Internet of Things and Cloud Computing
2025, Vol. 13, No. 4, pp. 94-109
https://doi.org/10.11648/j.iotcc.20251304.13

Research Article

Performance Analysis of a CNN-Fuzzy Logic Based Real-time Intrusion Detection for Industrial IoT Systems

Boye Aziboledia Frederick*, Onate Egerton Taylor

Department of Computer Science, Rivers State University, Port Harcourt, Nigeria

Abstract

The Industrial Internet of Things has enhanced automation, real-time monitoring, and predictive decision-making in modern industries. The study explores the mixed research methods (qualitative and quantitative). However, the growing connectivity of industrial IoT systems has exposed them to severe cyber threats such as Ransomware, MitM, and DDoS attacks, which can disrupt critical operations and compromise safety. Conventional Intrusion Detection Systems (IDS) often face limitations in achieving high accuracy, rapid detection, and low latency while minimizing false alarms. This study proposes a CNN-Fuzzy Logic hybrid model for real-time intrusion detection and prevention in industrial IoT environments. Convolutional Neural Networks (CNN) are employed to extract deep hierarchical features from industrial IoT traffic, while fuzzy logic is integrated to enhance decision-making under uncertainty and reduce false positives. The model was trained and evaluated using Kaggle cybersecurity datasets containing ransomware, MitM, and DDoS attacks. Performance evaluation demonstrates that the CNN-Fuzzy IDS achieves an accuracy of 92.5%, a detection rate of approximately 93%, a false positive rate (FPR) of 2.51%, a reduced latency with an average of 7.14% total latency (which corresponds to 1.207 μsec average latency) is very acceptable for most industrial IoT applications. These results highlight the effectiveness of hybrid intelligent systems in enhancing the resilience and reliability of industrial IoT cybersecurity. The proposed model provides a promising pathway for deploying scalable, adaptive, and real-time IDS solutions in critical industrial infrastructures. On system computational overhead researchers should employ a minimum practical setup with modern multi-core CPU, 8–16 GB RAM, SSD, stable OS (Windows 10 only if hardware is modern) or run a lightweight Linux on edge plus offload heavy tasks elsewhere. Future research should also focus on optimizing hybrid ML architectures for low performance metrics for deployment of resource-constrained industrial IoT devices, integrating the approach for threat detection, and expanding evaluation to real-world industrial environments.

Keywords

Industrial IoT, Intrusion Detection System (IDS), Convolutional Neural Networks (CNN), Fuzzy Logic, Real-time Cybersecurity, ML Metrics

*Corresponding author: aziboledia.boye@rsu.edu.ng (Boye Aziboledia Frederick)

Received: 19 September 2025; **Accepted:** 5 October 2025; **Published:** 26 November 2025

1. Introduction

While industrial IoT technologies enable operational efficiency, they introduce expanded attack surfaces for cyber adversaries. Threats such as ransomware, DDoS, MitM attacks etc. have increasingly targeted industrial IoT networks. Traditional signature-based intrusion detection systems (IDS) have limitations in detecting novel attacks due to their reliance on predefined patterns. Hence, machine learning (ML)-based IDSs have emerged as a promising solution. However, single ML algorithms often struggle to balance detection accuracy and computational performance in industrial IoT environments, where real-time detection is critical. This paper focuses on analyzing the threat detection performance of hybrid ML models for industrial IoT security and provides insights into their applicability for industrial systems. Prior studies have explored ML-based IDS frameworks for IoT and industrial IoT environments. Although numerous studies have explored the application of machine learning (ML) to improve the security of Industrial IoT systems, their distributed nature and critical role in industries kept them increasingly vulnerable to cyber threats such as DDoS, ransomware, and MitM attacks, etc. although various techniques have been employed in the development of network intrusion detection system to safeguard the network against the evolving nature of attack deployed by cyber-criminals [29]. Scholars proposed a study addressed the feasibility of using machine learning approaches to detect intrusions in the IoT home dataset (IoTID20) [1]. The study proposed machine learning methods, such as the linear algorithm, random forest, gradient boost algorithm, and many more, over the dataset to identify anomalies with high accuracy. The authors [2] highlighted several challenges on model creation, deployment, and retraining and turning. An anomaly detector must have ultra-high detection rate as well as ultra-low rate of false alarms.

1.1. Research Gap on Dataset Performance Metrics

The proposed hybrid intrusion detection model (CNN-BiLSTM) integrates the CNN and the Bi-directional long short-term memory (BiLSTM), to learn spatial and temporal features. Two datasets NSL-KDD and UNSW-NB15 are used to evaluate their proposed model. The CNN-BiLSTM model achieved an overall accuracy of 82.74% and 77.16% for NSL-KDD and UNSW-NB15, respectively [21]. The authors tested this method for DDoS detection using the most up-to-date Canadian data set (CIC IDS 2017). The authors use the Multi-Layer Perceptron, Convolutional Neural Network, and Deep Auto Encoders to create process anomaly detectors [22]. The scholar [3] listed several techniques in preventing MitM attacks, among the list of best prevention practices was to Ensure the right Tools. This study agrees with [4, 5] and [3] publications that intrusion detection and prevention as a tool will be a viable, efficient technique and defense layer mechanism in the industrial IoT network. [6] proposed an intrusion detection system (IDS) based on combining cluster centers and nearest neighbors, with KDD-Cup99 dataset. The dataset was trained with following algorithms, k-Nearest Neighbor (k-NN), Cluster Center and Nearest Neighbor (CANN) and Support Vector Machine with accuracy of 93.87%, 99.76% and 80.65% respectively. [8] proposed Anomaly Detection Based on Profile Signature in Network using Machine Learning Techniques. The algorithms used are Genetic Algorithm, Support Vector Machine and Hybrid Model with KDDCup ’99 dataset. The following accuracy results were obtained, GA gives 84.0333%, SVM results 94.8000% and Hybrid (GA+SVM) gives 98.333%, showing a low false positive rate (FPR). The authors [9] worked on Intrusion detection in computer networks using hybrid machine learning techniques using Hybrid model of supervised Neural Network, Support Vector Machine and unsupervised (K-Means) machine learning algorithms with NSL-KDD datasets. The authors, [10], proposed an anomaly-based network intrusion detection using ensemble machine learning technique. The study accuracy results in that the ensemble gives 85.20%. The work handled imbalanced data and selected only required features which greatly helped in reducing high false positive [11], appends that ensemble and hybrid classifiers have better predictive accuracy and detection rate than single classifiers. [12], explore detecting intrusions in computer network traffic with Machine Learning Approaches using both single and assembler classifiers, K-Nearest Neighbor (KNN) and ensemble technique. The model was evaluated using two different datasets and having a drawback of failing to address the problem of data high dimensionality. [4] detailed how organizations, including industrial IoT industries, should protect themselves from Ransomware attacks. The author implements multiple layers of defense to reduce the risk of a ransomware attack occurring, or to minimize the impact if an attack occurs, which includes Network Perimeter Defenses (NPDs), such as firewalls, network segmentation and intrusion detection & prevention (ID/IPs) systems (Managed Security Service Provider (MSSP) or integration with the SIEM), are effective at blocking malware before it enters the corporate network. The [13] lists of four (4) guards against Ransomware, includes intrusion detection and prevention systems (ID/PS) and that organizations, like industrial IoT firms outsource to a managed detection and response (MDR) specialist. MDR Services include monitoring, detecting, alerting, and managing responses to potential attacks on your system. The authors [15] propose a model to detect and track malicious URLs using machine learning classifiers and deep learning approaches. To enhance and secure the efficiency of our model, a novel dataset called Ransomware Detection Dataset (RDD) has been introduced [16]. The scholars [30] applied two experiments on CIC and Mal 2017 dataset to analyze six ML techniques (DT, RF,

Random Tree (RT), k-NN, NB, and SVM) for ransomware detection. Firstly, a dataset was applied with different forms and classes of ransomware on ML classifiers. Then they were applied to 10 ransomware families separately on classifiers. The results show that RF was the best in both experiments. In fact, research articles on cybersecurity and ransomware started getting published around the year 2016. There was research offering a defense plan to protect oil and gas automation and control systems [30]. The highest detection accuracy belongs to RF with accuracy score 83%, and 79% for DT. The accurate and timely detection of DoS, DDoS etc. attacks remain a priority for researchers in the field of cybersecurity, however, attackers keep modifying and developing new attacks to evade detection techniques by [31].

