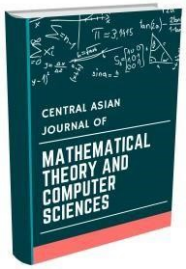

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 07 Issue: 01 | January 2026 ISSN: 2660-5309

Article

Hybrid Blockchain–Machine Learning Framework for Secure and Intelligent Intrusion Detection in Industrial Internet of Things

Zaid Mohammed Mortada[1], Ola Baqer Abdulhadi[2]

1. Department of Postgraduate Studies, University of Kufa, Najaf, Iraq
2. Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq

* Correspondence: zaidm.alhusaini@uokufa.edu.iq

Abstract: As a result of the Industrial Internet of Things (IIoT), the industrial ecosystem has been transformed by the exchange of data in real-time and the automation of operations in an intelligent manner. Cybersecurity risks are associated with this connectivity, especially for resource-constrained and heterogeneous IIoT networks. An intrusion detection framework incorporating blockchain technology and machine learning algorithms is proposed in this study to address these challenges. Data sharing and auditability are ensured through decentralized, tamper-resistant blockchain, while evolving threats can be detected through adaptive machine learning. For multiclass classification, the proposed system takes advantage of XGBoost, and for hyperparameter optimization, it uses the HAFSO algorithm. For multiclass classification, the proposed system takes advantage of XGBoost, and for hyperparameter optimization, it uses the HAFSO algorithm.

Received: 10th Nov 2025
Revised: 11th Dec 2025
Accepted: 24th Dec 2025
Published: January 26, 2026

Keywords: Blockchain, Intrusion Detection System (IDS), Machine Learning, Optimization Algorithm, Industrial Internet of Things (IIoT).

1. Introduction

As a result of real-time data exchange, automation, and intelligent decision-making enabled by the Industrial Internet of Things (IIoT), industrial systems have been revolutionized over the past few years. Due to increased connectivity, IIoT environments have become more vulnerable to sophisticated cyberattacks. In addition to being dynamic and heterogeneous, IIoT networks face tight resource constraints, making it difficult for traditional Intrusion Detection Systems (IDSs) to keep up. To address these challenges, we propose combining blockchain technology with machine learning (ML). In addition to providing secure data sharing and auditability, blockchains provide a tamper-resistant framework, while machine learning algorithms provide intelligence to detect abnormal patterns and evolving attack vectors. The proposed system incorporates blockchains immutable trust model and machine learning's predictive abilities to enhance IDS

accuracy, transparency, and resilience. By utilising a hybrid architecture, not only are single points of failure and data manipulation risks minimised, but also scalable, autonomous threat detection is supported. A smart contract can also be integrated into an automated response mechanism to mitigate intrusions as soon as they are detected. The research aims to establish robust, smart, and trustworthy security infrastructures tailored to the next generation of IIoT applications by using this synergistic framework.

In recent years, the Internet of Things (IoT) has become a research challenge that has penetrated our daily lives [1]. Using the Internet of Things, data can be collected, processed, and analysed faster, more securely, and more reliably. A recent Gartner study on IoT analysts predicts that there will be 25 billion internet-connected things by 2025, up from 20 billion in 2020 [2]. As Industry 4.0 continues to gain traction, actuators, controllers, sensors, and other devices are becoming connected via the Internet of Things (IoT) [3]. Figure 1 illustrates how IIoT technology has evolved from one stage to another.

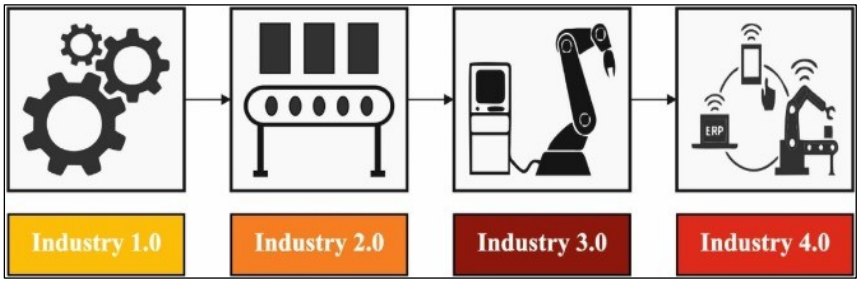

Figure 1: Evolution of IIoT.

