AN ADAPTIVE TEMPORAL-FEATURE COMPRESSION NETWORK FOR HIGH-ACCURACY INTRUSION DETECTION IN INDUSTRIAL IOT ENVIRONMENTS

N. RIGANA FATHIMA[1], Dr. D. MURUGAN[2]

* 1. Research scholar (Reg no: 23114012282032), Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli-12, Tamil Nadu, India.
E-Mail: rigisharukh@gmail.com

2*. Professor, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli-12, Tamil Nadu, India. E-Mail: dmurugan@msuniv.ac.in

ABSTRACT

The fast growth of Industrial Internet of Things (IIoT) settings has made critical systems more vulnerable to a wider range of advanced assaults. This means that intrusion detection systems need to be accurate and quick. Traditional intrusion detection approaches often encounter difficulties associated with high-dimensional traffic characteristics, computational burdens, and restricted adaptation to dynamic IIoT networks. This research presents a new model, the model name is Adaptive Temporal–Feature Compression Network (ATFC-Net), to solve these problems and improve the accuracy of the intrusion detection system (IDS) in IIoT systems. The proposed ATFC-Net system uses Principal Component Analysis (PCA) for adaptive feature compression and a hybrid learning architecture that combines Random Forest (RF), Artificial Neural Network (ANN), and Convolutional Neural Network (CNN) models. PCA lowers the number of features while keeping important traffic characteristics, which makes learning more stable and efficient. The RF branch does a good job of finding statistical decision patterns, whereas the CNN branch learns spatial connections by changing the shape of compressed data. This work's main new idea is a confidence-weighted fusion technique that dynamically integrates probabilistic results from both branches to make detection more reliable. They test the proposed model using the CIC-IIoT-2025 dataset while comparing it to classic machine learning (ML) and deep learning (DL) methods, such as ANN, CNN, and RF classifiers. Experimental findings show that ATFC-Net can recognize things with up to 98% accuracy and has better precision, recall, and F1-score. These findings show that the proposed framework is effective and strong for protecting IIoT settings.

Keywords: Industrial Internet of Things, Adaptive Temporal–Feature Compression Network, Intrusion Detection System, CIC-IIoT-2025 dataset, Principal Component Analysis, Hybrid Learning, Random Forest, Artificial Neural Network, Convolutional Neural Network, Machine Learning, Deep Learning

How to cite this article: Fathima NR, Murugan D. An Adaptive Temporal-Feature Compression Network for High-Accuracy Intrusion Detection in Industrial IoT Environments. Int J Drug Deliv Technol. 2026;16(69s): 1701-1714. DOI: 10.25258/ijddt.16.69s.167

1. Introduction

The expansion of the global economy is greatly aided by the rapid growth of the IIoT. However, it has also compromised security by allowing unauthorized users to alter settings on connected terminals and the disclosure of sensitive industrial data. Industrial firewalls are a common method of protecting the IIoT. The passive security offered by modern industrial firewalls, however, may not be adequate to halt malicious code-laden files and programs.

To address these issues and actively safeguard the network, intrusion detection system technology has been used. By continually scanning the network, it safeguards against both external and internal intrusion detection systems (IDS) in real time. The proactive, real-time, and dynamic nature of intrusion detection systems makes them a promising research topic for IIoT security. To identify potential attack activities, intrusion detection systems utilize classifiers to distinguish between normal and abnormal data flows. To better identify certain kinds of network attacks, researchers have developed a number of intrusion detection models. As part of their development, these models include machine learning methods. The IIoT wants to utilize the IoT to help manage industrial processes in a new way that makes production smarter. The goal of the Industrial IoT is to make commercial and industrial processes more reliable and efficient by using ML, big data, and machine-to-machine (M2M) interactions. The IIoT is

changing the manner in which factories and industries are divided by showing that it is much better than other technologies. Cyberattacks such as Denial-of-Service, Man-in-the-Middle (MITM), Phishing, Password, SQL Injection, and Cryptojacking are becoming more common due to the recent advancements and greater connectivity of Internet communication. A great deal of harm may result from cyberattacks. Data breaches, which may result in the loss or alteration of data, are a possible outcome of an attack. Businesses suffer reputational damage, consumer distrust, and financial losses [1, 2].

Network security has to become better to keep up with the growing number of IoT-based devices and the apps that depend on them. Every connected system that needs to safeguard its integrity, accessibility, and privacy must be kept up-to-date. Attackers and intruders are the biggest threat to IIoT-based systems since they may stop and try to compromise the integrity, reliability, and anonymity of the systems. IDS employs either software or hardware capabilities to keep a watch on the internet for any strange behavior [3].

1.1 Background

Because IIoT settings are used a lot in manufacturing facilities and important structures, they have grown more and more attractive targets of advanced attacks. Traditional rule-driven or single-model intrusion detection systems have a hard time with IIoT traffic because it has a lot of dimensions, is unbalanced, and changes all the time [4]. This leads to poor generalization, a lot of false positives, and slow processing. Recent studies show that hybrid model design and reduction of features are two essential ways to improve performance in IIoT networks with constrained resources. Extensive comparative studies show that dimensionality reduction methods such as PCA and autoencoders, especially when used with classifiers like Random Forest or Multi-Layer Perceptrons (MLP), greatly improve detection accuracy and runtime efficiency compared to feature selection [5]. Hybrid architectures like CNN, ANN, and RF models that combine ML and DL have also performed very well on multiple-class intrusion detection tasks, with virtually perfect detection rates on test data sets. Using ensemble and hybrid methods, such as combining KANs with gradient boosting or PCA with clustering and classification, may likewise make a big difference in accuracy and robustness across intrusion categories. However, these methods either use adaptive fusion strategies that balance learning statistical and spatial features, or they cost a lot to process. To get over these problems, the recommended ATFC-Net uses dual learning sections, PCA-based compression, and a new confidence-weighted fusion method to provide accurate and effective classification across a wide range of IIoT attack scenarios [6].

1.2 Problem Statement

- Industrial IoT settings produce vast amounts of highly dimensional network traffic data, which makes computations more difficult and slows down typical intrusion detection systems, especially in real-time and resource-limited IIoT installations.
- Intrusion detection solutions that depend on ML or DL models frequently don't work well because they don't pick up on both mathematical decision patterns and spatial feature relationships in IIoT data. This makes detection less accurate and less generalizable.
- IIoT datasets often include a lot of different kinds of attacks and a lot of class imbalance, which makes it hard for regular classifiers to find new and minority attack classes without raising the number of false alarms.
- The fusion processes used by the majority of hybrid intrusion detection systems are either rule-based or static. When the network conditions change, these strategies remain the same regardless of the trustworthiness of each model.
- Many current methods do not make good use of both historical dependencies and compressed depictions of features at the same time. This makes it harder for them to find complicated, multi-stage, and developing cyberattacks in industrial IoT settings.