1.2. Research Gap on Hybrid Machine Learning Metrics

The authors [17] introduced a detection model using dynamic machine learning techniques, such as conversation-based network traffic features, for consistent detection of windows ransomware network attacks. The authors of [18] implemented a network-based intrusion detection system, by employing two independent classifiers operating in parallel on two various levels: packet and flow levels for detecting the Locky ransomware. In another research work presented by [19] where a dynamic malware detection framework using Deep Neural Network (DNN) and Convolutional Neural Network (CNN) was proposed for malware detection. The evaluation report, a combination of DNN and LSTM provide effective in detecting new malware and achieved 91.63% accuracy. An artificially full-automated intrusion detection system for Fog security against cyberattacks was proposed by [20]. They use multi-layered recurrent neural networks applied to the NSL-KDD dataset for detecting four types of attacks: DDoS, Probe, U2R and R2L. In addition, they do not implement blockchain in their solution as an integrated mechanism for monitoring and securing industrial IoT networks. Furthermore, another group of authors developed an algorithm for detecting denial-of-service (DoS) attacks using a deep-learning algorithm. They use the same dataset employed by us, but they just aim to detect one attack (DoS) and do not integrate blockchain in their solution [23]. In that work, three machine learning techniques were used namely Random Forest, SVM and Logistic Regression for the experiments, giving 5.51%, 93.43% and 90.28% accuracy rate respectively [24]. A scheme combining a genetic algorithm and fuzzy logic for network anomaly detection was proposed. The genetic algorithm is used to generate a digital signature of a network segment, and the fuzzy logic scheme decides whether an instance represents an anomaly [28]. The scholars [25] proposed a machine learning security framework for IoT systems. They built a dataset based on the NSL-KDD dataset and evaluated their proposal in a real smart building scenario. As we said in the previous related works, an old dataset may not be suitable for modern IoT networks. They use one-class SVM (Support Vector Machine) technique for detecting four types of attacks: DDoS, Probe, U2R and R2L. [26] presented an intelligent detection system based on deep learning and One Class Support Vector Machine (OCSVM) for revealing the ransomware attacks. The LSTM and Convolutional Neural Network (CNN) were used in the first stage for classifying the collected API calls and then converting them to numerical values to detect if this activity is ransomware. In the efforts of [14], a hybrid approach of genetic algorithm and the fuzzy system was implemented. Therefore, the genetic algorithm presented as preprocess step of the proposed system. The testbed environment was implemented using the KDD-99 dataset. The proposed system recorded 0.94% as the average detection rate. Yet, a comparison of classification techniques applied for network intrusion detection and classification using single classifiers algorithms, Breadth-Forest Tree (BFTree), Naïve Bayes Decision Tree (NBTree), J48, Random Forest Tree (RFT), Multi-Layer Perceptron (MLP) and Naïve Bayes with a reduction in false positive and there is need to evaluate the model on the most updated datasets [7].

1.3. Research Gap in Real-time Threat Performance Metrics

A real-time hybrid intrusion detection approach was proposed by [32] in which misuse approach was used to detect well known attacks while anomaly approach to detect novel attacks. In this work a high detection rate was achieved because patterns of intrusions that could escape the misuse detection could be identified as attack by the anomaly detection technique. The model's accuracy increased incrementally each day up to a significant value of 92.65% on the last day of the experiment, also, as the model learns and trains the system each day, the rate of false negative decreases sharply. In another research work with MitM. [34] proposed real-time intrusion tolerance system, which is based on anomaly-based intrusion detection, is presented. The effectiveness of the approach was tested in a simulated environment, and various attacks could be detected. Also, a machine learning based approach to anomaly detection, using data from a live running industrial process control network, is presented in the same research. System has become a crucial part of computer security, which is used in detecting the above-mentioned threat [31]. The scholars [27] proposed a hybrid model based on improved fuzzy and data mining techniques, which can detect both misuse and anomaly attacks. The author proposed a scheme combining a genetic algorithm and fuzzy logic for network anomaly detection. With real network traffic, the proposed approach achieves an accuracy of 96.53% and a false positive rate of 0.56% by [27]. Also, the authors, [34], used deep neural network (DNN) to reveal a ransomware attack. Their proposed model was built based on features that were extracted from HTTP packet payload inspection. The results proved the efficiency of DNN in detecting ransomware

with accuracy of 93.9%. The plan of the suggested system is to apply the proposed intelligent IDS to an IoT-based network with dynamic network topology. Detecting and neutralizing an adversary inside your environment before they can compromise your backups or encrypting your data will considerably improve your outcomes [33].

2. Materials and Methods

In this study, the mixed research method, that is both qualitative and quantities data was used towards analyzing the given statement problems and shall focus on integrating the complement futures of both methods on conducting that research procedures that involve focus group, interview, secondary dataset. This is to ensure the correctness of the outcome by increasing the reliability and validity of the results of the research work. The data collection for the system analysis and design was based on focus groups in the industry, interviews and secondary dataset deployed after studying suitable data repository documents from Kaggle, established procedures, guidelines etc. the qualitative industrial IoT datasets and the non-numerical quantitative values from Kaggle IIoTset dataset were obtained for system design. The Comma Separated Value (CSV) File secondary dataset obtained from Kaggle.com, the list of attacks scenarios included in the Edge IIoTset and the normal traffic Kaggle dataset for industrial IoT attacks, also, the description of extracted dataset features from all the attacks and the normal traffic that contains network traffics of DDoS, MitM and Ransomware profile attacks will be used for the designing of the machine learning (ML) hybrid approach system in this study. CSV are the most common of the file format available on Kaggle and are the best choice for tabular data (Kaggle.com). In other words, Kaggle is a network activity dataset, which was extracted from the activities performed in an active network with attacks and anomalies. Such attacks/anomalies are the 229, 023 of DDoS attacks on an active Hyper Text Transfer Protocol (HTTP) flood attack, 1,230 of MitM attacks also done on an active network where the intruder mimics the real network protocols or activities. Then, finally 10, 926 Ransomware attacks on an active network using the loopholes in the system to gain access [33]. On the system dashboard, during the model training, 70% of the dataset network activity collected were used for training, 30% for the testing process. In other to support the secondary data obtained from Kaggle, a quantitative data tools like face-to-face interviews and focus groups with plant managers, process and control room panel operators in the industrial IoT domain like petrochemicals, refineries, oil and gas, and fertilizer plants were contacted during this research work within Nigeria.

2.1. The System Experimental Set-up

The experimental set-up depends on the system requirement as an outlined document that structured out the details on how the system will operate or serve the end-users. It is made of specified types, hardware and software requirements and others. It is considered stepwise:

2.1.1. Hardware Requirement

The recommended hardware for the experimental set up includes, Processor: recommended at least 600-MHZ Pentium III-class processor, Hard disk at least 10GB of available space required on system drive, 3.3 GB of available space required on installation drive, Display super VGA (1024 x 768) or higher resolution display with 256 colors, RAM: Minimum requirement of 4GB for best performance, CD-ROM Drive, USB port enabled and USB Fingerprint machine.

2.1.2. Software Requirement

The software considered for the experimental set up are as follows, Operating System: Microsoft Windows 10, Xampp Local Host Server Application Software Program, Python Tools and Sublime Code Editor and Internet Service Provider (ISP), Edge.

2.1.3. Experimental Procedure

The experimental procedures comprise of Software Set-Up, Python Programming Language 3.9, Python Frame/Libraries, Tensor, Python Flask, Seckit-learn, Database details-SQLite, Datasets Used (Ransomware_attack.csv, DDoS_HTTP_Flood_attack.csv, MITM_attack.csv, Normal.csv.

2.1.4. Balancing the Classes

The process of balancing the classes was done by down sampling the majority class (DDoS) to match the smallest class size and unsampled the minority classes (MITM, Ransomware, Normal) to match the target size. The preprocessing steps are removing irrelevant or redundant columns, filling NaN values in numeric columns (mqtt.topic, mqtt.protoname) with:

1) Mapped boolean values: False → 0, True → 1.
2) Converted hexadecimal features (e.g., tcp.flags, tcp.checksum) to decimal integers.
3) One-hot encoded categorical features such as HTTP methods, versions, and DNS query lengths.
4) Feature scaling using Min-Max normalization.
5) Labels encoded as integers: DDoS_HTTP=0, Ransomware=1, MITM=2, Normal=3.

The dataset was split into training (70%) and test (30%). The model training process used by the model Architecture is the 1D ResNet50 followed by Dense classification head. Input (92, 1) shaped features, the output: 4 classes (softmax activation), optimizer was the Adam with learning rate 0.001, Loss function that categorized cross entropy and metrics monitored are Accuracy, Recall, Precision. The training procedure was performed on preprocessed and scaled features., Labels one-hot encoded, Input reshaped to 3D for Conv1D: (samples,

features, 1), model compiled using Keras and batch size and number of epochs are not explicitly defined; default settings used if model. Fit is called externally.

