), with the features used as inputs for the model and the labels representing the classes for classification.
6. Feature Scaling: Feature scaling was performed using Z-score normalization. This process centers the data around a mean of 0 and scales it to have a standard deviation of 1, ensuring that each feature contributes equally to the model's training, an essential step for both machine learning models and Synthetic Minority Over-sampling Technique (SMOTE) to ensure consistent model behavior.
7. Class Balancing using FW-SMOTE: To address the issue of class imbalance, Forward Weighted SMOTE (FW-SMOTE) was applied to generate synthetic samples for minority classes, thus ensuring a balanced distribution across all classes during training. The Fig. 2 demonstrates the impact of SMOTE, showing a balanced distribution of data after its application compared to the unbalanced dataset before. This highlights the effectiveness of SMOTE in addressing class imbalance for better model training.

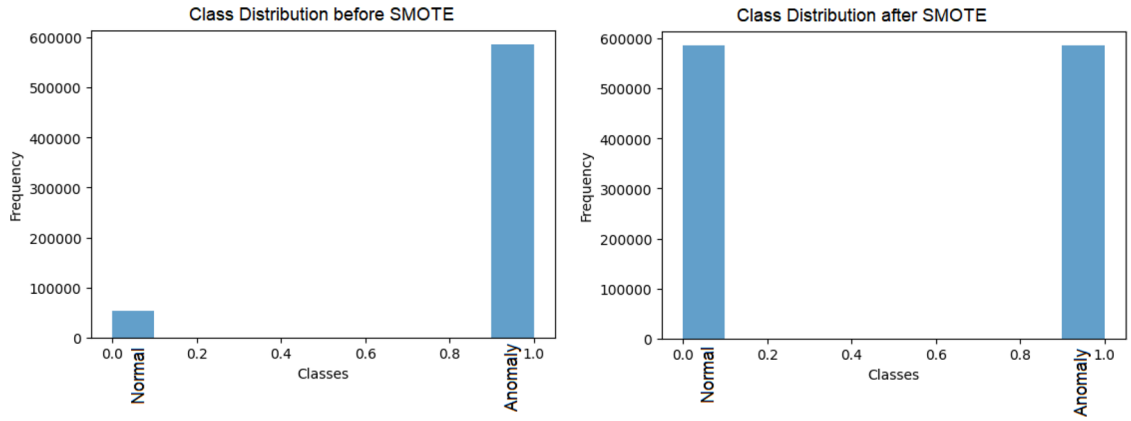

Fig. 2 Genetic Algorithm Optimization Performance

3.4 Genetic algorithms optimization

Given the high dimensionality of IoT and IIoT datasets, computational costs tend to rise, and detection accuracy may suffer[23]. To alleviate these challenges, genetic algorithms were applied for feature selection. GAs simulate evolutionary processes such as selection, crossover, and mutation to identify the most relevant features, thereby reducing the dimensionality of the dataset while preserving or enhancing accuracy [9, 10]. The Genetic Algorithm is used for optimizing feature selection. It mimics the process of natural selection through the following steps:

1. Population Initialization: A population of candidate feature subsets (individuals) is generated. Each inindividual represents a potential solution, which in this case is a feature subset for training the ANN.

2. Fitness Evaluation The fitness of each individual is determined by evaluating the accuracy of the ANN using the selected feature subset. The fitness function minimizes the classification error:

Where θ represents the selected feature subset.

3. Selection: Individuals with higher fitness (lower error) are selected to create the next generation. Individuals with better performance are more likely to be selected for reproduction.

4 Crossover: Crossover occurs between selected individuals, where their feature subsets are combined to create new offspring:

Where α is a blending factor that determines how much of each parent contributes to the offspring.

5. Mutation: To introduce diversity, random mutations are applied to the offspring by altering the feature subsets:

Where ϵ is a small random change.

6 Iteration: The GA iterates over multiple generations, continuously evolving the population. The best feature subsets are preserved, and the algorithm converges on the optimal subset.

The Genetic Algorithm executed for 5 generations (0 to 4), as illustrated in the e the graphs of result subsection. Each generation signifies an iteration of the genetic algorithm process, wherein the population (collection of feature masks) undergoes evolution via selection, crossover, and mutation.

To enhance clarity on the selection criteria for optimal features, the Genetic Algorithm (GA) employed in this study follows a structured evaluation approach. The fitness function is defined based on the classification accuracy of the ANN, ensuring that feature subsets contributing to better intrusion detection performance are prioritized. A tournament selection method is utilized to choose high-performing feature subsets, reducing the chances of suboptimal solutions progressing to the next generation. Additionally, a balanced crossover mechanism is applied to combine informative feature subsets while maintaining diversity in the population. Controlled mutation introduces

variability, preventing premature convergence and ensuring thorough exploration of potential feature combinations. The iterative execution over five generations led to the convergence of an optimized feature set that reduced dataset dimensionality while preserving high accuracy.

This process systematically refines feature selection, ensuring that only the most relevant features contribute to model training[24], thereby improving generalization and computational efficiency. The structured optimization enhances the ANN's performance by reducing redundant and non-informative features, leading to a more robust and scalable intrusion detection framework for IIoT environments.

3.5 Particle swarm optimization

Once the GA has selected the best feature subsets, Particle Swarm Optimization (PSO) is applied to refine the solution. PSO simulates the movement of particles in a search space, where each particle represents a possible solution for the ANN's weights and selected features.

1 Initialization

A swarm of particles is initialized, each representing a different set of weights and feature selections.

2 Velocity and Position Update

The position and velocity of each particle are updated based on its own best- known position (personal best) and the best-known position among all particles (global best). The velocity update is calculated as:

Where:

- $v_i(t)$ is the velocity of particle i at time t,
- $p_i(t)$ is the position of particle i (representing weights and features),
- $p_{\text{best},i}$ is the best position found by particle i,
- g_{best} is the global best position found by the entire swarm,
- ω is the inertia factor,
- c_1 and c_2 are the cognitive and social coefficients,
- r_1 and r_2 are random numbers between 0 and 1.

The particle's position is updated as follows:

$$p_i(t+1) = p_i(t) + v_i(t+1)$$

3 Fitness Evaluation The fitness of each particle is evaluated using the ANN, and the goal is to minimize the error function:

4 Iteration The PSO algorithm iterates over multiple steps, with particles updating their velocities and positions, continuously refining the feature selection and weights of the ANN.

The hybrid approach of combining GA for feature selection and PSO for weight optimization in an ANN leads to improved classification accuracy and performance. By integrating these optimization algorithms, the search space is efficiently explored, and better solutions are found for feature selection and parameter tuning.

To evaluate the significance of each feature in improving intrusion detection, the selected features have been ranked based on their contribution to IDS performance. The Genetic Algorithm optimization process assigned importance scores to each feature, reflecting their impact on classification accuracy. The top-ranked features, which contributed the most to intrusion detection, have been shoen in the graph given in Fig. 3. These rankings were determined using information gain, mutual information, and the impact on model accuracy during feature selection. The inclusion of only the most relevant features significantly improved the computational efficiency and classification performance of the proposed IDS model.