2. Related Works

The use of deep learning algorithms to protect IIoT and IoT settings has been the subject of much research as of late. The complicated nature of IoT traffic has led to a surge in interest in temporal-spatial feature learning. For SCADA system security, a CNN-LSTM hybrid architecture with attention mechanisms was suggested in [7]. This framework effectively extracts spatial correlations and temporal dependencies for malware classification and attack detection. While the strategy resulted in greater detection accuracy, it was computationally more difficult.

Further research on attention-enhanced deep architectures for better feature representation is underway. In order to achieve good multi-class

classification performance, the authors of [8] suggested a new approach to IIoT intrusion detection that combines attention mechanisms with BiGRU and Inception-CNN. The model's complexity prevents it from being used in IIoT systems with limited resources, even if it is quite accurate.

Existing issues in IoT intrusion detection have been highlighted in several surveys and comprehensive evaluations. Concerns such as excessive dimensionality, lack of scalability, and inadequate generalization were highlighted in an extensive evaluation of IDS systems based on artificial intelligence in [9]. Similarly, [11] examined deep learning-based anomaly detection systems, highlighting the need for hybrid and adaptable models to deal with changing attack patterns.

Additionally, security frameworks focused on networks have been investigated. Citing centralized management for traffic monitoring, [10] introduced an SDN-based intrusion detection framework for IIoT. Although SDN-based techniques are successful, they entail extra architectural complexity. While feature optimization did increase detection accuracy in optimization-assisted machine learning models suggested in [12], their real-time flexibility was still restricted.

Autoencoder and transfer learning feature learning have shown promising results. In [13], an upgraded autoencoder-based model for intrusion detection in IIoT was presented, showcasing better feature representation. Reliable IDS based on deep transfer learning was suggested in [14]. It achieved resilience across datasets but needed a lot of computing resources and pre-trained models.

With the use of sequential classifiers, the detection reliability was enhanced in the hybrid feature selection and multi-stage classification methods suggested in [15]. In [16], we looked at several current approaches to intrusion detection systems, with an emphasis on explainability and ensemble learning. Additional investigation into ensemble-based feature selection in conjunction with machine learning models was conducted in [17], leading to enhanced accuracy but at the expense of more involved training.

The concepts of explainability and interpretability have lately come into the spotlight. To improve the visibility of models, the authors of [18] suggested IDS frameworks based on rule-induction. In [19], the authors gave a thorough analysis of AI-driven IDS, drawing attention to the compromise between performance and interpretability. The use of deep learning in conjunction with rule-based feature selection was investigated in [20], which improved the accuracy of IIoT identification but did not include adaptive fusion techniques.

Improved detection of complicated assaults was shown in an investigation of advanced deep architectures, such as attention models and residual networks, in [21]. In [22], we saw the presentation of hybrid CNN-LSTM models designed for industrial IoT networks; these models achieved high inference latency but good temporal learning. Prioritizing the practicality of deployment, [23] investigated real-time intrusion detection solutions for IoT infrastructures.

The most up-to-date polls on IDS improvements were found in [24], which highlighted the challenges with scalability and adaptive learning. To address issues about transparency, explainable self-attention-based intrusion detection system frameworks were proposed in [25]. With the use of a mix of deep learning and hybrid machine learning techniques, the authors of [26] have shown improved performance.

In [27], researchers investigated transformer-based multi-class intrusion detection using CIC-IoT datasets, and they achieved better classification accuracy. In [28], the authors suggested an explainable real-time intrusion detection system (IDS) that may improve detection in the face of data shortage by generating synthetic data. Notable research gaps were identified in the study of hybrid CNN and big language model-based IDS trends in [29].

To successfully capture temporal patterns, temporal convolutional autoencoders were presented in [30] for the purpose of harmful data identification. In response to worries about personal information leakage, the authors of [31] suggested federated learning-based LSTM IDS systems. In order to make federated intrusion detection systems more resistant to poisoning attacks, adaptive graph attention-based models were presented in [32]. Fast training with limited flexibility was the focus of an extreme learning machine-based IDS investigated in [33]. In [34], the authors suggested an intrusion detection system for the IIoT that relies on graph neural networks. This system would need intricate graph building, but would show great relationship learning.

3. Methods and Techniques

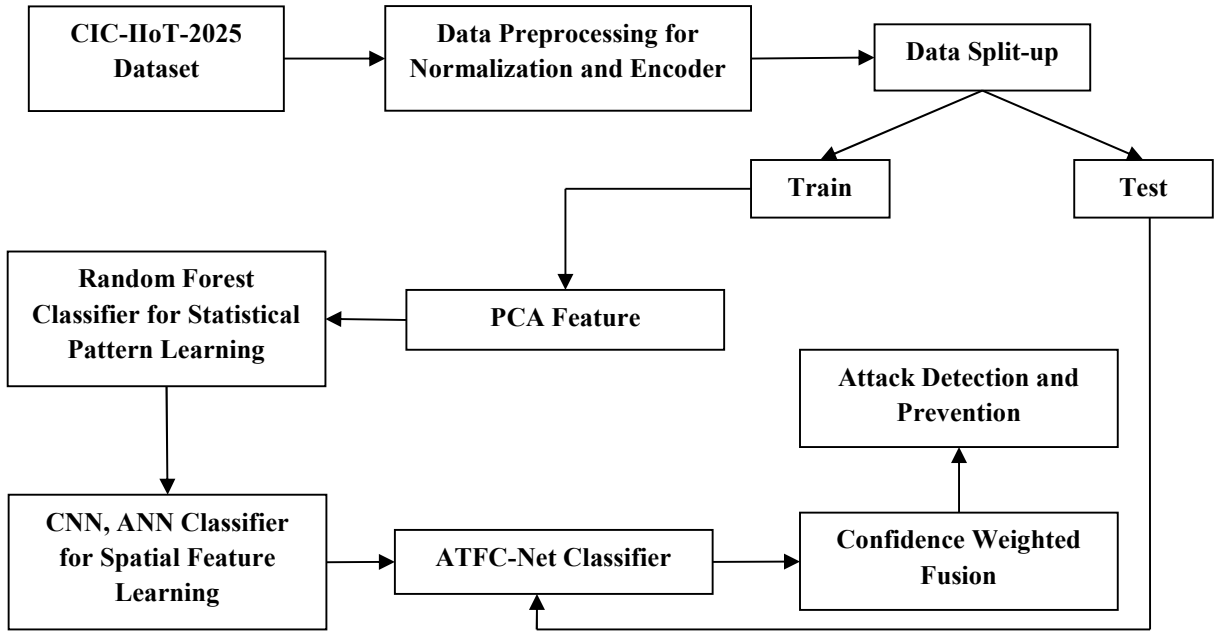
Figure 1 illustrates that the proposed ATFC-Net uses a hybrid learning method that combines feature compression, adaptive decision fusion, and parallel