2.2. Model Evaluation

The machine performance metrics calculated the Accuracy, Precision, Recall, F1-score, Confusion Matrix, False Positive Rate (FPR) and False Negative Rate (FNR), ROC-AUC score (macro average for multi-class) and Classification report for each class. The model evaluation also provides the visualization and confusion matrix heatmap with that of accuracy, precision, recall. F1 Score AUC ROC curve as shown in figure.

2.3. Model Simulation

The model simulation was done with Random sample selected from test set, Zeros replaced with half of the maximum feature value (to simulate realistic anomaly). Predicted attack type recorded along with affected hardware simulation (SCADA, PLC, DCS, HMI) and detection Rate and system performance analyzed graphically. The Experiments were conducted using Kaggle cybersecurity datasets with traffic samples of Ransomware, MitM, DDoS, and normal activity.

3. CNN-Fuzzy Hybrid Architecture

The proposed system architecture of the real-time intrusion detection and prevention in industrial IoT systems comprises of four (4) levels of phases. The entry level, inference level, industrial level and action level. At the entry level, comprising the quarantine database, network monitor and incoming network traffic. The Inference level is situated immediately below the entry level, and it comprises two-machine learning (ML) features, it is made of the CNN inference engine and fuzzy logic inference engine. While CNN inference engine uses several types of classifiers, fuzzy logic inference engine uses membership functions. At the Industrial level, several industrial hardware or software (industrial instrumentation on automation with different operational principles connected on network) been protected by the system. At the action level, domain experts or admins are willing to use their skills to perform incident response and assist the operations of the system. Figure 1 reflects the proposed ML model architecture.

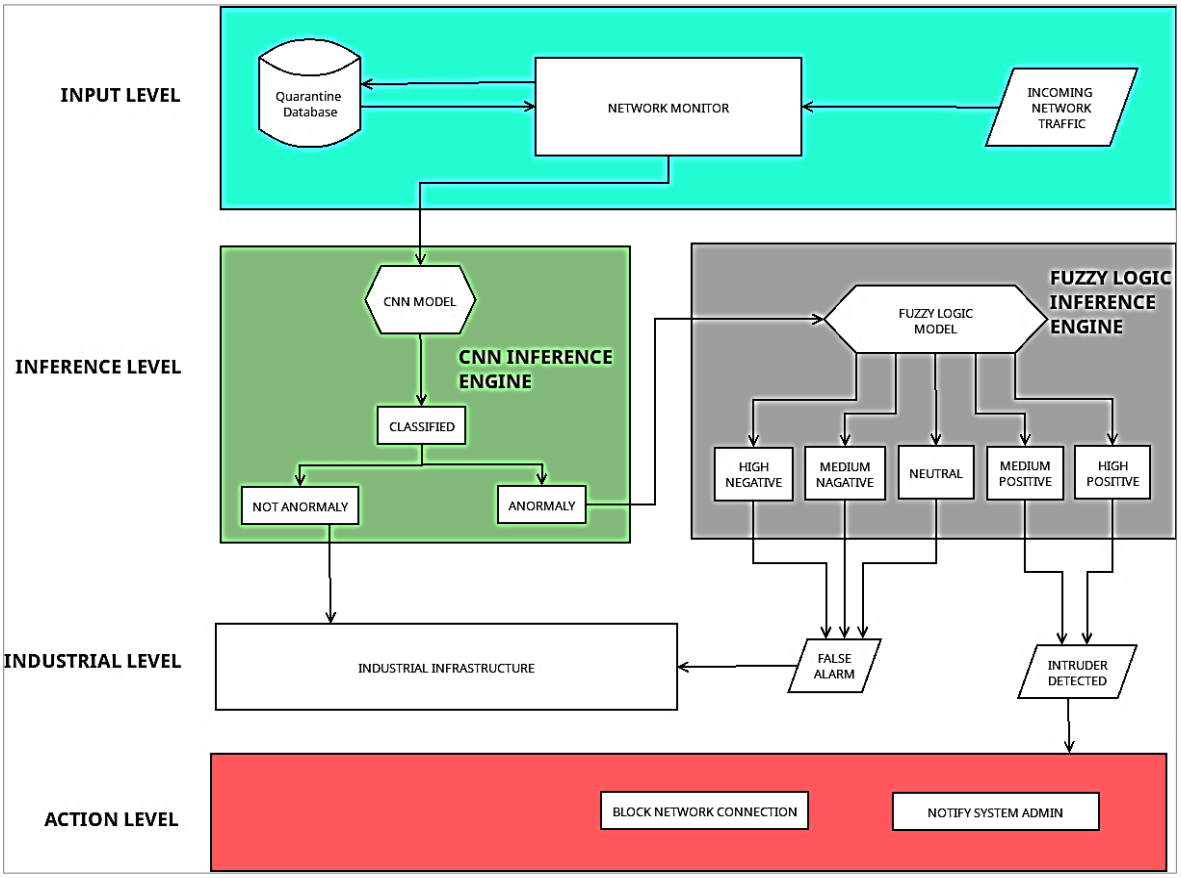

Figure 1. *The Proposed Model Architecture.*

3.1. CNN Inference Engine

Responsible for hierarchical feature extraction using convolutional and pooling layers, followed by dense layers for classification. The Convolutional Layers (CL) then apply convolutional filters (kernels) to the input and incoming data, performs element-wise multiplication, capturing local patterns and features like edges and textures. The non-linear activation function like ReLU (Rectified Linear Unit) during the CNN extraction process is applied after each convolution to introduce non-linearity into the model, allowing it to learn more complex patterns. Batch Normalization during the extraction process may be applied to stabilize and speed up training by normalizing the output of the CL. On the Pooling Layer (PL) which the feature extraction process occurred, the input from the CL is reduced to spatial dimensions of the feature maps, preserving essential information while decreasing computational load. The final layer of the feature process usually contains neurons corresponding to the output classes, using a softmax activation function for multi-class classification tasks indicating the presence of certain features or anomalies. After feature extraction, the processed data is passed through fully connected layers that classify the input into predefined categories (e.g., detecting objects, identifying action). Figure 2 reflects the CNN inference engine diagram.

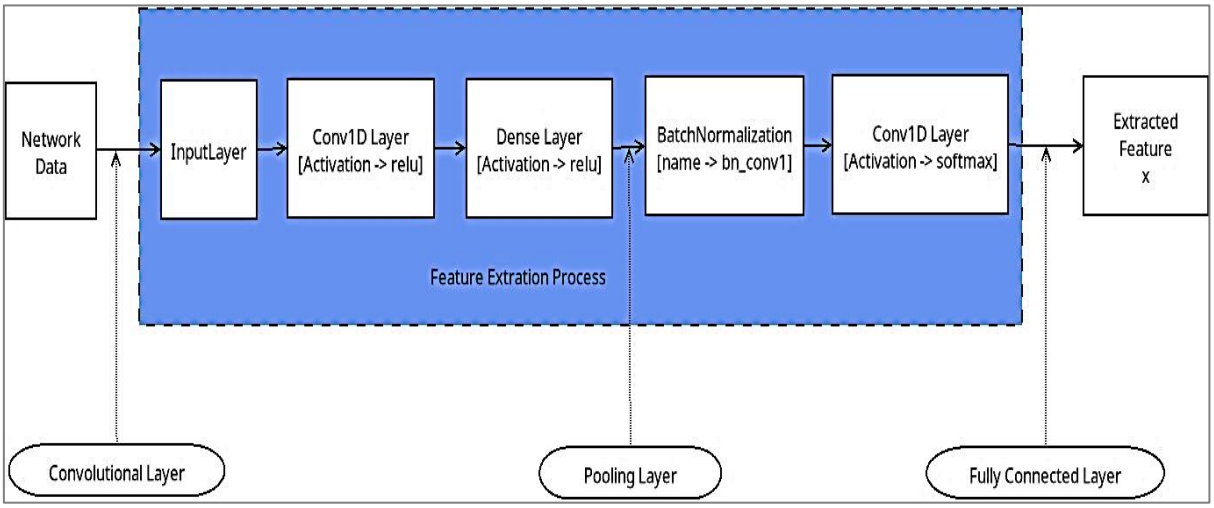

Figure 2. *CNN Inference Engine.*

In the proposed hybrid real-time IDS, the CNN operates as the feature extractor and preliminary classifier, while the fuzzy inference system acts as a decision refinement layer. CNN outputs class-confidence probabilities that are fuzzified into linguistic variables (Low, Medium, High) and combined with statistical anomaly indicators in a fuzzy rule base. The fuzzy inference engine applies Mamdani aggregation and defuzzification to generate a final, crisp intrusion decision. This cascaded CNN→Fuzzy architecture was selected over a parallel fusion scheme to minimize computational overhead which is recommended for further research studies and latency while improving detection robustness in uncertain or overlapping traffic conditions.