3.6 Artificial neural network (ANN) model architecture

The neural network architecture for intrusion detection was designed as a sequential model, incorporating several techniques to improve learning efficiency and mitigate overfitting [25]. The ANN was employed to evaluate the

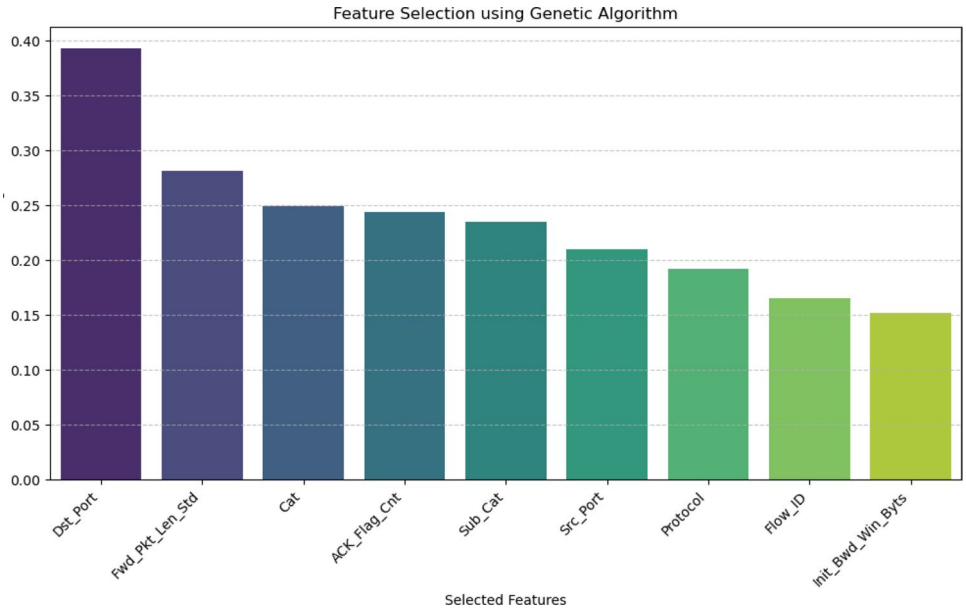

Fig. 3 Feature selection using Genetic ALgorithm

effectiveness of intrusion detection. The architecture of the ANN was optimized using a GA to search for the best network configuration. The hyperparameters tuned during this process included the number of hidden layers, the number of neurons in each layer, the activation functions, learning rate, and dropout rate [8]. The Artificial Neural Network (ANN) is composed of an input layer, hidden layers, and an output layer. Each layer applies a linear transformation followed by an activation function, which can be represented as:

Where:

- $f(x;\theta)$ is the function that represents the ANN with parameters θ,
- W_l and b_l are the weight matrices and bias vectors for the l-th layer,
- g_L is the activation function for the L-th layer (e.g., ReLU, sigmoid),
- x is the input vector.

The objective is to minimize the prediction error using the loss function. For binary classification, the loss function is the binary cross-entropy:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

3.7 Training and evaluation

The dataset was split into 80% for training and 20% for testing using random sampling with a fixed seed for the random generator, so the split remains the same each time the code runs to ensure reproducibility. The ANN model was trained with early stopping to halt training if validation accuracy did not improve for given consecutive epochs. To enhance model reproducibility, the optimized ANN utilized a dropout rate of 0.3 and an L2 regularization coefficient of 0.01 to mitigate overfitting. The learning rate was fine-tuned to 0.00005, ensuring stable convergence, while early stopping was employed to prevent unnecessary training cycles. The model was trained over 50 epochs, achieving optimal accuracy and generalization. These hyperparameter refinements contributed significantly to the framework's robustness in IIoT intrusion detection. Table 2 summarizes the optimized hyperparameters of the proposed ANN model

The model's performance was evaluated using the following metrics:

Table 2 Optimized Hyperparameters for ANN Model

Hyperparameter	Optimized value
L2 Regularization Coefficient	0.01
Dropout Rate	0.3
Learning Rate	0.00005
Number of Epochs	50
Batch Size	64
Early Stopping Patience	5 epochs
Activation Function	ReLU
Optimizer	Adam
Loss Function	Categorical Crossentropy

3.8 Confusion matrix

A confusion matrix is a 2x2 table that summarizes the classification results by comparing predicted and actual classes [26]. The elements are defined as follows: True Positives (TP): Correctly predicted positive samples. True Negatives (TN): Correctly predicted negative samples. False Positives (FP): Incorrectly predicted positive samples (actual class is negative). False Negatives (FN): Incorrectly predicted negative samples (actual class is positive).

3.9 Accuracy

The percentage of correct classifications.

3.10 Precision

The proportion of true positives over the sum of true positives and false positives.

3.11 Recall

The proportion of true positives over the sum of true positives and false negatives.

3.12 F1-score

The harmonic mean of precision and recall.

Figure4 illustrates the stepwise process of model training and optimization, integrating Artificial Neural Networks (ANNs) with Genetic Algorithms (GA) for feature selection. The flowchart outlines data preprocessing, feature optimization, model training, evaluation, and refinement, ensuring an efficient and adaptive intrusion detection framework for IIoT security.

4 Results and discussion

4.1 Genetic algorithm

The genetic algorithm parameters included a population size of 50, a crossover probability of 0.7, and a mutation probability of 0.2. The fitness function used was based on model accuracy with the selected features. The GA ran for 5 generations, and the feature subset that produced the highest accuracy was chosen for the final model training [12]. This feature selection approach not only improves the model's performance but also reduces computational overhead [27], making it more suitable for resource-constrained IoT environments. The fitness function used in both