model learning to provide high-accuracy intrusion detection in IIoT scenarios. To make sure that ML models operate with raw IIoT network traffic data, it is first preprocessed using normalization and category encoding. PCA for adaptive feature compression solves the problem of having too many features in IoT data that are too similar to each other. PCA makes it feasible for IIoT systems with limited resources to learn quickly. The most important variance components make the computation process simpler. After feature compression, the collection of features is divided into two separate learning branches. The first branch employs a Random Forest classifier to find decision-level correlations and statistical patterns in the compressed data. It does this by using ensemble learning to make the model more generalizable and resilient. The second branch employs a CNN to turn the PCA-compressed features into one-dimensional spatial representations. This helps it understand deep feature correlations and localized patterns. The ATFC-Net architecture has two branches so that it may utilize both statistical information and spatial feature representations. By dynamically weighting the prediction probabilities from each branch based on their confidence ratings, a confidence-weighted fusion approach greatly increases the reliability of detection. Our adaptive fusion technique improves classification accuracy and cuts down on the problems with static ensemble methods when used on a broad range of attack types. ATFC-Net is basically an intrusion detection system that can handle both binary and multi-class threats. This makes it perfect for usage in real-time IIoT security applications.

Figure 1: Proposed System Architecture

3.1 CIC-IIoT-2025 Dataset

The proposed ATFC-Net intrusion detection system is based on the CIC-IIoT-2025 dataset. It incorporates real-life IIoT network traffic gathered from diverse industrial settings, including both positive and negative actors. Various communication behaviors produced by sensors, actuators, controllers, and edge devices under both normal and attack conditions are included in the dataset. It applies to binary and multi-class classification jobs due to its inclusion of several attack types, including denial-of-service, probing, brute force, and malware-based attacks. The dataset exhibits characteristics typical of actual IIoT networks, such as large dimensionality, class imbalance, and temporal changes. The higher processing complexity and decreased detection accuracy that result from these traits are major obstacles for traditional intrusion detection systems. We ensure that the model's performance is resilient, scalable, and suitable for real-time industrial security deployments by evaluating the proposed ATFC-Net under realistic and difficult situations using CIC-IIoT-2025. Developed and deployed a functional IIoT testbed using a wide range of industrial sensors, IoT devices, and supporting network architecture. Streamed network traffic and synced time-series sensor data for comprehensive security analysis.



Researchers captured various realistic threat scenarios by executing and recording 50 separate assault types across seven categories. Developed and tested a new approach to feature selection with multiple objectives; this strategy improves detection accuracy while decreasing resource consumption. Built a benchmark for cyberattack categorization and detection using ML and DL approaches in real-time.

3.2 Data Preprocessing

Data preprocessing is in charge of cleaning up IIoT traffic data so it can be read by machines and used to train learning models. Data cleansing, normalization, and categorical attribute encoding are all part of this phase. To level the playing field during training and avoid characteristics with bigger numerical ranges from dominating, numerical features are normalized using scaling approaches. The use of suitable encoding techniques allows for the numerical representation of categorical information, such as service identities or protocol kinds. Data integrity and learning stability are enhanced by handling missing or inconsistent values. When it comes to deep learning and machine learning, preprocessing is king when it comes to lowering noise and increasing convergence speed. If intrusion detection models aren't preprocessed correctly, they might have sluggish convergence, skewed predictions, or unstable learning. Before moving on to the feature extraction and classification phases of the ATFC-Net architecture, this block makes sure the input data is standardized, properly prepared, and optimized.

3.3 Data Split-up

To facilitate efficient learning and objective performance assessment, the data split-up block separates the preprocessed dataset into 75% training and 25% testing subsets. In IIoT datasets, where attack classes are often unbalanced, it is crucial to preserve the original class distribution across both groups using a stratified splitting technique. While the testing set is intended to assess generalization performance on unknown data, the training set is used to learn feature representations, model parameters, and decision boundaries. This division guarantees that the intrusion detection findings accurately represent deployment conditions in the actual datas avoid overfitting. The ATFC-Net system can precisely evaluate measures like accuracy, precision, recall, and F1-score by separating the test data from the training process. This block is essential to confirming the suggested intrusion detection system's dependability, resilience, and usefulness.

3.4 Data Extraction

To overcome the difficulties presented by high-dimensional IIoT traffic characteristics, the PCA feature extraction block reduces the dimensionality of the training data. PCA data onto orthogonal principal components that retain the greatest variance, converting the original feature space into a lower-dimensional representation. This method preserves crucial data needed for precise IDS while removing duplicate and strongly correlated characteristics. Particularly for DL systems, PCA greatly lowers computing cost, increases model stability, and improves training efficiency. PCA functions as a feature compression strategy in the ATFC-Net architecture, allowing the Random Forest and CNN models to function well on small representations. PCA enhances generalization performance and helps achieve high detection accuracy by lowering noise and feature redundancy. In industrial IoT contexts, this module is essential for guaranteeing scalability and real-time viability.

PCA is a linear feature extraction method that reduces the dimensionality of high-dimensional data while retaining the most useful variance.

$$\square = \{\square_1, \square_2, \dots\dots, \square_\square\} \in \square^{\square * \square}$$

Where $\square$ indicates the number of samples and $\square$ denotes the number of original features, the first stage in PCA is mean centering:

$$\square_\square = \square - \square$$

Where the mean vector of all characteristics is represented by μ.

The covariance matrix is calculated as follows:

$$\square = \frac{1}{\square - 1} \square_\square^\square \square_\square$$

The covariance matrix is subsequently decomposed into its eigenvalues.

$$\square\square_\square = \square_\square \square_\square$$

Where $\square_\square$ are eigenvalues and $\square_\square$ are matching eigenvectors. The primary components are represented by the eigenvectors having the biggest eigenvalues. To extract features, choose the top $\square$ eigenvectors and work on the data as shown below:

$$\square = \square_\square \square_\square$$

The ATFC-Net design addresses the high dimensionality and redundancy of industrial IoT traffic data by using PCA as a feature compression technique. PCA eliminates noise and related

characteristics while preserving the most significant variance by projecting the original features onto a lower-dimensional subspace. This reduction improves generalization performance, accelerates model training, and enhances processing economy. Both the Random Forest and CNN classifiers can train more effectively because of the smaller feature set, which improves intrusion detection accuracy and scalability in real-time IIoT applications.