3.2. Fuzzy Logic Inference Engine

Takes CNN outputs and applies linguistic rules (e.g., If intrusion probability is high and uncertainty is medium, classify as intrusion). The model uses the fuzzy logic inference block to analyze data inputs and make decisions based on various parameters, also for the prevention and mitigating against anomalies, FL technique also helps in making proactive decisions to prevent incidents, which are mainly the anomalies from the CNN inference engine predicted (x) extracted features. So, the FL inference engine block does the prediction of the five states using a membership function to classify the suspected anomalies network into a High Negative anomaly, (x== 0), Medium Negative anomaly, (x>0 && x<=0.5) a Neutral anomaly, (x== 0.5), Medium Positive anomaly, (x>0.5) and a High Positive anomaly.(x=1). If the signature is found to be between a High Negative and Neutral anomaly, then it is a false alarm (x==0 and x==0.5), the network is then permitted to go through to the respective industrial infrastructure. Finally, if the ML is kept on auto, the anomaly is blocked or quarantined by the system, but if there is false alarm, the operator admin acts (the concern dept). Here it will involve the engineering team concern to acknowledge and reset the alarm on the industrial infrastructure (IIoT) systems concern. The FL inference engine block is shown in Figure 3 below.

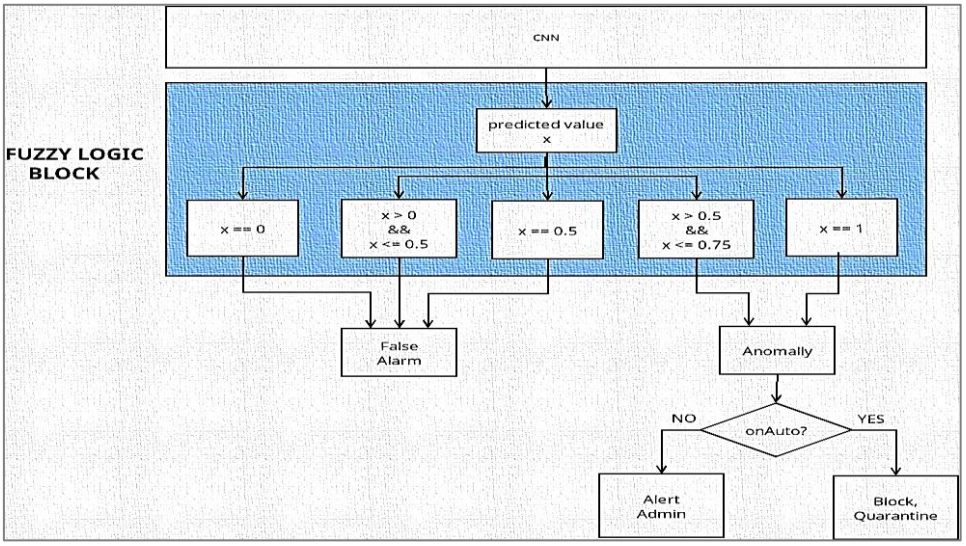

Figure 3. *Fuzzy Logic Inference Engine.*

The fuzzy inference rules were initially formulated using expert knowledge of industrial IoT network traffic behavior, focusing on CNN output confidence and anomaly levels. Membership functions were manually defined with overlapping triangular and trapezoidal shapes for interpretability. Subsequently, the parameters of these functions were optimized through data-driven tuning using the training dataset to minimize false positives, false negatives, and detection latency. This hybrid expert–data-driven rule derivation strategy ensured both transparency and adaptability of the decision-making layer, enabling robust real-time intrusion detection in dynamic IIoT environments.

4. Results and Discussion

In this section we explore decisive and effective machine leaning evaluation metrics to know performance of the model, meaning by evaluation, each value of accuracy, precision, recall, F1-score and AUC ROC were calculated to provide a comprehensive evaluation of its performance. The metrics asses' effectiveness of the model in correctly classifying the dataset sample and the general system prediction time (PT) that revolves around several factors will be determine. The model evaluation results are displayed on the main industrial IoT system panel when a user (operator, engineers etc.) clicked on the model button (MB) on the dashboard. The five (5) buttons, Create Model (CM), Evaluate Model (EM), View Plot (VP), View Confusion Matrix Plot and Clear Results (CR) appear on the dashboard and use them appropriately for the evaluation. The Evaluation Report and Classification Results, and the Graphical Representation of performance Evaluation Metrics are displayed on the industrial IoT system on a user clicked at both the view plot (VP) and confusion matrix buttons. Figure 4 below shows the performance evaluation of ML metrics.

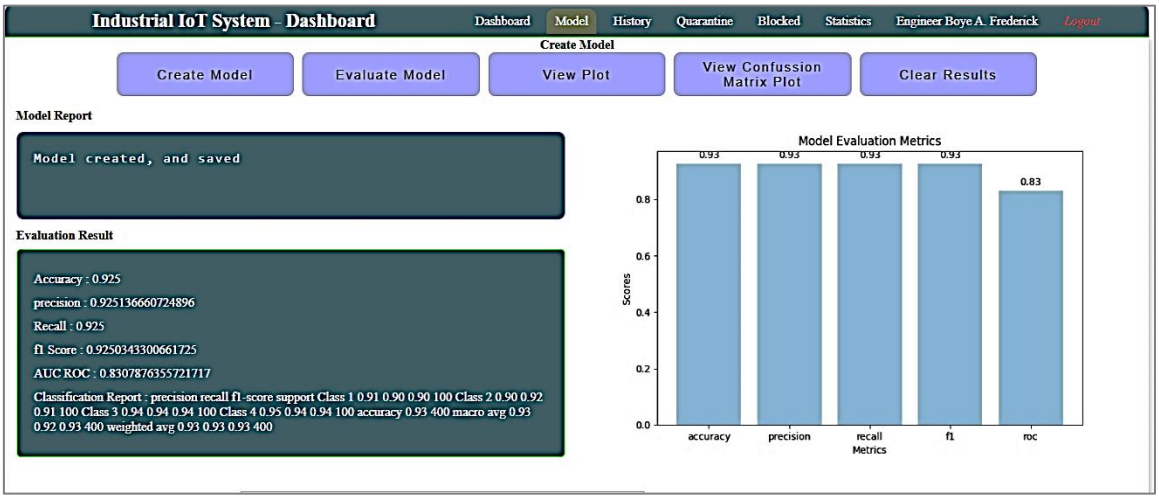

Figure 4. *Performance Evaluation of ML metrics.*

4.1. Industrial IoT System User Interface (UI)

The industrial IoT System comprises of the following list of menus, buttons etc. the System User Interface comprises of the Dashboard, Model, History, Quarantine, Blocked, Statistics, Username, Logout button. Also, the Network Action Buttons (Start, Stop, Quarantine, Block, Delete), An Incoming Network Signal Plotter, Allowed Network Permitter List and the Attack Statistics. The main system dashboard is accessed after user (operator, engineer etc.) login with user correct credentials. The system dashboard is shown in Figure 5 below. It comprises of six (6) buttons, Model, Dashboard, History, Quarantine, Block, Username, Logout. Immediately after the dashboard for user to seamlessly operate industrial IoT system. On clicking the Model Menu, a panel will pop-up showing the following buttons, Create Model, Evaluate Model, View Plot and Clear Results. Each of the button's functions as design. On clicking the Quarantine Menu display another panel. The quarantined menu is on the right-side panel. When clicked the quarantined IPs and the attack type (DDoS, Ransomware or MitM) pop-up showing the selected network signature plot, time, date and month of the quarantined attack type. On the other hand, the Block Menu displays same with the quarantine menu. The History Menu on the dashboard when clicked shows the IPs of overall network statistics where their details in percentage and pie chart represented. This shows the blocked, quarantine and normal IPs networks. When a user clicks on any of IP's listing on the left side of the panel, allows a selected IP network signature plot is display showing a graphical view of IPs. This main UI dashboard is shown in Figure 5 below.