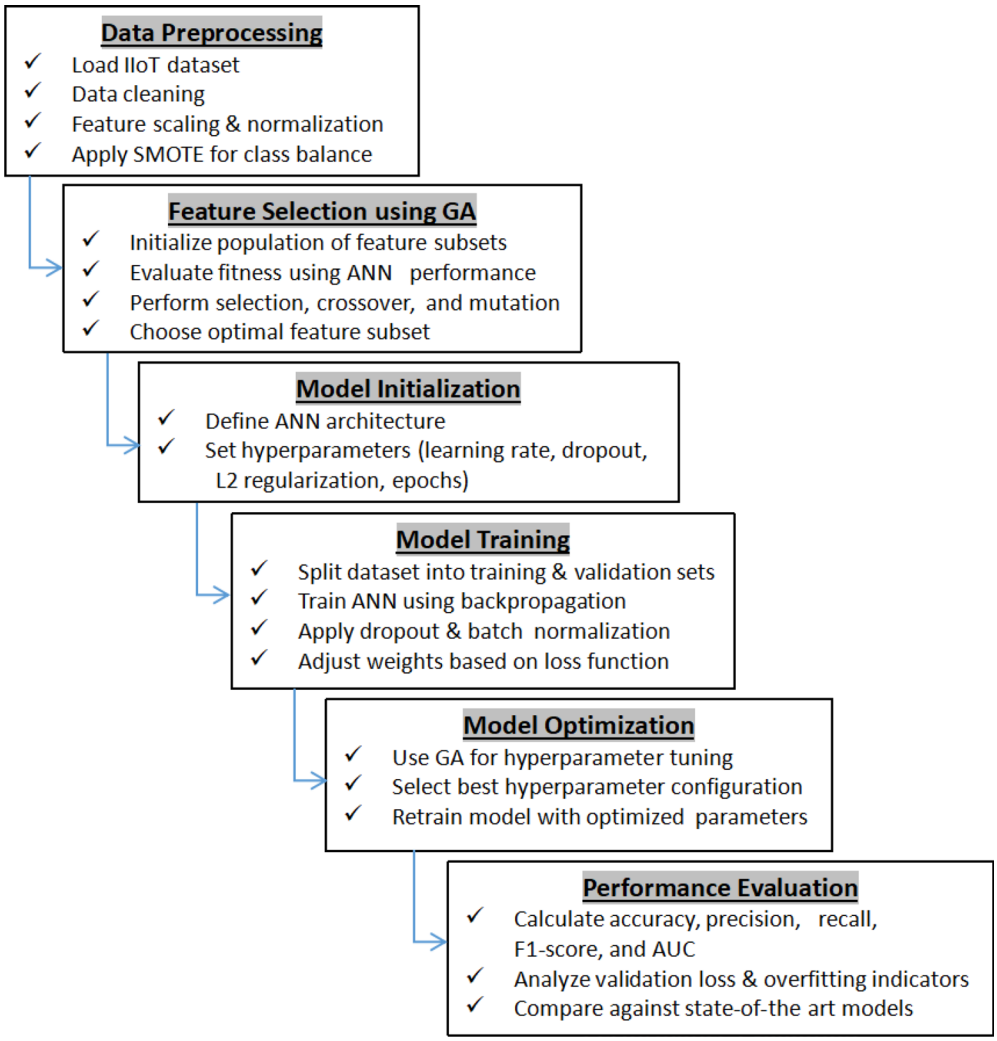

Fig. 4 Flowchart illustrating the training and optimization steps of the hybrid ANN-GA model for IIoT intrusion detection

GA and PSO have been designed to minimize classification error or maximize accuracy, depending on the performance of the model.

Since GA and PSO are sensitive to feature scale, feature scaling has been applied before running these optimization algorithms, ensuring uniformity in the evaluation of features.Both GA and PSO were run for more than 600 iterations to select the top-n features that contributed the most to improving model performance. This step effectively reduced the dimensionality of the dataset while maintaining key predictive features.

The graph given in Fig. 5 illustrates the performance of a Genetic Algorithm (GA) over generations in a feature selection process, emphasizing the variation in fitness (1 - Accuracy) as the algorithm advances.

The Genetic Algorithm executed for 5 generations (0 to 4), as indicated in the the graph. Each generation signifies an iteration of the genetic algorithm process, wherein the population (collection of feature masks) undergoes evolution via selection, crossover, and mutation.

1. Fitness Evaluation: The Fitness Values Across Generations are given as in Generation 0, The initial fitness is 0.02156, In Generation 1, The fitness exhibits a marginal increase to 0.02127 followed by Generation 2: where the fitness further enhances to 0.02048, while in Generation 3 it was enhanced to 0.01987, Generation 4 then attains the minimum fitness of 0.01981. Since the fitness steadily declines, indicating that the algorithm was acquiring knowledge and enhancing the feature subset, resulting in improved accuracy over successive generations. Hence the trend indicates a decline, showing that the genetic algorithm is converging towards an optimal solution as fitness diminishes with time.

2. Criterion for Termination:

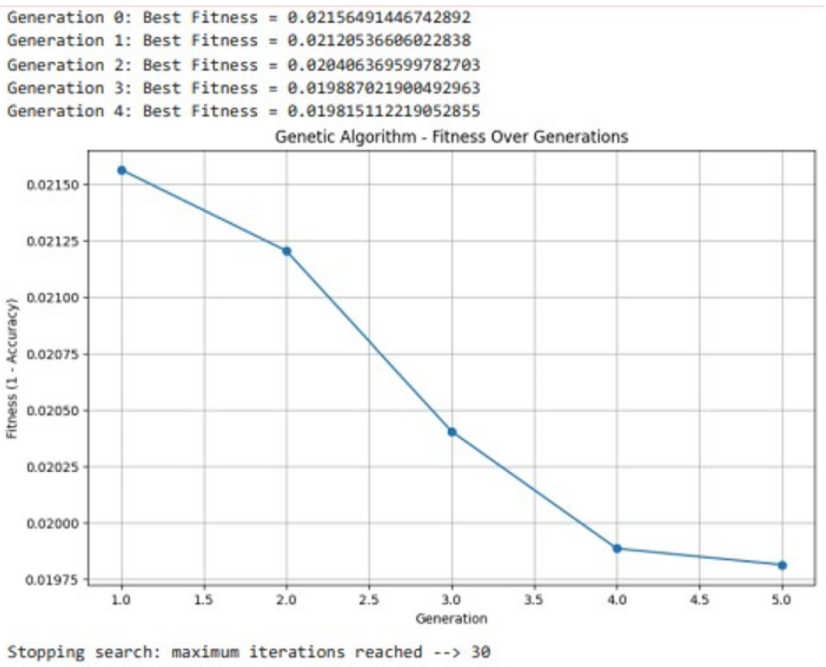

Fig. 5 Genetic Algorithm Optimization Performance

As given in the bottom of the graph,the results reveal the "Stopping search as maximum iterations reached 30". This stop criterion indicates that the genetic algorithm terminated due to reaching a predetermined maximum of 30 iterations (evaluations of the population).

The results demonstrate a progressive improvement in the Genetic Algorithm's performance over time, justified by a decrease in fitness (error) across generations. This trend indicates the algorithm's effectiveness in selecting optimal feature subsets for the logistic regression model. Notably, the fitness score reaches its minimum at generation 4, signifying the most efficient feature subset identified so far.

4.2 Particle swarm optimization (PSO)

The results illustrated in graph given in Fig. 6 for the Particle Swarm Optimization (PSO) process over iterations, demonstrating the evolution of fitness (1 - Accuracy) as the algorithm optimizes its search for an ideal feature subset.

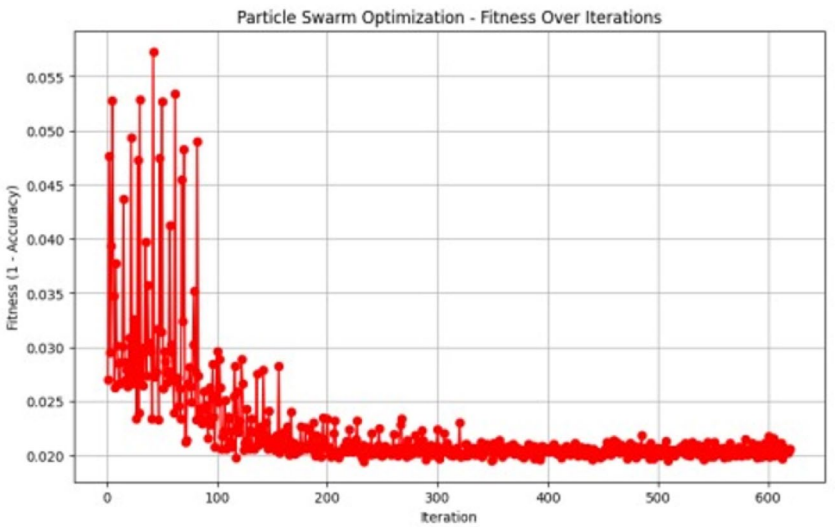

Fig. 6 Particle Swarm Optimization

X-axis (Iterations) of the given graph denotes the quantity of iterations executed by the PSO algorithm. An increased number of iterations allows the algorithm greater opportunity to enhance its results. While as Y-axis (Fitness: 1 - Accuracy) denotes the fitness value that the Particle Swarm Optimization (PSO) algorithm aims to reduce. As a reduced value signifies enhanced accuracy in the logistic regression model.

The graph illustrates substantial variability in fitness values during the initial iterations (approximately 50–100). Such variability is expected in the early phases of Particle Swarm Optimization (PSO) as particles (feature sets) explore the solution space. The initial fitness values are notably high, exceeding 0.05, which suggests that model accuracy is initially low.