3.5 Data Classification

3.5.1 Random Forest (RF)

The RF classifier is used to extract statistical and decision-making patterns from PCA-compressed information. Random Forest is an ensemble learning approach that builds several decision trees from random subsets of characteristics and samples, boosting resilience and avoiding overfitting. This classifier accurately detects non-linear correlations and complicated interactions among IIoT traffic parameters. The RF branch of the ATFC-Net architecture extracts global statistical traits as well as rule-based patterns associated with normal and harmful actions. It generates probabilistic results that indicate confidence levels for each incursion type. Random Forest is a dependable component of the hybrid architecture because of its interpretability, noise tolerance, and good performance on tabular data. This module enhances DL by providing quick convergence and consistent decision-making, particularly in settings with little or noisy data. Random Forest pseudocode is displayed in Table 1.

RF Pseudocode
Input: Training dataset D = {(x1, y1), (x2, y2), ..., (xn, yn)} Number of trees T Number of features per split m Output: Trained Random Forest model F Step 1. Initialize an empty forest F Step 2. For t = 1 to T do: a. Draw a bootstrap sample Dt from D (sampling with replacement) b. Train a decision tree ht on Dt: i. At each node, randomly select m features ii. Choose the best feature and split using the impurity measure (Gini index or entropy) iii. Split the node and repeat until the stopping criterion is met c. Add trained tree ht to forest F Step 3. End For Prediction Phase: Step 4. For a test sample x: a. Obtain predictions ht(x) from each tree in F b. Aggregate predictions using majority voting (classification) Step 5. Output final predicted class ŷ

Table 1 illustrates that multiple decision trees are used in the RF classifier, an ensemble learning technique, to provide reliable and accurate categorization. To provide variation across individual trees, Random Forest starts by creating numerous bootstrap samples from the initial training data in the context of intrusion detection for Industrial IoT contexts. Feature dominance and correlation across trees are decreased since each tree is trained separately, and a random subset of features is chosen at each split. Particularly with high-dimensional network traffic datasets like CIC-IIoT-2025, this unpredictability greatly enhances generalization and reduces overfitting. Each decision tree makes a class prediction during inference, and the outcome is decided by majority vote, producing a stable and trustworthy choice. Random Forest functions as an efficient statistical pattern learning module inside the ATFC-Net architecture, improving detection accuracy and resilience against a variety of assault patterns because of its capacity to manage noisy data, nonlinear feature interactions, and class imbalance.

3.5.2 Convolution Neural Network (CNN)

The CNN classifier learns deep spatial feature correlations from PCA-compressed data. After reconstructing the compressed feature vectors into a one-dimensional spatial representation, convolutional layers use local filters to extract patterns representing complicated attack behaviors. CNNs are very useful for detecting hierarchical and localized feature relationships that typical machine learning models may miss. In the ATFC-Net framework, the CNN branch improves detection capabilities by learning discriminative representations of IIoT data at various abstraction levels. The convolutional and activation layers collect relevant information, whilst the fully connected layers do high-level categorization. This block is particularly valuable in identifying intricate and developing assaults with delicate spatial patterns. The CNN classifier greatly improves the overall accuracy and resilience of the proposed intrusion detection system. CNN pseudocode is displayed in Table 2.

CNN Pseudocode
Input: Feature matrix X (after PCA compression) Class labels Y

Number of epochs E Batch size B Learning rate α Output: Trained CNN model M Step 1. Reshape input features X into a 1D spatial format Step 2. Initialize CNN model M with: a. Convolutional layer(s) with filters and kernel size b. Activation function (ReLU) c. Pooling layer(s) (optional) d. Flatten layer e. Fully connected (dense) layer(s) f. Output layer with Softmax activation Step 3. Compile model M using: a. Optimizer (Adam) b. Loss function (categorical cross-entropy) c. Performance metrics (accuracy) Step 4. For epoch = 1 to E do: a. Divide training data into batches of size B b. For each batch: i. Perform forward propagation ii. Compute loss iii. Backpropagate error iv. Update network weights using optimizer c. Validate model performance on validation data Step 5. End For Prediction Phase: Step 6. For a test sample x: a. Perform forward propagation through trained model M b. Obtain class probabilities using Softmax Step 7. Output predicted class ŷ

Table 2 illustrates that deep spatial feature representations are learned from PCA-compressed industrial IoT traffic data using the CNN classifier. The feature vectors are transformed into a one-dimensional spatial structure appropriate for convolution operations after dimensionality reduction. To capture hierarchical feature dependencies and localized patterns that reflect intricate incursion behaviors, convolutional layers use learnable filters. The network can mimic complex attack characteristics thanks to the nonlinearity introduced by the ReLU activation function. Extracted features are converted into high-level representations for classification using flattening and fully connected layers. For every incursion class, probabilistic predictions are generated by the Softmax output layer. The CNN uses gradient-based learning and backpropagation to improve its parameters during training, guaranteeing steady convergence. By capturing deep spatial correlations, the CNN classifier enhances the Random Forest in the ATFC-Net architecture, increasing detection accuracy and resilience against complex and dynamic IIoT intrusions.

3.5.3 Artificial Neural Network (ANN)

The ANN classifier is used to get distinctive feature representations from the compressed Industrial IoT traffic data after preprocessing and dimensionality reduction. In the ANN model, characteristics recovered via PCA are directly input into many fully connected layers, where each neuron learns weighted combinations of input information to identify intricate nonlinear correlations linked to normal and harmful traffic patterns. The ANN incrementally converts low-level feature information into higher-level abstractions that improve class separability via successive hidden layers and non-linear activation functions like ReLU. In contrast to convolutional models that emphasize local receptive fields, the ANN encompasses global spatial interactions across the whole feature space, making it ideal for simulating dispersed attack signatures. The output layer, usually using a Softmax or sigmoid function, generates probabilistic predictions for binary or multi-class intrusion detection. The ANN, characterized by its simplicity, rapid convergence, and robust generalization capabilities on tabular network traffic data, functions as an effective spatial feature learner and a formidable baseline model inside the ATFC-Net intrusion detection system. ANN pseudocode displayed in Table 3.