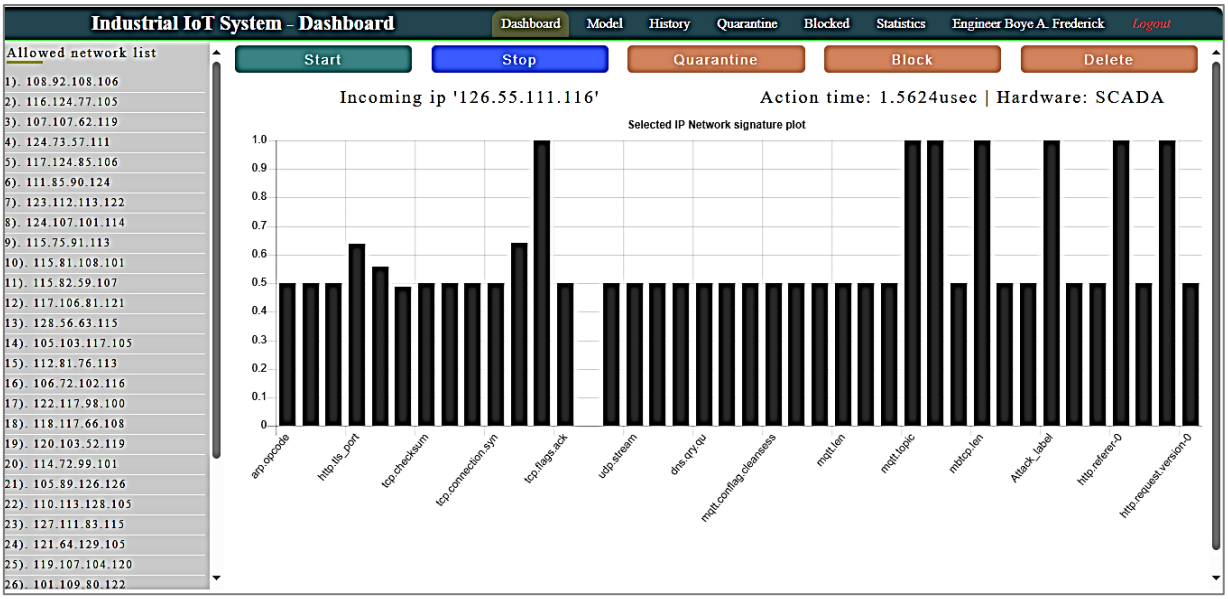

Figure 5. *Main System User Interface.*

The results from Figure 4 show the analyzing performance evaluation on Accuracy, Precision and Recall which achieved 92.5% F1-score 92.5%, AUR ROC 84.5%, False Positive Rate (FPR) average of 2.51% whereas the detection rate (DR), computational cost (latency) results were based on Table 3 with 14 domain experts users. The model classification report Table 3 is achieved using the confusion matrix Table 1 below with the parameters for precision, recall, F1 Score and support.

Table 1. *Model Classification Report.*

Class	Precision	Recall	F1-Score	Support
Class 1	0.91	0.90	0.90	100
Class 2	0.90	0.92	0.91	100
Class 3	0.94	0.94	0.94	100
Class 4	0.95	0.94	0.94	100
Accuracy			0.93	400
Macro Avg	0.93	0.92	0.93	400
Micro Avg	0.93	0.93	0.93	400

The false positives can trigger unnecessary security actions, causing production delays or interruptions in industrial processes. For instance, if an IDPS falsely flags normal PLC

communication as an attack, it may block or isolate a critical system in an industrial plant. A low False Positive Rate (FPR) of ~2.50% is a good indicator, but its acceptability depends on the specific use case and security requirements of the industrial system. Hence, in industrial environments or ecosystem, false positives can cause significant disruptions. Industrial IoT systems must maintain real-time operations. False positives that trigger automated blocking or responses can impact safety-critical systems. Figure 6 shows the performance evaluation metrics on FP (red colour), TN (green colour) and FPR mean or average for class 1 to 4.

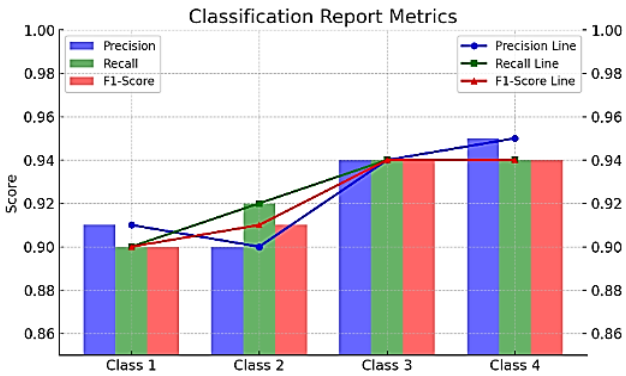

Figure 6. *Model Classification Metrics Report.*

4.2. False Positive Rate System Performance Metric

The study evaluates the FPR of the system performance during the implementation phase as shown in Table 3 from the fourteen (14) industry users from the industry that deployed industrial IoT systems for their operations. False positive rate (FPR) measures how often normal (benign) activity is incorrectly classified as malicious. The system achieved the FPR as shown from Table 3 and can be computed from confusion matrix obtained during the simulation:

1. *False Positives (FP)* for each class from confusion matrix:
   a) *Class 1:* $FP_1 = 4 + 3 + 2 = 9$
   b) *Class 2:* $FP_2 = 5 + 2 + 3 = 10$
   c) *Class 3:* $FP_3 = 3 + 2 + 1 = 6$
   d) *Class 4:* $FP_4 = 2 + 2 + 1 = 5$
2. *True Negatives (TN)* for each class from confusion matrix:
   a) *Class 1:* $TN_1 = 400 - (90+9+10) = 291400 - (90 + 9 + 10) = 291400 - (90 + 9 + 10) = 291$
   b) *Class 2:* $TN_2 = 400 - (92+10+8) = 290400 - (92 + 10 + 8) = 290400 - (92 + 10 + 8) = 290$
   c) *Class 3:* $TN_3 = 400 - (94+6+6) = 294400 - (94 + 6 + 6) = 294400 - (94 + 6 + 6) = 294$
   d) *Class 4:* $TN_4 = 400 - (94+5+6) = 295400 - (94 + 5 + 6) = 295400 - (94 + 5 + 6) = 295$
3. *False Positive Rate (FPR) Calculation:* using the formula-FP/FP + FN
   a) *Class 1:* $FP_1/ FP_1 + TN_1 = 99/99+291= 0.03$
   b) *Class 2:* $FR_2/FR_2 + FN_2 = 10/10+290 = 0.0333$
   c) *Class 3:* $FR_3/FR_3 + FN_3 = 6/6+294 = 0.02$
   d) *Class 4:* $FR_4/FR_4 + FN_4 = 5/5+295 = 0.0167$

The average FPR from the confusion matrix: FPR1 + FPR2 + FPR3 + FPR4/4 = 0.03 + 0.0334 + 0.0201 + 0.0167/4 = 0.0251 Therefore, the average FPR is 2.51%, this FPR value corresponds to the value obtained from Table 3 during the industrial IoT system simulation by the fourteen (14) users from different industries using industrial IoT systems.

This FPR evaluated results means only 2.51 out of every 100 normal events are mistakenly flagged as threats. In industrial settings where system stability and uptime are critical, this is a low and acceptable rate, indicating high model precision. Hence, 2.51% of FPR lies in the excellent performance range for industrial IoT cybersecurity applications. The above FPR was obtained from the confusion matrix table during the system evaluation as shown below.

Table 2. *True Negative Calculation.*

Class	TP (Diagonal)	FP (Column Sum-TP)	FN (Row Sum-TP)	TN (Cal)	TN
Class 1	90	(4+3+2) = 9	(5+3+2) =10	400-(90+9+10) = 291	291
Class 2	92	(5+2+3) =10	(4+2+2) = 8	400-(92+10+8) = 290	290
Class 3	94	(3+2+1) = 6	(3+2+1) = 6	400 - (94+6+6) = 294	294
Class 4	94	(2+2+1) = 5	(2+3+1) = 6	400-(94+5+6) = 292	292

4.3. Industrial IoT System-attack Statistics

This attack statistics window on the industrial IoT system main dashboard gives the result of the systems attacks with date options. When a registered domain user clicked on this required button, attack type and number of attacks are displayed as results for the admin (user) viewing. Figure 7 is an example of a display of attacks statistics results for number of intrusions to the system. It was successfully system simulation from one of the 14 system users showing the Attack type-DDoS, MitM, network access and ransomware, number of attacks: 70 DDoS_HTTP, 56 MitM, and 63 Ransomware attacks.