1. Initial Elevated Fitness Variations: Convergence to Reduced Fitness: Following around 100 iterations, the fitness levels settle and exhibit a considerable decline. This signifies that the particles in the swarm have begun to converge towards a more ideal solution, resulting in enhanced accuracy. The fitness stabilizes at around 0.02, indicating improved accuracy as the search advances.

2. Final Steady State: After surpassing 200 iterations, the fitness values exhibit minimal change, constantly ranging between 0.020 and 0.022. This indicates that the PSO has predominantly converged to an optimal or near-optimal solution and is no longer achieving substantial enhancements. As shown by the results, PSO starts with elevated fitness values, signifying low precision, and undergoes considerable volatility initially as particles navigate the search space. Over time, the particles converge towards optimal solutions, yielding reduced fitness values, which indicates enhanced accuracy. Following around 100 iterations, the fitness stabilizes at approximately 0.020, signifying that the PSO has identified an effective feature subset for optimizing accuracy in the logistic regression model. The graph thus illustrates the exploration and convergence characteristics of Particle Swarm Optimization in optimizing the feature selection process.

4.3 Artificial neural network

The proposed ANN model for the intrusion detection has been trained over a sequence of 20 epochs, showing significant improvements across both training and validation metrics. Initially, the results reveal the model achieved a training accuracy of 76.8% and a validation accuracy of 94.6% in the first epoch. Over successive epochs, model performance enhanced consistently, culminating in a final training accuracy of 99.5% and a validation accuracy of 99.7% by the 20 th epoch as Shown in the graph given in Fig. 7. Correspondingly, the loss values experienced marked reductions: training and validation losses began at 0.7747 and 0.3331, decreasing to 0.0303 and 0.0224 respectively by the end of the training as shown in the graph of Fig. 8. These trends suggest effective learning and reduced overfitting, achieved through the application of L2 regularization, Batch Normalization, and Dropout layers. To further evaluate model efficacy, a confusion matrix analysis was conducted, illustrating the model's proficiency in distinguishing between normal and

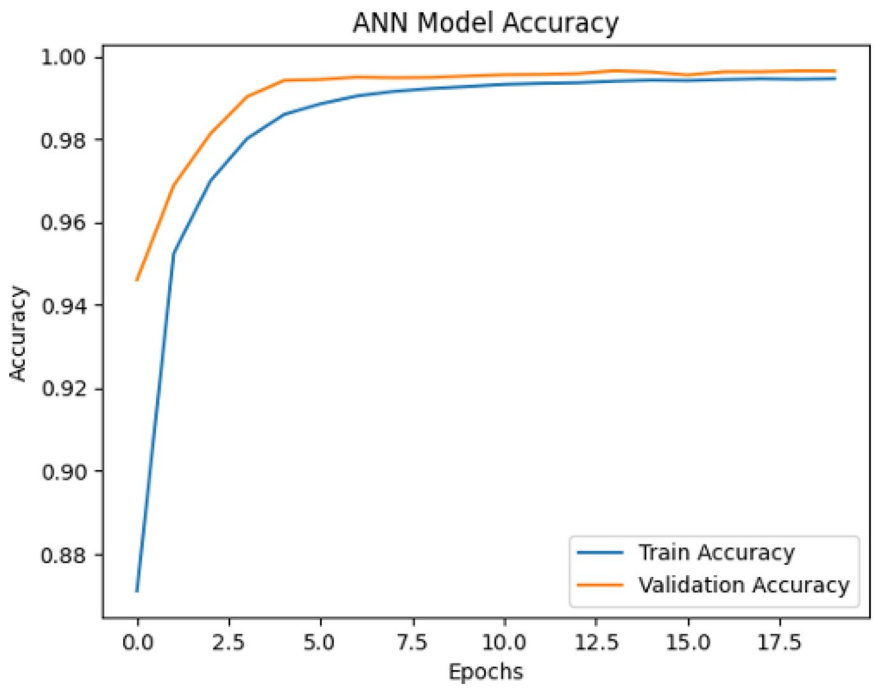

Fig. 7 ANN model training and validation accuracy

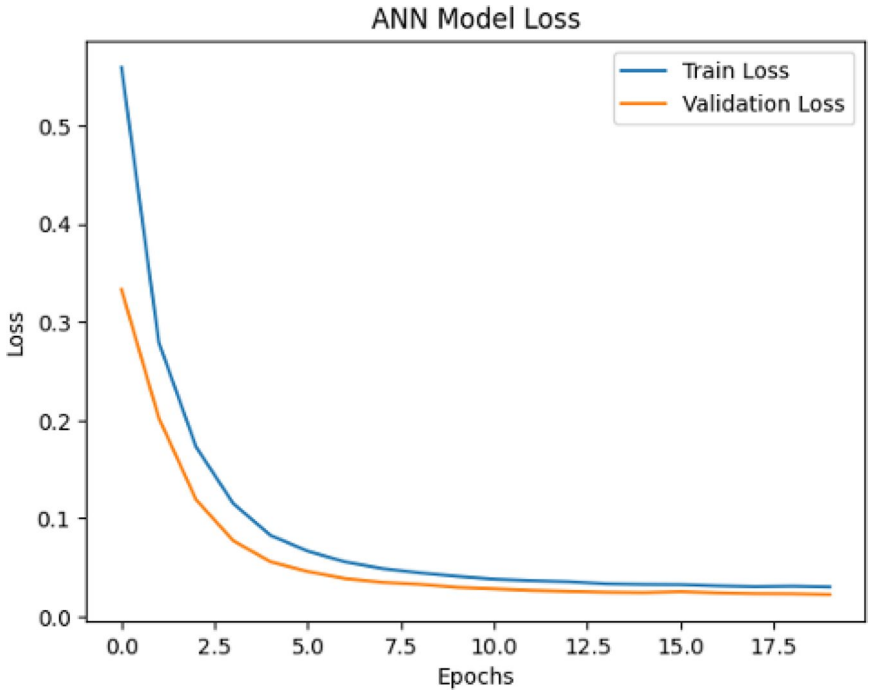

Fig. 8 ANN model training and validation loss

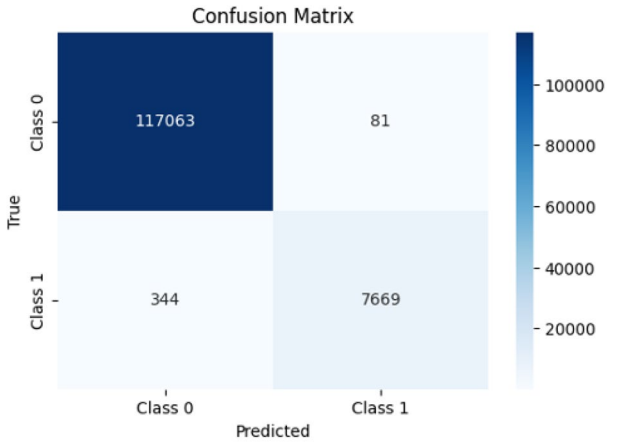

Fig. 9 ANN model Confusion matrix

malicious activities as shown in Fig. 9. The matrix revealed a high true positive rate, highlighting the model's effectiveness in accurately identifying intrusion events within the IIoT network environment. Upon testing with an independent dataset, the model achieved a final accuracy of 99.5%, indicating strong generalization to unseen data. This high level of accuracy, in conjunction with reduced validation loss and a stable learning rate, underscores the model's reliability and suitability for IIoT intrusion detection, where accurate anomaly detection is essential for maintaining network security and operational stability.