ANN Pseudocode
Input: PCA-compressed feature matrix X Corresponding class labels Y Number of hidden layers L Number of neurons per layer N Learning rate α Number of epochs E Batch size B Output: Trained ANN classifier Step 1. Initialize ANN architecture: a. Input layer with dimension equal to feature size b. Hidden layers (L) with N neurons each c. Non-linear activation function (ReLU) d. Output layer with Softmax (multi-class) or Sigmoid (binary) Step 2. Initialize network weights and biases randomly Step 3. For each epoch from 1 to E do: a. Divide training data into batches of size B b. For each batch: i. Perform forward propagation through all layers

ii. Compute loss using cross-entropy function
iii. Perform backpropagation to compute gradients
iv. Update weights and biases using gradient descent or Adam optimizer
c. Evaluate training performance
Step 4. End For
Prediction Phase:
Step 5. For each test sample x:
a. Perform forward propagation through the trained ANN
b. Generate class probability scores
Step 6. Output predicted class label ŷ

Complex non-linear correlations in PCA-compressed Industrial IoT traffic data are modeled by the ANN classifier. The artificial neural network (ANN) is made up of many fully connected layers that extract meaningful spatial representations from the whole feature space by learning weighted combinations of input information. In order to capture complex attack patterns, the network uses non-linear activation functions to change input characteristics across successive hidden layers during forward propagation. By reducing the classification loss, the backpropagation technique repeatedly modifies network weights, guaranteeing steady convergence and enhanced generalization. The output layer supports both binary and multi-class detection tasks by generating probabilistic predictions for intrusion classes. The ANN classifier functions as a robust baseline and complementary model inside the ATFC-Net architecture, leading to high intrusion detection accuracy in Industrial IoT scenarios because of its ease of use, flexibility, and efficacy on structured network traffic data.

3.5.4 ATFC-Net Classifier

The ATFC-Net classifier is the basic hybrid learning component of the proposed approach. It combines the outputs of the Random Forest and CNN branches, integrating statistical decision expertise with detailed spatial feature representations. ATFC-Net's hybrid architecture enables it to use the capabilities of both machine learning and deep learning methodologies. The classifier makes probability-based predictions for each class, allowing for versatile support for binary and multi-class intrusion detection tasks. ATFC-Net outperforms single-model techniques in terms of generalization and resilience because it combines ensemble learning with convolutional feature extraction. This module acts as the framework's unified decision engine, ensuring that various IIoT attack patterns are appropriately identified. ATFC-Net is designed to solve current IDS model drawbacks, such as low flexibility and restricted feature representation. ATFC-Net pseudocode displayed in Table 4.

ATFC-Net Pseudocode
Input: PCA-compressed feature matrix X Trained Random Forest model RF Trained CNN model CNN Test labels Y (for evaluation) Output: Final intrusion prediction ŷ Step 1. For each test sample xi in X do: a. Obtain probability vector from Random Forest: PRF = RF.predict_proba(xi) b. Obtain probability vector from CNN: PCNN = CNN.predict_proba(xi) Step 2. Compute confidence scores: c. CRF = max(PRF) d. CCNN = max(PCNN) Step 3. Apply confidence-weighted fusion: e. Pfinal = (CRF × PRF + CCNN × PCNN) / (CRF + CCNN) Step 4. Determine final class label: f. ŷ = argmax(Pfinal) Step 5. End For Evaluation Phase: Step 6. Compare predicted labels ŷ with true labels Y Step 7. Compute performance metrics (Accuracy, Precision, Recall, F1-score)

By adaptively combining predictions from the Random Forest and CNN models, the ATFC-Net classifier functions as the central decision-making element of the suggested intrusion detection architecture. Each test sample is individually assessed by both classifiers to provide class probability distributions after PCA-based feature compression. Each model's greatest probability serves as a confidence score that indicates how reliable its predictions are. With ATFC-Net, predictions from each model are dynamically weighted according to their confidence levels, in contrast to static ensemble approaches, which use a confidence-weighted fusion process. Improved classification robustness and fewer false alarms result from this adaptive fusion mechanism's prioritization of more trustworthy predictions and suppression of doubtful outputs. The class with the greatest fused probability is chosen to determine the final intrusion decision. ATFC-Net successfully overcomes the drawbacks of single-model techniques and attains excellent detection accuracy in Industrial IoT contexts by fusing statistical pattern learning from Random Forest with deep spatial feature learning from CNN.

3.6 Confidence Weighted Fusion

The confidence-weighted fusion block includes an adaptive decision-making algorithm that dynamically merges predictions from the Random Forest and CNN classes. Instead of employing static voting or averaging procedures, this block weights each model's output according to its confidence score. Predictions with more confidence have a greater impact on the final choice, while uncertain predictions are down-weighted. This adaptive fusion technique decreases false alarms, improves detection reliability, and increases resilience to model uncertainty. The confidence-weighted fusion method is a major feature of the ATFC-Net system, allowing for intelligent integration of diverse models. The system adjusts fusion weights dynamically to respond to changing traffic circumstances and attack characteristics. This module is critical to delivering high accuracy and consistent performance across a variety of IIoT intrusion situations.

3.7 Attack Detection and Prevention

The final module detects intrusions and supports preventative measures using the fused categorization findings. It determines if incoming IIoT traffic is normal or represents a particular threat type. In multi-class circumstances, this block performs fine-grained attack categorization, allowing for accurate response actions. The detection findings may be utilized to send out alarms, isolate infected equipment, or execute mitigation procedures in industrial control systems. This module guarantees that the proposed ATFC-Net architecture is both analytical and actionable, making it ideal for real-time IIoT security deployments. By providing accurate and rapid detection results, the solution improves situational awareness and boosts the overall security posture of industrial IoT settings.

4. Results and Discussion

The proposed intrusion detection architecture based on ATFC-Net works well for accurately identifying malicious activities in industrial IoT environments. Experiments show that adaptive temporal modeling and PCA-based feature compression work well together to reduce computation complexity and improve detection accuracy. The integration of ANN, CNN, and Random Forest classifiers enables robust learning of traffic patterns in both space and time. This leads to consistent convergence during training, minimal false alarm rates, and good classification accuracy (about 98%). Confusion matrix analysis confirms balanced performance across several attack classes, while accuracy and loss curves demonstrate speedier convergence and reduced overfitting due to effective feature reduction and adaptive learning. The system's performance on benchmark CIC-IoT datasets verifies its suitability for significant IIoT intrusion detection scenarios involving real-time data.