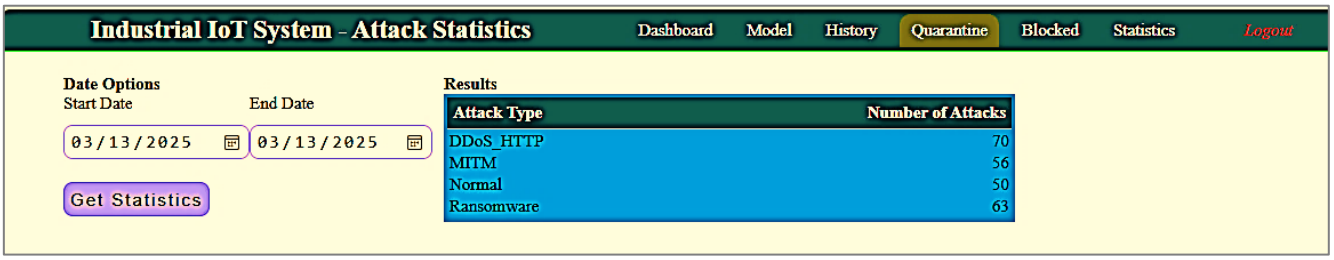

Figure 7. *Industrial IoT System Attack Statistics.*

4.4. Industrial IoT System Simulation (IISS)

The results of the performance evaluation metrics (PEM) of the 14 users (Table 3) during the implementation analysis are taken from various metrics, and each user performs the simulation using the industrial IoT system with a time limit between 30 minutes to 1 hour. The table captures all the metrics as per industrial IoT system design specifications.

Table 3. *System Threat Detection Performance Metrics Results.*

Date	User(s)	Total Attacks	DDoS Attack	MitM Attack	Ransomware Attack	Normal Network	Blocked Attacked (%)	Quarantined Attacks (%)	Normal Network (%)	IPs	Latency (µsec)	FPR	Accuracy	ROC
010225	Boye	187	49	76	62	55	0.773	0.000	0.227	242	1.5624	2.50	92.5	8.30
080225	Taylor	146	61	47	67	60	0.745	0.008	0.255	196	1.5620	2.48	92.5	6.00
210225	Clement	158	55	58	52	58	0.731	0.000	0.269	216	0.8986	2.51	93.0	5.90
220225	Daniel	187	58	64	65	69	0.730	0.000	0270	256	0.3994	2.50	93.0	7.20
230225	Momotimi	198	57	62	79	57	0.788	0.000	0.212	269	1.5821	2.51	92.5	6.00
240225	Grace	191	56	67	66	58	0.767	0.000	0.233	249	1.5456	2.50	92.0	7.00
250225	Israel	181	60	65	56	54	0.770	0.000	0.230	235	1.2543	2.54	92.5	8.05
260225	Godswill	145	46	41	58	59	0.711	0.000	0.289	204	1.5804	2.50	92.5	7.81
270225	Rose	177	60	65	52	62	0.738	0.000	0.263	231	1.1465	2.50	92.5	6.50
280225	Miracle	149	52	50	47	52	0.741	0.000	0.259	201	0.4249	2.52	92.5	6.14
010325	Isaac	220	65	79	76	67	0.767	0.000	0.233	287	1.5802	2.50	92.5	5.60
070325	Patel	140	46	45	49	58	0.797	0.000	0.293	198	1.5626	2.49	92.5	7.33
110325	Saheed	253	78	94	81	75	0.771	0.000	0.229	328	0.8004	2.51	92.5	7.34
130325	Kelvin	189	70	56	63	50	0.791	0.000	0.209	239	1.4984	2.50	92.5	7.00
Average		179	57	63	62	59.57	0.760	0.000	0.248	240	1.207	2.51	92.54	6.90

4.4.1. System Latency Performance Analysis (SLPA)

The study evaluates the delay between user input or request and the corresponding output from the system or response within the industrial IoT system. A system low latency indicates a faster responsive system, while a high latency results in delays by the system's behaviour.

Computing the latency values in percentage (%) and μseconds using the above Table 3, we have the following. To calculate the average of the latency in microseconds (μsec) and then express that average as a percentage, we need to define: We also compute the Latency (response time) Percentage (%) per Date using the formula:

= Latency % = (Latencyi/16.8978)×100, this will give us the following Table 4 as shown below.

Table 4. *System Latency (%) Per Date Performed by System Users.*

Date	Latency (μseconds)	Latency (%)
010225	1.5624	9.24%
080225	1.5620	9.24%
210225	0.8986	5.32%
220225	0.3994	2.36%
230225	1.5821	9.36%
240225	1.5456	9.15%
250225	1.2543	7.42%
260225	1.5804	9.35%
270225	1.1465	6.78%
280225	0.4249	2.51%
010325	1.5802	9.35%
070325	1.5626	9.24%
110325	0.8004	4.74%
130325	1.4984	8.87%

4.4.2. System Total Latency (μsec)

Latency (response time) is crucial in networking because it is the time it takes data packet to move from its source to destination. One of the objectives achieved in the study is the system latency which corresponds to the average in percentage of 7.14% computed as follows.

Total Latency = 16.8978 μsec.

Average Latency (Response Time) = 16.8979/14 = 1.207 μseconds.

Converting the total latency to percentage (%) we have Average Latency (%) = [1.207/16.8979] × 100. Average Latency (%) = 7.14%.

Average Latency = 7.14%. In industrial IoT systems, latency requirements are highly application-specific, but 7.14% of total latency (which corresponds to 1.207 μsec average latency) is very good and acceptable for most industrial IoT applications.

Table 5. *Latency in industrial IoT Applications.*

IIoT Application	Typical Latency Requirements
Factory Automation	1-10 milliseconds (ms)
Motion Control (Real-Time)	≤ 1ms
Remote monitoring & SCADA	10-100ms
General industrial communication	≤ 50ms
Industrial 5G URLLC	≤ 1ms (ultra-reliable low latency)

The performance of the industrial IoT system shows that the latency average in milliseconds = 1.207 μsec = 0.001207ms which is far below the tightest requirement (1ms)

for ultra-critical real-time control. This refers to very time-sensitive operations where even small delays can lead to failure, damage, or safety issues. These are common in Industrial IoT applications like: Robotics (e.g., robotic arms in manufacturing), Autonomous vehicles (e.g., AGVs in warehouses), Real-time safety systems (e.g., emergency shutdowns) and High-speed conveyor systems. This means that in these systems, data must be transmitted and acted on within 1 millisecond (ms) to avoid disruptions or hazards.

4.5. Intrusion Detection System Performance Analysis

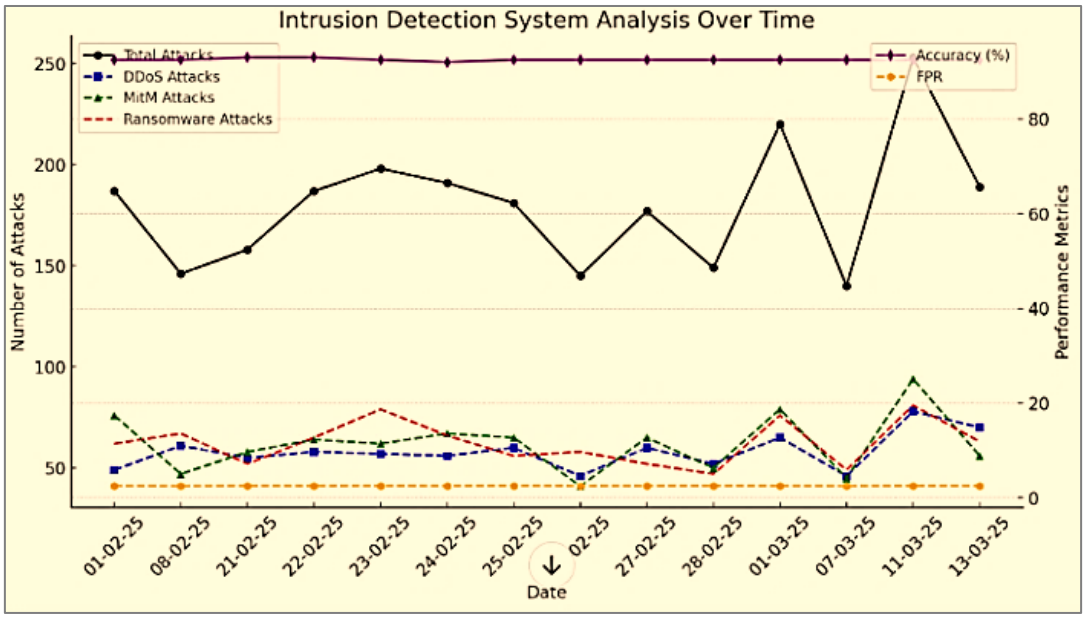

Figure 8. *IDS Performance Graphical Analysis.*

Intrusion Detection System performance analysis" provides insights into the attack trends of an intrusion detection system (IDS) from 01-02-25 to 13-03-25. It includes multiple metrics plotted time, total attacks (Black Line), the total number of attacks fluctuates significantly, with values ranging approximately between 140 and 220 attacks per day. There is a noticeable dip around 08-02-25 and 07-03-25, followed by peaks around 23-02-25 and 11-03-25. Attack types (Stacked/Individual Categories): DDoS attacks (Blue Dotted Line): Steady but variable, generally between 40 and 70 attacks per day, with peaks around 11-03-25. MitM attacks (Green Dashed Line): Highest at the start (~75 on 01-02-25), dips after 08-02-25, and rises again with a significant peak around 11-03-25. Ransomware Attacks (Red Dashed line). Consistent between 40 and 75, peaking near 23-02-25 and 01-03-25. Figure 8 shows the IDS analysis during the industrial IoT system implementation analysis.