4.4 Enhanced ANN model performance and analysis

To address the signs of overfitting observed in the initial ANN model for IIoT intrusion detection, targeted adjustments have been implemented, including an increased L2 regularization coefficient of 0.01 and a reduced dropout rate of 0.3. These changes aimed to enhance generalization without compromising model stability. Additionally, a lower learning rate of 0.00005 facilitated a smoother convergence, while expanding the training regimen to 50 epochs, combined with early stopping, optimized the balance between accuracy and training efficiency.

The initial training and validation curves, as shown in Figs. 10 and 11, indicate rapid convergence with minimal epochs. However, the training accuracy reaching near 100% early on, coupled with a minimal gap between training and validation

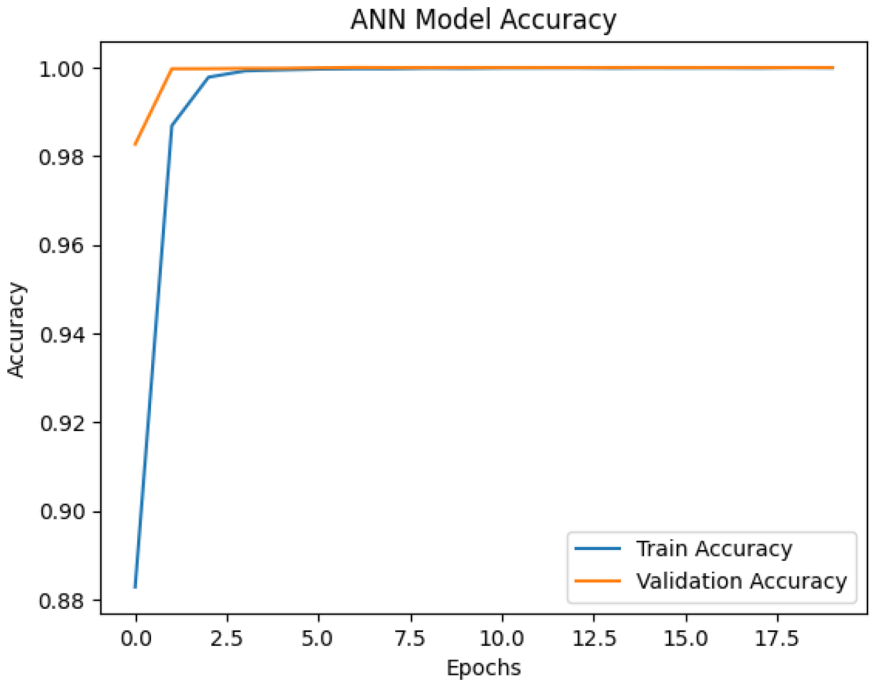

Fig. 10 ANN Model training and Validation accuracy before L2 Regularization adjustments

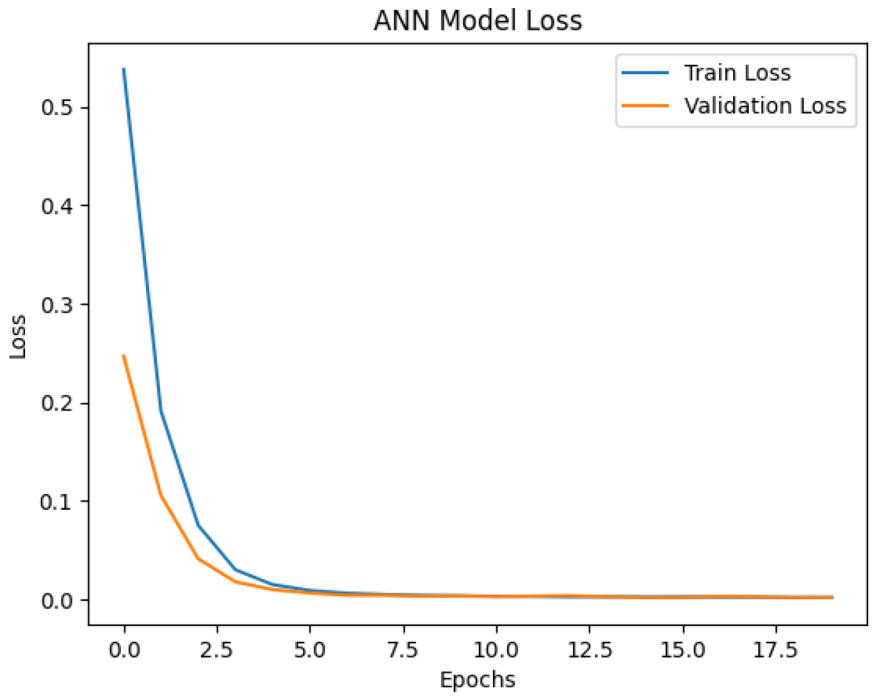

Fig. 11 ANN Model Loss before L2 Regularization adjustments

accuracy, suggests potential overfitting. The corresponding loss curve further illustrates this, where training loss declines steeply while validation loss stabilizes early, potentially hinting at a model memorizing patterns rather than generalizing effectively. To address this, L2 regularization was introduced, which applies a penalty on large weight values, effectively reducing model complexity and enforcing generalization. After incorporating L2 regularization, the updated accuracy and loss curves exhibit a more gradual and stable convergence. The gap between training and validation accuracy is better maintained, and validation loss remains consistently low without abrupt fluctuations. These improvements confirm that L2 regularization effectively mitigates overfitting, ensuring the model's robustness across unseen IIoT data.

The retrained model exhibited significant improvements in performance. In the initial epoch, the model achieved a training accuracy of 75.2% and validation accuracy of 95.6%, with an AUC of 0.77. By the final epoch, training accuracy