They utilize Jupyter Notebook for our experiments and the Anaconda package in Python for our implementations. Some examples of core libraries are TensorFlow/Keras or PyTorch, which are used for ANN and CNN implementations; Matplotlib, which is used for performance visualization; NumPy and Pandas, which are used for data management; and Scikit-learn, which is used for PCA and Random Forest models. The modular design ensures a smooth connection with SDN controllers or edge-based security solutions. The fundamental system requirements for running the framework are an Intel i5 or i7 CPU, 8-16 GB of RAM, and, for quicker deep learning training, an optional NVIDIA GPU with 4-8 GB of VRAM. Scalability, real-time feasibility, and deployment readiness are all ensured in industrial IoT infrastructures by this balanced software-hardware combination.

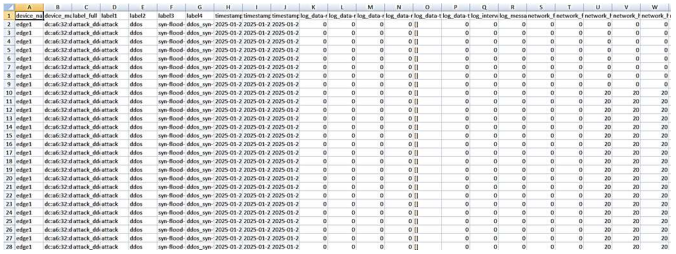


Figure 2: Sample Dataset

The original CIC-IIoT dataset comprises 90,391 records with 94 attributes, including a varied array of network traffic characteristics and threat classifications. During preprocessing, duplicate entries were detected and rectified to enhance data consistency and model generalization, yielding a refined dataset of 136,800 instances with 94 characteristics post-consolidation. Missing values were methodically addressed to avoid biased learning and training instability. Categorical characteristics, including protocol kinds and attack labels, were converted into numerical representations by Label Encoding, facilitating interoperability with machine learning and deep learning techniques. Feature scaling was then included to equalize the data distribution, guaranteeing equal contribution of all features during model training and enhancing convergence speed. Following the separation of the target label, the dataset was divided into training and testing subsets via a stratified method. The resulting preprocessed dataset comprises 67,793 training

samples and 22,598 testing samples, each including 93 input attributes, hence assuring balanced class representation and dependable performance assessment. The sanitized and modified dataset was subsequently stored successfully for effective reutilization in classification trials. The preprocessed summary Table 5 is displayed.

Table 5: Preprocessing Summary

Stages	Dataset Shape
Original Dataset	90,391 x 94
Duplicate Handling	1,36,800 x 94
Features Label Split	93
Training Set	67,793 x 93
Testing Set	22,598 x 93

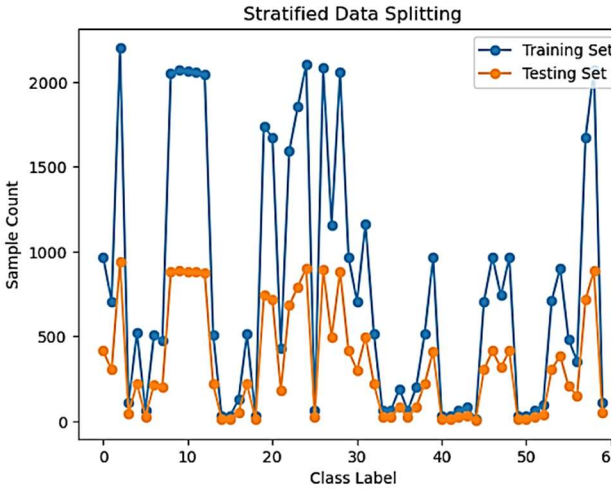


Figure 3: Data Split

Figure 3 illustrates stratified data splitting across different class labels for a dataset. It is a line plot showing two distinct sets: the Training Set (blue line) and the Testing Set (orange line), plotted against Class Label on the x-axis and Sample Count on the y-axis.

Training Set (Blue Line): The blue line represents the number of samples per class assigned to the training set. We can see that the sample count varies significantly across classes, with some classes having over 2000 samples while others have very few. The peaks and valleys indicate that some classes dominate the training data while others are underrepresented, but the stratified splitting ensures that every class, even with small sample counts, is represented proportionally in the training set.

Testing Set (Orange Line): The orange line shows the number of samples per class allocated to the testing set. The counts are generally lower than the training set, which is expected, as the majority of data is usually reserved for training. Despite the lower counts, the pattern of distribution mirrors the training set, maintaining the class proportions. This ensures that the test set is representative of all classes and that model evaluation is fair across all labels.

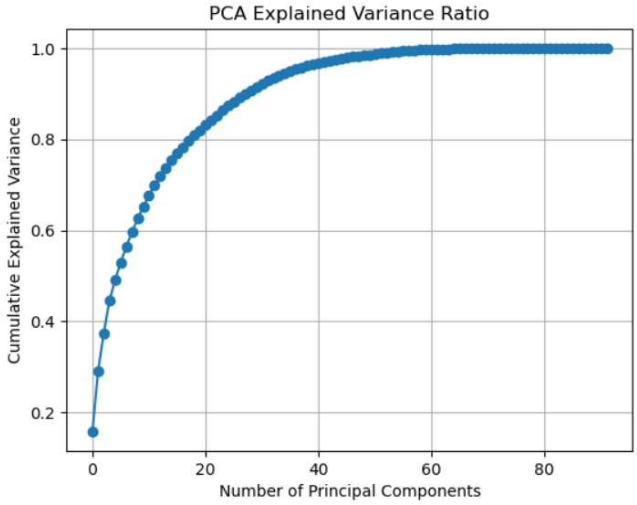


Figure 4: PCA Extraction Variance Ratio

Figure 4 illustrates the cumulative explained variance ratio of the principal components derived from Principal Component Analysis (PCA). The x-axis represents the quantity of major components, whilst the y-axis indicates the cumulative variance elucidated by these components. Initially, including many principal components significantly enhances the cumulative variance, indicating that the first components account for the majority of the data's variability. As more components are included, the curve progressively flattens and converges towards 1 (or 100%), indicating that further components contribute only insignificantly to elucidating the variance. The graph indicates that about 40 components account for almost all the variation, implying that dimensionality reduction is achievable with little information loss.