4.5.1. System Detection Rate (SDR)

Industrial IoT systems are vulnerable systems are open to cyber threats and attacks therefore integrating a system to monitor, detect any anomaly becomes imperative. Meaning the percentage of the actual threats or anomalies, the design model successfully identifies out of the total number of threats or active anomalies. The system evaluates the detection rate (DR) of the attack profiles (DDoS, Ransomware and MitM) during the implementation phase achieving the following results.

Computing for the Detection Rate (DR) using Table 3.

Recalling the Formula for Detection Rate (DR). In intrusion detection, the Detection Rate (DR) is usually defined as: DR=TP/TP+FN ×100

Where:

TP (True Positives) = Correctly detected attacks (Blocked + Quarantined).

FN (False Negatives) = Attacks that were not detected (slipped through as "Normal").

Extract Values from Table 3:

Total Attacks = 179

Normal Network (%) = 0.248 ≈ 24.8% (represents undetected traffic or FN share).

Blocked Attacks (%) = 0.760 ≈ 76.0% (represents TP share).

Quarantined (%) = 0.000

So, in terms of proportions of traffic:

TP = 0.760

FN = 0.248

Computing the Detection Rate

Detection Rate (DR) =TP/TP + FN ×100

Substituting, We the Detection Rate (DR) = 0.760/0.760+0.248 ×100

Detection Rate (DR) = 0.760/1.008 ×100 = 92.9%. The 92.9% detection rate (DR) was obtained by comparing blocked/quarantined attacks vs. the total of blocked + missed (normal) attacks. The 93% approximately detection rate in integrating CNN & Fuzzy Logic was computed using the implementation system threat detection performance capabilities Table 3. In practice, many published industrial IoT intrusion detection models report 90–95% detection rate (DR) with carefully tuned machine learning or hybrid approaches.

4.5.2. Computational Overhead

The system has a computational overhead (CPU/Memory Usage), and this is with the hardware/software list provided as a post graduate student during the experimental setup which gave several compatibility and performance concerns for modern real-time intrusion detection, model inference, or development work. The listed hardware/software (especially a 600-MHz Pentium III and 4 GB RAM) is too appropriate for reliable, real-time intrusion detection & prevention in modern industrial IoT deployments. This will experience computational overheads that risk missed detections, high latency, or excessive false alarms due to resource contention. Minimum practical setup for real-time industrial IoT ID should come with modern multi-core CPU, 8–16 GB RAM, SSD, stable OS (Windows 10 only if hardware is modern).

4.6. Performance Evaluation Metrics Analysis

The threat detection performance analysis and capabilities on latency were achieved using parameters on Table 3 as the key system average (A) measured values indicators with an Accuracy of 92.54, ROC average of 6.90, FPR average of 0.025 (2.51%), and latency of 1.207 µsec corresponding with an average of 7.14% with detection rate of 92.9% which are the main metrics reflecting detection performance, response time of the system. Figure 9 helps us visualize system performance capability during the model implementation analysis. It provides a clear visual summary of how the hybrid machine learning approach performs in terms of detection efficiency, latency, reliability, and robustness with metrics. The model metric performance accuracy (A) achieving more than 90%, which indicates that the system correctly identifies both normal and malicious activities with over 93% certainty. This reflects a strong ability to generalize and adapt to the Kaggle dataset's real-world traffic. The average latency, which is measured in 0.001207ms results to 7.14% which is very efficient and reliable for industrial IoT applications. The threat detection rate, 92.9%, is exceptionally high, meaning virtually that almost all actual attacks (DDoS, Ransomware, MitM) were detected. It demonstrates excellent sensitivity and minimal chances of missed threats—crucial in real-time cybersecurity. Finally, the chart gives the False Positive Rate (FPR) as 2.51% during the implementation analysis.

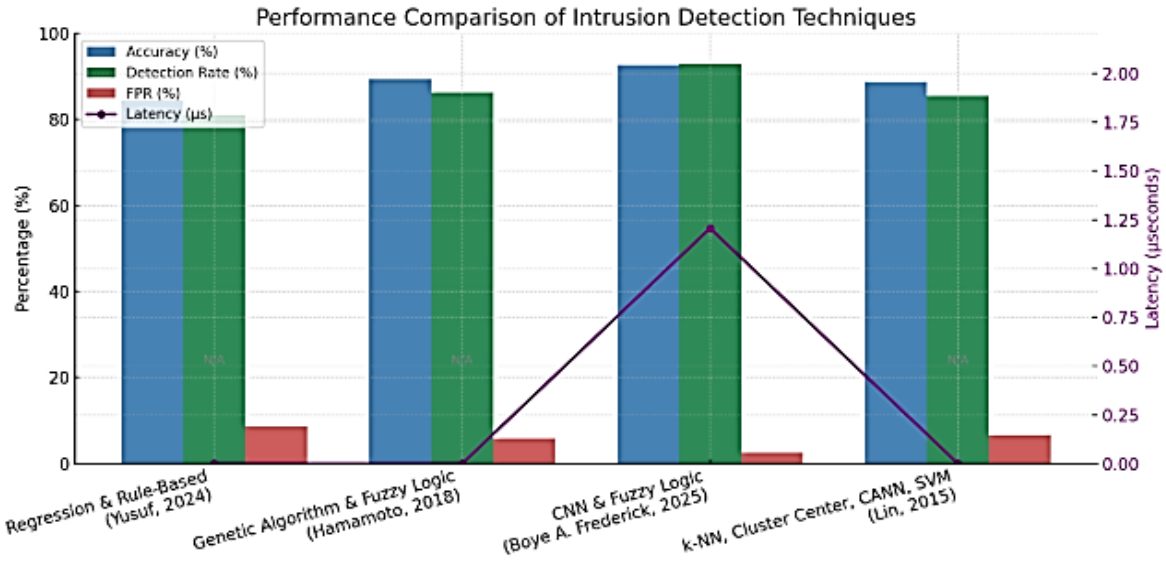

Figure 9. *Threat Comparative Performance Metrics Analysis.*

Figure 9 compared based on accuracy, detection rate (DR), and false positive rate (FPR) using the combined DDoS, Ransomware, and MitM attacks profiles from Kaggle datasets resources deployed for the study. Table 5 represents the results of the four (4) different ML techniques based on their implementation results.

Table 6. *Performance Comparative Techniques Performance.*

Techniques	Accuracy	Detection Rate (DR)	(FPR)
Regression & Rule-Based (Yusuf, 2024)	84.4	80.9	8.75
Genetic Algorithm & Fuzzy Logic (Hamamoto, 2018)	89.3	86.2	5.9
CNN & Fuzzy Logic (Proposed System, 2025)	92.54	92.9	2.51
k-NN, Cluster Centre, CANN, SVM (Lin, 2015)	88.7	85.4	6.7

5. Conclusion

This paper proposed a CNN-Fuzzy Logic hybrid IDS for real-time industrial IoT security. The model achieved high accuracy, strong detection rates, low FPR, and reduced latency, outperforming conventional IDS models. The combination of CNN's deep feature learning with fuzzy logic's adaptability improved robustness and reliability in dynamic industrial IoT environments. The hybrid ML model demonstrated detection rates of more than 90% compared to single classifiers and comparatively the graphical results shown in Figure 8 confirm that the model metrics are reliably met the industrial benchmark with more than 5% accuracy compared to other techniques for industrial IoT applications. The false positives (FP) (2.5%) average rate were significantly reduced. It indicates that the system can catch nearly all attacks while maintaining high accuracy (>92%). Computational performance analysis showed the system latency results show an average percentage less than 1ms which can be deployed in industrial IoT applications. The hybrid model performed well in detecting unknown attack patterns missed by signature-based IDSs and the approach by integration will be reliable and efficient as a layer defense mechanism in an existing industrial IoT network architecture and will enhance multiple intrusion detections. This study provides a detailed performance analysis framework including latency of 1.207 µsec on 7.14% average percentage latency. which can guide future industrial IoT security implementations. Future research will focus on optimizing hybrid ML architectures for low performance metrics for deployment of resource-constrained industrial IoT devices, integrating the hybrid machine learning approach for threat detection, and expanding evaluation to real-world industrial environments. On system computational overhead researcher should employ a minimum practical setup for real-time industrial IoT IDS with modern multi-core CPU, 8–16 GB RAM, SSD, stable OS (Windows 10 only if hardware is modern) or run a lightweight Linux on edge plus offload heavy tasks elsewhere.