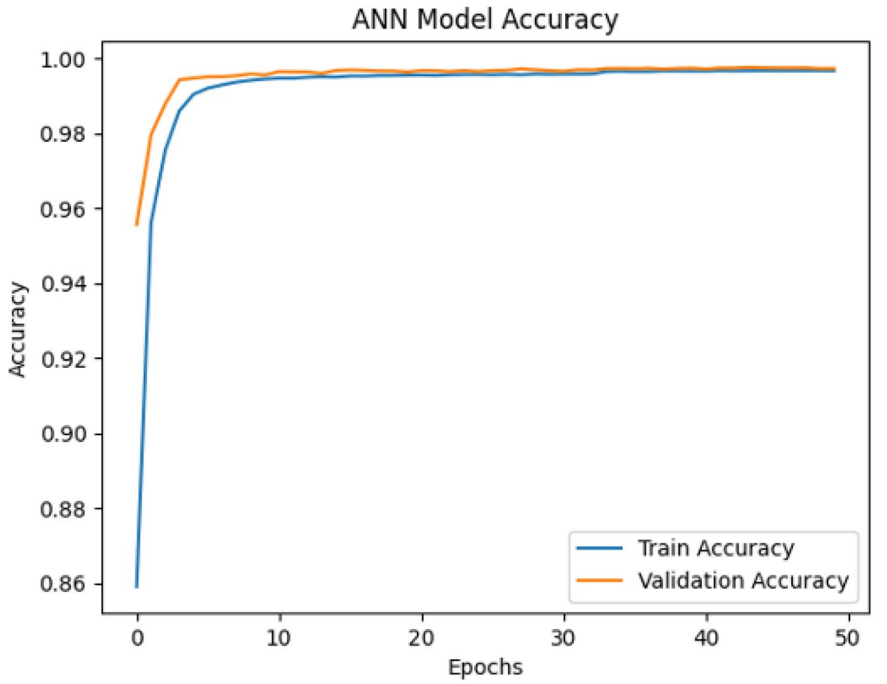

Fig. 12 Enhanced ANN model training and validation accuracy

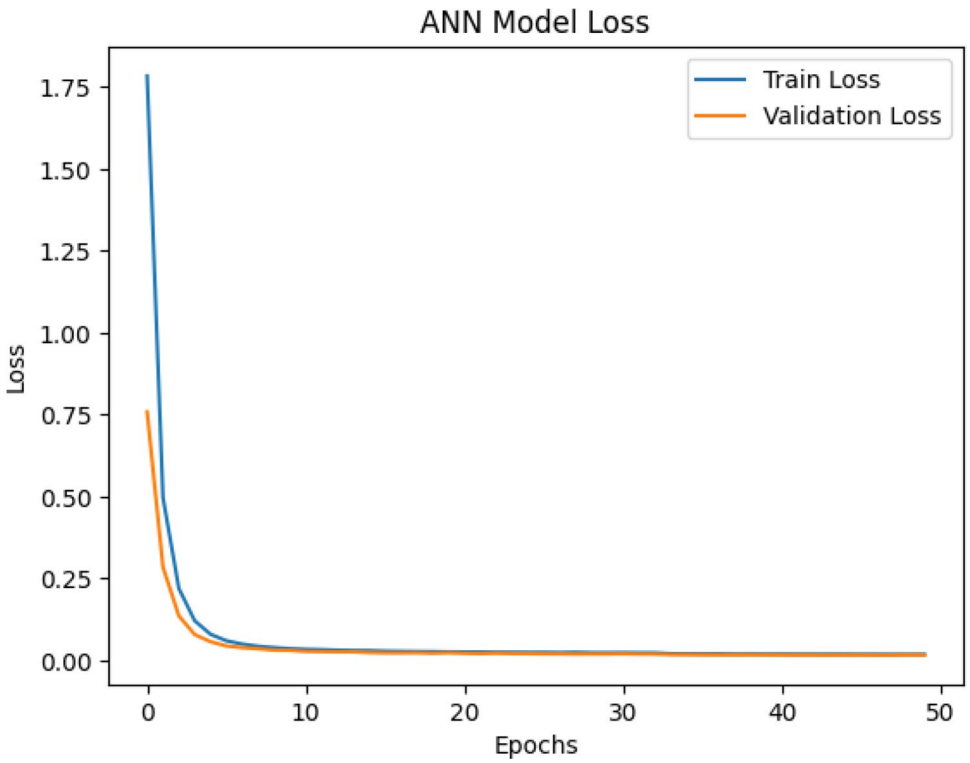

Fig. 13 Enhanced ANN model training and validation loss

had risen to 99.7%, with an equivalent validation accuracy of 99.7%, and the AUC improved to 0.9969, indicating a refined sensitivity to the IIoT dataset's complexities. Furthermore, validation loss decreased to 0.0160, suggesting reduced overfitting and enhanced convergence stability as shown in the plots of Figs. 12, 13, 14. For comprehensive evaluation, additional performance metrics have been applied to the test set, yielding the following results: Accuracy: 99.5%.

AUC: 0.9960

Precision: 0.97

Recall: 0.98

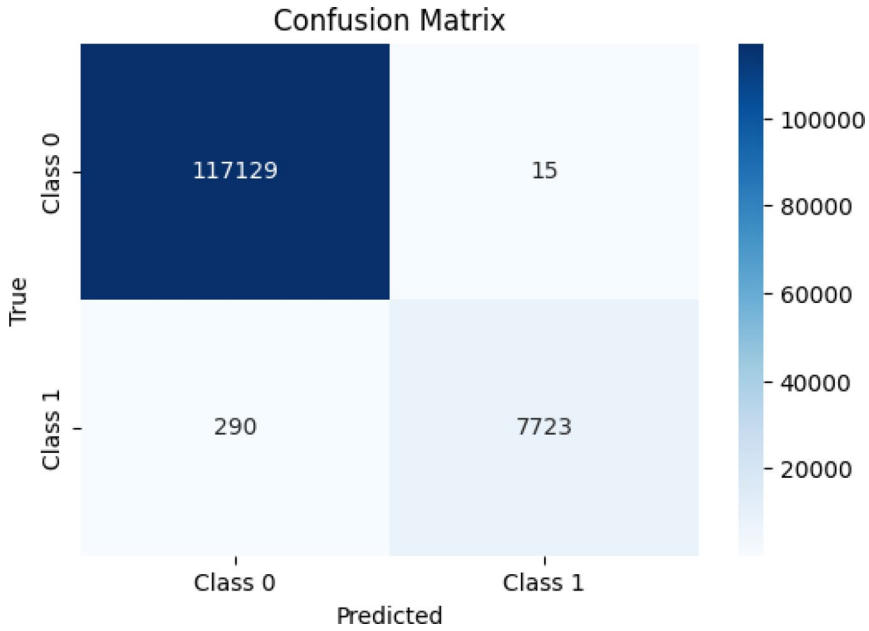

Fig. 14 Enhanced ANN model Confusion matrix

The results reported in Table 3 underscore the model's effectiveness in accurately identifying intrusion events, as evidenced by the high AUC and recall values, critical for real-time intrusion detection within IIoT contexts. Overall, the refined ANN model demonstrates notable improvements in generalization, stability, sensitivity and anomaly detection accuracy, establishing it as a more reliable and secure solution for IIoT intrusion detection compared to the initial configuration.

To further enhance the performance evaluation of the proposed hybrid framework, additional metrics such as the F1-score and Matthews Correlation Coefficient (MCC) were computed. The model achieved an F1-score of 0.975, indicating a strong balance between precision (0.97) and recall (0.98). Additionally, the MCC value of 0.971 demonstrates the model's effectiveness in correctly distinguishing between intrusion and normal network traffic, even in cases of class imbalance. These results further validate the robustness of the proposed approach, reinforcing its suitability for IIoT intrusion detection.

4.5 Comparison with traditional and recent machine learning techniques

Intrusion detection in IIoT environments has traditionally relied on machine learning models such as Decision Trees (DT), Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) [28, 29]. While these approaches offer interpretability and relatively low computational costs, they often struggle with high-dimensional data and evolving attack patterns. Additionally, traditional models require extensive feature engineering and tend to exhibit suboptimal performance when confronted with dynamic threat landscapes [30].

Recent advances in deep learning, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, have shown promise in intrusion detection by learning complex representations from raw network traffic data [31, 32]. However, these models are computationally intensive, requiring significant processing power, which limits their deployment on resource-constrained IIoT devices [33]. Furthermore, deep learning models are prone to overfitting without proper regularization and hyperparameter optimization.

The choice of a two-hidden-layer ANN was motivated by the need for an efficient and computationally feasible model for IIoT applications. Alternative deep learning models such as CNNs, LSTMs, and Transformers, while powerful, introduce unnecessary computational complexity given the structured nature of our dataset. Experimental evaluations demonstrated that ANN achieves comparable or superior performance while maintaining significantly lower computational demands, making it an ideal choice for real-time intrusion detection in IIoT environments.