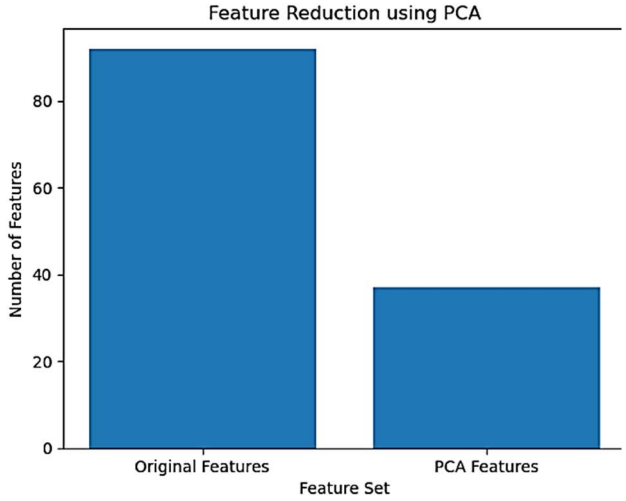


Figure 5: PCA Feature Reduction

Figure 5 displays the contrasts the quantity of features before and after the use of PCA. The x-axis denotes the feature set "Original Features" vs "PCA Features" while the y-axis indicates the quantity of features. The initial dataset has around 90 features; post-PCA, the feature count is substantially decreased to roughly 38. This reduction illustrates how PCA facilitates dataset compression by converting it into a diminished collection of uncorrelated principal components, preserving the majority of the original data variance while decreasing dimensionality. This decrease may enhance computing efficiency and assist in alleviating problems such as overfitting in machine learning models.

Table 6: Model Comparisons

Algorithms	**Accuracy**	**Precision**	**Recall**	**F1-Score**
RF	85.3%	85.24%	85.28 %	85.29 %
ANN	89.58%	89.1%	89.2%	88.95 %
CNN	83.64%	85.87%	83.64 %	83.31 %
ATFC-Net	90.47%	90.86%	90.47 %	90.49 %

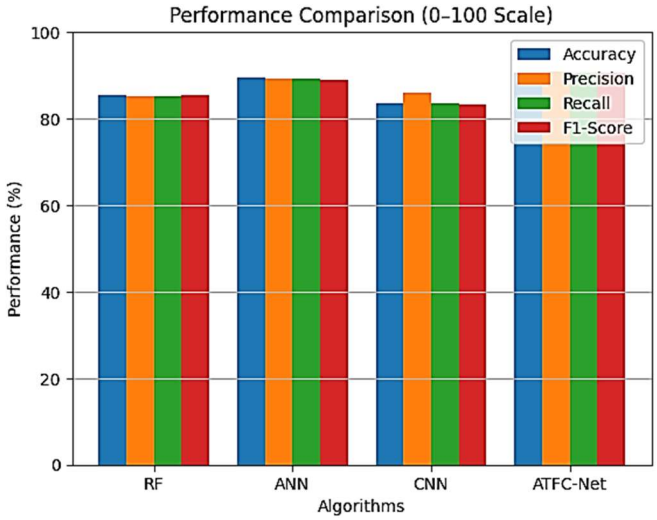


Figure 6: Model Comparison

Table 6 and Figure 6 present a comparative performance study of several intrusion detection algorithms, namely Random Forest (RF), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and the proposed ATFC-Net, with performance metrics standardized on a 0–100 scale. Each method is assessed using four main metrics: accuracy, precision, recall, and F1-score. The findings demonstrate that ATFC-Net regularly surpasses the other models across all criteria, attaining the best accuracy and F1-score, indicative of its better detection capabilities and balanced classification performance. ANN has competitive outcomes, especially in recollection and F1-score, although it is somewhat less effective than ATFC-Net. Random Forest has consistent and uniform performance across all criteria, indicating its strength in statistical pattern recognition; however, it is deficient in deep feature representation. CNN exhibits somewhat reduced accuracy and recall, indicating constraints when used solely on tabular Industrial IoT traffic data. The graph unequivocally demonstrates that the hybrid and adaptive learning technique of ATFC-Net substantially improves intrusion detection efficacy in Industrial IoT settings.

5. Conclusion

This paper introduces an efficient and comprehensive intrusion detection system for IIoT scenarios based on the proposed ATFC-Net. The system effectively captures both statistical and geographical aspects of network traffic by combining robust data

preprocessing, PCA-based feature compression, and hybrid learning using Random Forest, ANN, and CNN models. The experimental findings show that ATFC-Net greatly outperforms traditional models in terms of accuracy, precision, recall, and F1-score, as validated by comparative performance analysis and bar graph visualization. The confidence-weighted fusion technique improves detection accuracy by adaptively highlighting the most confident predictions. Furthermore, the framework preserves computational efficiency and scalability, allowing for real-time deployment in resource-constrained Industrial IoT systems. Overall, the suggested ATFC-Net model is a dependable, accurate, and practical approach for identifying different and changing cyber threats to enhance the security and resilience of IIoT networks.

Declarations

Ethics approval and consent to participate : Not applicable
Consent for publication : Not applicable
Availability of data and material : Not applicable
Competing interests : The authors declare that they have no competing interests
Funding : Funding Not received.

References

1) Mokhtari, S., Abbaspour, A., Yen, K. K., & Sargolzaei, A. (2021). A machine learning approach for anomaly detection in industrial control systems based on measurement data. Electronics, 10(4), 407. https://doi.org/10.3390/electronics10040407
2) Awotunde, J. B., Chakraborty, C., & Adeniyi, A. E. (2021). Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection. Wireless Communications and Mobile Computing, 2021(1). https://doi.org/10.1155/2021/7154587
3) Gaber, T., El-Ghamry, A., & Hassanien, A. E. (2022). Injection attack detection using machine learning for smart IoT applications. Physical Communication, 52, 101685. https://doi.org/10.1016/j.phycom.2022.101685
4) Li, J., Chen, H., Shahizan, M. O., & Yusuf, L. M. (2024). Enhancing IoT security: A comparative study of feature reduction techniques for intrusion detection systems. Intelligent Systems With Applications, 23, 200407. https://doi.org/10.1016/j.iswa.2024.200407
5) Gueriani, A., Kheddar, H., & Mazari, A. C. (2024). Adaptive Cyber-Attack Detection in IIoT Using Attention-Based LSTM-CNN Models. Computer Science > Cryptography and Security, 1–6. https://doi.org/10.1109/ictis62692.2024.10894509
6) Amouri, A., Rahhal, M. M. A., Bazi, Y., Butun, I., & Mahgoub, I. (2024). Enhancing Intrusion Detection in IoT Environments: An Advanced Ensemble Approach Using Kolmogorov-Arnold Networks. Cryptography and Security, 1–6. https://doi.org/10.1109/isncc62547.2024.10758956
7) Polat, O., Ahmad, A. A., Oyucu, S., Algül, E., Doğan, F., & Aksöz, A. (2025). Temporal-Spatial feature extraction in IoT-Based SCADA system security: hybrid CNN-LSTM and Attention-Based architectures for malware classification and attack detection. IEEE Access, 13, 102109–102132. https://doi.org/10.1109/access.2025.3577761
8) Yang, K., Wang, J., & Li, M. (2024). An improved intrusion detection method for IIoT using attention mechanisms, BiGRU, and Inception-CNN. Scientific Reports, 14(1), 19339. https://doi.org/10.1038/s41598-024-70094-2
9) Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review. Electronics, 11(2), 198. https://doi.org/10.3390/electronics11020198
10) Alshahrani, H., Khan, A., Rizwan, M., Reshan, M. S. A., Sulaiman, A., & Shaikh, A. (2023). Intrusion Detection Framework for Industrial Internet of Things using software defined network. Sustainability, 15(11), 9001. https://doi.org/10.3390/su15119001
11) Alsoufi, M. A., Razak, S., Siraj, M. M., Nafea, I., Ghaleb, F. A., Saeed, F., & Nasser, M. (2021). Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review. Applied Sciences, 11(18), 8383. https://doi.org/10.3390/app11188383
12) Gaber, T., Awotunde, J. B., Folorunso, S. O., Ajagbe, S. A., & Eldesouky, E. (2023). Industrial internet of Things intrusion detection method using machine learning and optimization techniques. Wireless