Abbreviations

IIoT	Industrial Internet of Things
MitM	Man-in-the-Middle
DDoS	Distributed Denial-of-Service
SCADA	Supervisory Control and Data Acquisition
PLC	Programmable Logic Controller
HMI	Human Machine Interface
DCS	Distributed Control System
PEM	Performance Evaluation Metrics
ML	Machine Learning
FPR	False Positive Rate
IISS	Industrial IoT System Simulation

Acknowledgments

The author acknowledges Kaggle for providing datasets and the research community for their contributions to industrial IoT cybersecurity research. Also acknowledging is the head of instrumentation IFL Port Harcourt for the impactful assistance.

Data Availability Statement

The data that supports the findings of this study can be found at: www.kaggle.com/datasets?fileType=csv. The authors at reference [33] make use of the same industrial IoT dataset. Also, the data is available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Sharma, K., Soumya, B., & Brijesh, K. (2022). Intrusion Detection System in IoT Network using ML, Volume 20, Issue 13, Page 3597-3601| https://doi.org/10.14704/nq.2022.20.13.NQ88441

[2] Ahmed C. M, Gauthama R. M R, Aditya M. (2021). Challenges in Machine Learning based approaches for Real-Time Anomaly Detection in Industrial Control Systems, ResearchGate.

[3] Ramya, M. (2022). What Is a Man-in-the-Middle Attack? Definition, Detection, and Prevention Best Practices for 2022, www.spiceworks.com/it-security/data-security/articles/man-in-the-middle-attack/

[4] H-ISAC, (2021). Distributed Denial of Service (DDoS), Health-ISAC, www.H-isac.org

[5] Chika, A. (2023). Ransomware - An Overview, https://www.nomoreransom.org

[6] Lin, P., Lee, Y., & Huang, C. (2015). A Hybrid Machine Learning Approach for Intrusion Detection Using K-NN, Cluster Center, CANN, and SVM. International Journal of Network Security, 17(6), 674–683.

[7] Aziz, A. S. A. (2016). Comparison of classification techniques applied for network intrusion detection and classification. Journal of Applied Logic 24. Elsevier, 109-118.

[8] Kayvan A, S. Yahya, Amirali R. & Siti Hazyanti Binti M. H (2016). "Anomaly Detection Based on Profile Signature in Network using Machine Learning Techniques." Presented at the 2016 IEEE Region 10 Symposium (TENSYMP), pp. 71–76.

[9] Deyban, P., Miguel, A. A., Perez, A. D. & Eugenio, S. (2017). Intrusion detection in computer networks using hybrid machine learning techniques. XLIII Latin American Computer Conference IEEE, 1-10.

[10] Vinoth, Y. & Kamatchi, K. (2020). Anomaly Based Network Intrusion Detection using Ensemble Machine Learning Technique. International Journal of Research in Engineering, Science and Management, (290-296).

[11] Usman S. M. Megha C., Aniso A. & Mandeep K., (2020). Intrusion Detection System using Machine Learning Techniques: A Review, ResearchGate.

[12] Maniriho et al. (2020). Detecting Intrusions in Computer Network Traffic with Machine Learning Approaches. International Journal of Intelligent Engineering and Systems. INASS. (433-445).

[13] Guardian Nigeria (2022). Technology, Ransomware hits 71% of Nigerian organisations, guardian.ng/technology/ransomware-hits-71-of-nigerian-organisations/

[14] Danane, Y. & Parvat, T. (2015). Intrusion detection system using fuzzy genetic algorithm," in Pervasive Computing (ICPC), 2015 International Conference on. IEEE, 1–5.

[15] Ramadhan, A. M. A., Wael, M. S. Y., Hashem, A., Ghilan, Al-Madhagy, T. H., Abdel-Hamid, M. E. & Ahmed, A. W. (2022). Ransomware Detection using Machine and Deep Learning Approaches. International Journal of Advanced Computer Science and Applications, 13(11).

[16] Alsaidi, R. (2021). Ransomware detection dataset (RDD) dataset. Ransomware detection dataset (RDD) dataset. Available: Kaggle, https://www.kaggle.com/ramdhanamalsaidi/a-novel-datasetcontaining-405836-url Accessed November 28, 20.

[17] Alhawi, M., Baldwin, J. & Dehghantanha, A. (2018). Leveraging machine learning techniques for windows ransomware network traffic detection. Cyber Threat Intelligence, 93- 106.

[18] Almashhadani, A. O., Kaiiali, M., Sakir S., & Philip O. (2019). A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware. IEEE Access, 7, 47053-47067.

[19] Ghanei H., Manavi F. & Hamzeh A. (2021). A novel method for malware detection based on hardware events using deep neural networks. Journal of Computer Virology and Hacking Technology, 17(4), 319–331.

[20] Almiani, M., AbuGhazleh, B., Al-Rahayfeh, A., Atiewi, S. & Razaque, A. (2020). Deep Recurrent Neural Network for IoT Intrusion Detection System. Science Direct Simulation Model for Practical Theory, 101, 102031.

[21] Jiang, K., Wang, W., Wang, A. & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access, 8(32), 464 – 476.

[22] Asad, M., Asim, M., Javed, T., Beg, M. O., Mujtaba, H. & Abbas, S. (2020). Deepdetect: detection of distributed denial of service attacks using deep learning. Ae Computer Journal, 63(7), 983–994.

[23] Susilo, B & Sari, R (2020). Intrusion Detection in IoT Networks Using Deep Learning Algorithm. Information 2020, 11, 279.

[24] Djaballah, K. A, Boukhalfa, K., Ghalem, Z. & Boukerma, O. (2020). A novel approach for the detection and analysis of phishing in social networks: the case of Twitter. In 2020 Seventh International Conference on Social Networks Analysis, Management and Security.

[25] Bagaa, M., Taleb, T., Bernal, J. & Skarmeta, A (2020). A machine learning Security Framework for Iot Systems. IEEE Access, 8, 114066–114077.

[26] Homayoun, S., Dehghantanha, A., Ahmadzadeh, M., Hashemi, S., Khayami, R., Choo, K. K. & Newton, D. E. (2019). DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer. Future Generation Computer Systems, 94-104.

[27] Shanmugam, B. & Idris, N. B. (2019). Improved intrusion detection system using fuzzy logic for detecting anomaly and misuse type of attacks. In Proceeding of 2009 International Conference of Soft Computing and Pattern Recognition, 212-217.

[28] Hamamoto, A. H. Carvalho, L. F. L., Sampaio, D. H., Abrão, T. & Proença, M. L. (2018). Network anomaly detection system using genetic algorithm and fuzzy logic. Expert Systems with Applications, 92, 390-402.

[29] Davies, I., Taylor, O., Anireh, V., & Bennett, E. (2024). Adaptive Hybrid Case-Based Neuro-Fuzzy Model for Intrusion Detection and Prevention for Smart Home Network.

[30] Taylor, O. E., Ezekiel, P. S., & Igiri, C. G. (2021). Anomaly based intrusion detection/prevention system using deep reinforcement learning algorithm. Int. Journal of Adv. Research in Computer and Communication Engineering, 10(1), 58-65.

[31] Boye, A. F., Taylor, E. O. and Bhagat, D., (2024). AI and Performance Capability of Cybersecurity in the Energy Industry. ISAR Journal of Science and Technology, 2(12), 29-36.

[32] Tseng A, Chen Y, Kao Y & Lin T. (2016). Deep learning for ransomware detection. IEICE Tech. Rep. 116(282), 87-92.

[33] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., Member, S. & Janicke, H. (2022). Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning". Institute of Electrical and Electronic Engineering TechRxiv conference.

[34] Boye A. F. and Onate Egerton Taylor (Analysis on Cybersecurity Control and Monitoring Techniques in Industrial IoT: Industrial Control Systems, 2023; 11(1), 1-17 https://doi.org/10.11648/j.iotcc.20231101.11