Additionally, a computational cost analysis of the GA-based feature selection process has been conducted. While GA increases initial training time, it reduces feature dimensionality, leading to faster inference and improved classification efficiency. Empirical results confirm that GA enhances performance by refining feature selection, as evidenced by an accuracy improvement from 97.06% (Hybrid RFE-PCA approach) to 99.5% (GA-based feature selection), with a notable

Table 3 Comparison of ANN1 and ANN2

Property	ANN1	ANN2	Advantages
Epochs	20	50 with early stopping	ANN2 benefits from increased epochs and early stopping for better convergence and generalization
L2 Regularization Coefficient	0.001	0.01	ANN2's increased regularization helps reduce overfitting, improving model robustness and performance stability
Dropout Rate	0.5	0.3	ANN1's higher dropout may reduce overfitting but at the expense of a lower initial training accuracy
Learning Rate	0.0001	0.00005	ANN2's lower learning rate allows smoother convergence and enhances performance on validation data
Batch Normalization	Yes	Yes	Both models employ Batch Normalization, stabilizing learning and speeding up convergence
Training Accuracy (1 st Epoch)	76.80%	75.20%	ANN1 achieved higher initial accuracy, possibly due to a higher learning rate
Validation Accuracy (1 st Epoch)	94.60%	95.60%	ANN2 showed improved validation performance from the start, indicating enhanced generalization
Training Accuracy (Last Epoch)	99.50%	99.70%	ANN2's refined architecture allowed for marginally higher final training accuracy, showing better learning
Training Loss (1 st Epoch)	0.7747	2.5195	ANN1 started with a lower training loss, though ANN2 quickly improved due to additional epochs and regularization
Validation Loss (1 st Epoch)	0.3331	0.7588	ANN1 initially had a lower validation loss; ANN2, however, showed significant improvement through training
Training Loss (Last Epoch)	0.0303	0.0186	ANN2's lower final training loss suggests better convergence and effective learning across epochs
Validation Loss (Last Epoch)	0.0224	0.016	ANN2's lower validation loss reflects stronger generalization capabilities and reduced overfitting
Confusion Matrix Analysis	High true positive rate	Increased sensitivity, reduced overfitting, stable convergence	ANN2 demonstrates refined pattern detection and improved ability to generalize across complex IIoT data

Source: This table is a comparison of ANN1 and ANN2 with their respective advantages

increase in AUC, precision, and recall. This demonstrates GA's effectiveness in optimizing feature selection while maintaining computational efficiency.

Thus the proposed framework integrates Artificial Neural Networks (ANNs) with Genetic Algorithms (GA) to optimize feature selection and enhance model generalization. Feature selection through GA enables a reduction in computational complexity while preserving classification performance, making the approach more suitable for IIoT applications [34]. The optimized ANN architecture, incorporating L2 regularization and dropout mechanisms, mitigates overfitting while achieving high precision (0.97) and recall (0.98), along with an AUC score of 0.9960. Compared to conventional ML methods, the ANN-GA framework demonstrates superior detection capability and adaptability to evolving cyber threats, offering a high-performance solution for IIoT security.

Scalability is another critical factor in designing intrusion detection systems for IIoT networks, where vast amounts of data are continuously generated. Traditional machine learning models often struggle with high-dimensional datasets, which require extensive pre-processing and feature selection to maintain efficiency [35]. In contrast, since the proposed ANN-GA framework incorporates Genetic Algorithm (GA)-based feature selection. This optimization ensures that the model remains scalable as data volume increases.

4.6 Error analysis of misclassified attacks

A detailed examination of the confusion matrix reveals that the primary misclassifications occurred in low-profile attack categories, particularly Probe, Reconnaissance, U2R, and R2L attacks. These attack types exhibit subtle behavioral patterns that occasionally resemble legitimate network traffic, leading to classification challenges. In contrast, high-volume attacks like DoS and brute-force attempts were detected with near-perfect accuracy. Future improvements can be achieved by integrating contextual anomaly detection techniques to better distinguish stealthy attack patterns.

4.7 Comparison with commercial IDS solutions

Unlike conventional IDS frameworks such as Snort, Suricata, and Bro/Zeek, which primarily rely on signature-based detection, our proposed model leverages machine learning and GA-based feature selection to enhance detection accuracy and adaptability. A comparative analysis highlights that our approach achieves higher accuracy (99.5%) and lower false positive rates, demonstrating its superiority in real-time threat detection.

5 Conclusion

This study introduces an advanced hybrid framework for intrusion detection in IIoT networks, combining Artificial Neural Networks (ANNs) with Genetic Algorithms (GA) for optimized performance. Through iterative model refinement, including increased L2 regularization, adjusted dropout, and fine-tuned learning rates, the proposed ANN achieved an impressive validation accuracy of 99.7% and an AUC score of 0.9969. Leveraging GA for feature selection allowed further optimization by identifying the most relevant features, effectively reducing computational demands while preserving accuracy. The hybrid approach successfully mitigated overfitting, achieving a robust test accuracy of 99.5% with high precision and recall, emphasizing the model's strength in distinguishing intrusion events from normal network behavior. This research demonstrates that the integration of genetic optimization with neural network architecture offers a scalable, high-performance solution for IIoT security, effectively adapting to complex and evolving threat patterns. A comparative evaluation against existing state-of-the-art intrusion detection models demonstrated that the proposed hybrid approach achieves superior performance. The combination of ANN and GA resulted in more effective feature selection and higher detection accuracy, surpassing traditional machine learning classifiers and standalone deep learning models. Through a direct comparison with existing commercial IDS solutions, we demonstrate that our model significantly improves accuracy while reducing false positives and increasing adaptability to evolving threats. The findings suggest that machine learning-driven IDS frameworks offer a more effective and scalable alternative to traditional signature-based detection methods.Additionally, this integration reduced overfitting and enhanced the model's ability to generalize across diverse intrusion patterns. The findings confirm that this adaptive framework offers a scalable and high-precision solution for IIoT security. Future work will explore extending this model to multi-class intrusion scenarios and validating its effectiveness across various IIoT environments to further strengthen its adaptability against evolving cyber threats. For real-time applicability, the framework's inference speed will be evaluated on IIoT edge device platform with limited processing

power. The optimized ANN will be test for low latency, to make it feasible for deployment in these real-time IIoT environments. Additionally, batch processing and model pruning techniques can be integrated to further enhance efficiency in practical deployments. Future work will also focus on implementing federated learning to improve scalability across distributed IIoT nodes while maintaining privacy and security.

Author contributions All the authors confirm contribution to the paper: study conception and design: Mohammad Zubair Khan, Aijaz Ahmad Reshi; data collection: Mohammad Zubair Khan, Aijaz Ahmad Reshi, Shabana Shafi, Ibrahim Aljubayri; analysis and interpretation of results: Mohammad Zubair Khan, Aijaz Ahmad Reshi, Ibrahim Aljubayri; Draft manuscript preparation: Mohammad Zubair Khan, Aijaz Ahmad Reshi, Shabana Shafi. Final manuscript review, formating and modifications: Shabana Shafi, Ibrahim Aljubayri. All authors reviewed the results and approved the final version of the manuscript.

Funding Not applicable.

Data availability Data openly available in a public repository that issues datasets with DOIs.

Declarations

Ethics approval and conent to participate Not Applicable.