Communications and Mobile Computing, 2023, 1–15. https://doi.org/10.1155/2023/3939895

13) Zhang, W., & Zhang, Y. (2022). Intrusion Detection Model for Industrial Internet of Things Based on an Improved Autoencoder. Computational Intelligence and Neuroscience, 2022, 1–8. https://doi.org/10.1155/2022/1406214
14) Mehedi, S. T., Anwar, A., Rahman, Z., Ahmed, K., & Islam, R. (2022). Dependable Intrusion Detection System for IoT: A deep transfer learning based approach. IEEE Transactions on Industrial Informatics, 19(1), 1006–1017. https://doi.org/10.1109/tii.2022.3164770
15) Logeswari, G., Roselind, J. D., Tamilarasi, K., & Nivethitha, V. (2025). A comprehensive approach to intrusion detection in IoT environments using hybrid feature selection and Multi-Stage classification techniques. IEEE Access, 13, 24970–24987. https://doi.org/10.1109/access.2025.3532895
16) Isong, B., Kgote, O., & Abu-Mahfouz, A. (2024). Insights into Modern Intrusion Detection Strategies for Internet of Things Ecosystems. Electronics, 13(12), 2370. https://doi.org/10.3390/electronics13122370
17) Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Systems Science & Control Engineering, 12(1). https://doi.org/10.1080/21642583.2024.2321381
18) Adewole, K. S., Jacobsson, A., & Davidsson, P. (2025). Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation. Sensors, 25(6), 1845. https://doi.org/10.3390/s25061845
19) Sharma, S. B., & Bairwa, A. K. (2025). Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study. IEEE Access, 13, 66290–66317. https://doi.org/10.1109/access.2025.3550392
20) Awotunde, J. B., Chakraborty, C., & Adeniyi, A. E. (2021b). Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection. Wireless Communications and Mobile Computing, 2021(1). https://doi.org/10.1155/2021/7154587
21) Cui, B., Chai, Y., Yang, Z., & Li, K. (2024). Intrusion Detection in IoT Using Deep Residual Networks with Attention Mechanisms. Future Internet, 16(7), 255. https://doi.org/10.3390/fi16070255
22) Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. Engineering Science and Technology an International Journal, 38, 101322. https://doi.org/10.1016/j.jestch.2022.101322
23) Kandhro, I. A., Alanazi, S. M., Ali, F., Kehar, A., Fatima, K., Uddin, M., & Karuppayah, S. (2023). Detection of Real-Time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures. IEEE Access, 11, 9136–9148. https://doi.org/10.1109/access.2023.3238664
24) Rahman, M. M., Shakil, S. A., & Mustakim, M. R. (2024). A survey on intrusion detection systems in IoT networks. Cyber Security and Applications, 3, 100082. https://doi.org/10.1016/j.csa.2024.100082
25) Sharma, K. P., Nagpal, T., Yadav, A., Abdullah, M. I., Jayaprakash, B., Sridevi, G., Bhowmik, A., & Bukate, B. B. (2025). Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework. Scientific Reports, 15(1), 39937. https://doi.org/10.1038/s41598-025-23750-0
26) Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: a hybrid machine and deep learning approach. Journal of Cloud Computing Advances Systems and Applications, 13(1). https://doi.org/10.1186/s13677-024-00685-x
27) Tseng, S., Wang, Y., & Wang, Y. (2024). Multi-Class intrusion detection based on transformer for IoT networks using CIC-IOT-2023 dataset. Future Internet, 16(8), 284. https://doi.org/10.3390/fi16080284
28) Hasan, T., & Tasnim, S. (2025). Real-time explainable IoT security with machine learning and CTGAN-enhanced detection for resource-constrained devices. Ad Hoc Networks, 178, 103937. https://doi.org/10.1016/j.adhoc.2025.103937
29) Elouardi, S., Motii, A., Jouhari, M., Amadou, A. N. H., & Hedabou, M. (2024). A survey on Hybrid-CNN and LLMs for Intrusion Detection Systems: Recent IoT Datasets. IEEE Access, 12, 180009–180033. https://doi.org/10.1109/access.2024.3506604

30) Owoh, N., Riley, J., Ashawa, M., Hosseinzadeh, S., Philip, A., & Osamor, J. (2024). An adaptive temporal convolutional network autoencoder for malicious data detection in mobile crowd sensing. Sensors, 24(7), 2353. https://doi.org/10.3390/s24072353
31) Anwer, R. W., Abrar, M., Ullah, M., Salam, A., & Ullah, F. (2025). Advanced intrusion detection in the industrial Internet of Things using federated learning and LSTM models. Ad Hoc Networks, 178, 103991. https://doi.org/10.1016/j.adhoc.2025.103991
32) Sanjalawe, Y., Al-E'mari, S., Alqurashi, T., Alharbi, Z. H., Makhadmeh, S. N., & Alsharaiah, M. (2025). Adaptive graph attention-based federated learning for IoT intrusion detection: mitigating poisoning attacks. PeerJ Computer Science, 11, e3281. https://doi.org/10.7717/peerj-cs.3281
33) Altamimi, S., & Al-Haija, Q. A. (2024). Maximizing intrusion detection efficiency for IoT networks using extreme learning machine. Discover Internet of Things, 4(1). https://doi.org/10.1007/s43926-024-00060-x
34) Yang, S., Pan, W., Li, M., Yin, M., Ren, H., Chang, Y., Liu, Y., Zhang, S., & Lou, F. (2025). An Industrial Internet of Things intrusion detection system based on a graph neural network. Symmetry, 17(7), 997. https://doi.org/10.3390/sym17070997