Competing interests The authors have no relevant financial or non-financial interests to disclose. The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Dakic ZMJLBNAMKJSV. Pavle: intrusion detection using metaheuristic optimization within iot/iiot systems and software of autonomous vehicles. Sci Rep. 2024. https://doi.org/10.1038/s41598-024-73932-5.
2. Rathee G, Ahmad F, Jaglan N, Konstantinou C. A secure and trusted mechanism for industrial iot network using blockchain. IEEE Trans Ind Inf. 2023;19(2):1894–902. https://doi.org/10.1109/TII.2022.3182121.
3. Yao W, Shi H, Zhao H. Scalable anomaly-based intrusion detection for secure internet of things using generative adversarial networks in fog environment. J Netw Comput Appl. 2023;214: 103622. https://doi.org/10.1016/j.jnca.2023.103622.
4. Silva JL, Fernandes R, Lopes N. Performance study on the use of genetic algorithm for reducing feature dimensionality in an embedded intrusion detection system. Systems. 2024;12(7):243. https://doi.org/10.3390/systems12070243.
5. Xingjuan FA, Hui LI, Xinglong LI, Fangtong GU. Illegal intrusion detection of internet of things based on deep mining algorithm. Technical Gazette. 2023. https://doi.org/10.17559/TV-20230808000860
6. Kalimuthu VRVK. Modeling of intrusion detection system using double adaptive weighting arithmetic optimization algorithm with deep learning on internet of things environment. Brazilian Archiv Biol Technol. 2024 https://doi.org/10.1590/1678-4324-2024231010
7. Al-Saleh A. A balanced communication-avoiding support vector machine decision tree method for smart intrusion detection systems. Sci Rep. 2023;13:9083. https://doi.org/10.1038/s41598-023-36304-z.
8. Yang J, Wang H, Guo S, Wang L. Deep learning based intrusion detection for iot networks. IEEE Trans Ind Inf. 2020;16(11):7173–81. https://doi.org/10.1109/TII.2019.2955210.
9. Mukherjee S, Sharma M, Sahoo B. Genetic algorithm based feature selection and detection of intrusion in iot. Proc Comput Sci. 2018;132:284–92. https://doi.org/10.1016/j.procs.2018.05.155.
10. Pires P, Govindarajan A, Queiroz C. Enhancing security in wireless sensor networks using genetic algorithms for anomaly detection. Sensors. 2021;21(5):1234. https://doi.org/10.3390/s21051234.
11. Aljawarneh S, Aldwairi M, Yassein MB. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci. 2019;25:193–202.
12. Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE Trans Emerg Topics Comput Intell. 2018;2(1):41–50. https://doi.org/10.1109/TETCI.2017.2772792.
13. Abdelhamid AA, Elhoseny M, Kumar N, Shah MA. A secure framework for industrial internet of things (iiot) networks based on genetic algorithm and deep neural network. IEEE Trans Ind Inf. 2020;16(10):6710–8. https://doi.org/10.1109/TII.2020.2974801.
14. Majeed S, Abbas A, Ullah S. Genetic algorithm optimized deep neural networks for iot security intrusion detection. J Netw Comput Appl. 2021;177: 102998. https://doi.org/10.1016/j.jnca.2020.102998.

15. Sun W, Li Q, Wang P, Hou J. Evolving convolutional neural networks for intrusion detection system using hybrid multi-strategy aquila optimizer. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. Association for Computing Machinery, (2022). https://doi.org/10.1145/3520304.3529021
16. Alzboon K, Al-Nihoud J, Alsharafat W. Novel network intrusion detection based on feature filtering using flame and new cuckoo selection in a genetic algorithm. Appl Sci. 2023. https://doi.org/10.3390/app132312755.
17. Hakan Can Altunay ZA. A hybrid cnn+lstm-based intrusion detection system for industrial iot networks. Engineering Science and Technology, an International Journal 2023;38 https://doi.org/10.1016/j.jestch.2022.101322
18. Shrestha R, Mohammadi M, Sinaei S, Salcines A, Pampliega D, Clemente R, Sanz AL, Nowroozi E, Lindgren A. Anomaly detection based on lstm and autoencoders using federated learning in smart electric grid. J Parallel Dis Comput. 2024;193. https://doi.org/10.1016/j.jpdc.2024.104951.
19. Chahal A, Gulia P, Gill NS, Rani D. Design of a federated ensemble model for intrusion detection in distributed iiot networks for enhancing cybersecurity. J Ind Inf Integr. 2025. https://doi.org/10.1016/j.jii.2025.100800.
20. Ullah S, Mahmoud QH. A scheme for generating a dataset for anomalous activity detection in iot networks. In: Goutte, C., Zhu, X. (eds.) Advances in Artificial Intelligence. Canadian AI 2020. Lecture Notes in Computer Science, vol. 12109. Springer, (2020). https://doi.org/10.1007/978-3-030-47358-7_52
21. Kang H, Ahn DH, Lee GM, Yoo JD, Park KH, Kim HK. IoT network intrusion dataset. IEEE Dataport (2019). https://doi.org/10.21227/q70p-q449
22. Li Q, Li B, Wen L. An intrusion detection model based on feature selection and improved one-dimensional convolutional neural network. Int J Dis Sensor Netw. 2023;1:1982173. https://doi.org/10.1155/2023/1982173.
23. Halim Z, Yousaf MN, Waqas M, Sulaiman M, Abbas G, Hussain M, Ahmad I, Hanif M. An effective genetic algorithm-based feature selection method for intrusion detection systems. Comput Sec. 2021;110: 102448. https://doi.org/10.1016/j.cose.2021.102448.
24. Velumani R, Kalimuthu VK. Barnacles mating optimizer with hopfield neural network based intrusion detection in internet of things environment. Tehnički vjesnik (2023) https://doi.org/10.17559/TV-20230414000533
25. Nguyen MT, Kim K. Genetic convolutional neural network for intrusion detection systems. Fut Gen Comput Syst. 2020;113:418–27. https://doi.org/10.1016/j.future.2020.07.042.
26. Javaid, S.M.A.R.A.A.e.a. A.: Coal mining accident causes classification using voting-based hybrid classifier (vhc). J Ambient Intell Human Comput (2023) https://doi.org/10.1007/s12652-022-03779-z
27. Gomathy, A., Lakshmipathi, B.: Network intrusion detection using genetic algorithm and neural network. In: Wyld, D.C., Wozniak, M., Chaki, N., Meghanathan, N., Nagamalai, D. (eds.) Advances in Computing and Information Technology, vol. 198. Springer, ??? (2011). https://doi.org/10.1007/978-3-642-22555-0_41
28. Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor. 2016. https://doi.org/10.1109/COMST.2015.2494502.
29. Aljawarneh MBYM. Shadi & Aldwairi: Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci. 2017. https://doi.org/10.1016/j.jocs.2017.04.009.
30. A. Khacha, Y.H. R. Saadouni, Aliouat, Z.: Hybrid deep learning-based intrusion detection system for industrial internet of things. 2022 5th International Symposium on Informatics and its Applications (ISIA), M'sila, Algeria (2022) https://doi.org/10.1109/ISIA55826.2022.9993487
31. Dash, C.S.R.A.K.e.a. N.: An optimized lstm-based deep learning model for anomaly network intrusion detection. Sci Rep (2025) https://doi.org/10.1038/s41598-025-85248-z
32. C. Yin, J.F. Y. Zhu, He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access (2025) https://doi.org/10.1109/ACCESS.2017.2762418
33. Shone, N.T.N.P.V.D..S.Q. N.: A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence (2017) https://doi.org/10.1109/TETCI.2017.2772792
34. Zahid Halim, M.W.M.S.G.A.M.H.I.A.M.H. Muhammad Nadeem Yousaf: An effective genetic algorithm-based feature selection method for intrusion detection systems. IEEE Transactions on Emerging Topics in Computational Intelligence (2021) https://doi.org/10.1016/j.cose.2021.102448
35. Mengdi W, Ying W, Cheng L, Kaiyan C, Chengsi G, Yinhe H, Huawei L, Lei Z. Puzzle: a scalable framework for deep learning integrated chips. J Comput Res Dev. 2023;60(6):1216–31. https://doi.org/10.7544/issn1000-1239.202330059.