Having a high information generation rate, low latency, high reliability, and a large number of sensors, actuators, devices, and machines, IIoT applications can benefit from edge computing [4]. A significant amount of research has been conducted on blockchain since Bitcoin and other cryptocurrencies rose to prominence. In addition to Internet of Things, blockchain is also applicable to supply chains, healthcare, and healthcare data storage [5]. As reported by [6], [7], [8] blockchain could be used for several applications in the IoT.

Many aspects of our daily lives have been transformed by IoT technology, including supply chain management, healthcare, and RFID-based identity management. IoT can be used to enhance data analysis and modeling capabilities, often in conjunction with cloud computing. Numerous industries are able to grow as a result of this [9]. The majority of IoT systems, however, are based on centralized platforms, putting security and privacy at risk. Because sensitive personal, financial, and medical information is stored and collected centrally, a centralised architecture is prone to data breaches and unauthorised access [10],[11].

The decentralized and immutable nature of distributed ledgers makes blockchain technology an attractive solution to these issues [12]. Blockchain technology enables IoT devices to be synchronised and for real-time data to be exchanged without relying on third parties [13],[14]. In addition to enhancing security and privacy, the Practical Byzantine Fault Tolerance (PBFT) algorithm also improves reliability. There are, however, some drawbacks to blockchain, including DDoS attacks targeting mempools, miners, and users. These attacks can increase transaction costs and make the network more susceptible to spam.

With IoT devices generating large amounts of data, security management becomes more complex, requiring efficient analysis and processing of data [15]. It can detect zero-day attacks and advanced persistent threats (APTs) that traditional approaches overlook by analyzing patterns and behaviors [16]. The integration of ML with IDS raises additional

challenges, such as ensuring data privacy and sharing Cyber Threat Intelligence (CTI) among organizations [17].

2. Related Work

According to [18], DDAML is an algorithm for discovery. In the study, machine learning algorithms and MLPs were used to identify DDoS attacks. The best ROC curve was achieved by DDAML when compared to SVM, RF, KNN, and LR algorithms. DDAML and NB algorithms have AUCs of 0.912, while SVM, CIC-SVM, and DDADA have AUCs of 0.908. There is an AUC of 0.891 for NB, 0.893 for SVM, 0.895 for CIC-SVM, and 0.899 for DDADA. Gradient-boosted machines (GBMs) are proposed to improve intrusion detection systems (IDS) using an anomaly-based approach [19]. The performance metrics of GBM technology are then compared with those of well-known classifiers to determine its effectiveness. On the NSL-KDD, UNSW-NB15, and GPRS datasets, holdout and tenfold cross-validation are currently the most effective methods for achieving the best results [20]. As part of the detection process combines genetic algorithms (GAs) with optimised gradient-boosted decision trees (OGBDTs). Improved categorization was achieved through enhanced African buffalo optimizations (EABOs). MLTs and conventional IDSs (OGBDTs) were compared. A comparison of accuracy, precision, recall, and F-scores was performed on the UNBS-NB 15, KDD 99, and CICIDS2018 datasets [21]. Across all datasets, the suggested IDS predicts attacks the fastest and detects them most effectively. It was found and discussed in [22] that IoT anomalies were reproduced using a virtual network for message queuing telemetry transport (MQTT). The detection and prevention of DDoS attacks was analyzed using various machine learning algorithms, including multilayer perceptrons (MLPs), naive bayes (NBs), decision trees (DTs), and artificial neural networks (ANNs). To analyze network traffic, 4998 records were analyzed, 34 characteristics were identified, and eight types of traffic were examined. Its accuracy was 99.94%, which was the highest of any classifier [23]. It is possible to launch decentralized attacks on internal and external infrastructures when IoT devices communicate improperly. It becomes difficult to develop an IoT detection mechanism when these challenges are present [24]. In a project called [25] lightweight algorithms are used to encrypt and validate IoT transactions. According to this work [26], a confidence-validation model is developed to determine whether a node is benign or malignant using probability calculations. Their source-embedded scheme embeds the hash of a traversed node's identification into the data packet for continued tracking. A data packet's path is verified to ensure data integrity. The current proposal actively maintains defences even during an attack, similar to the previous proposal. Contrary to that, the proposal presented here optimizes defenses and provides solutions with constant resource usage.

An author [27] proposed an architecture for smart manufacturing quality control that leverages smart contracts, BC, and ML. A BC distributed ledger was also proposed for storing information about automation contracts among suppliers, manufacturers, and retailers. The authors presented a similar architectural diagram to illustrate how data can be preprocessed, features selected, and training and testing sets split to perform predictive analysis using XGBoost (a flexible, highly efficient algorithm that avoids overfitting).

3. Proposed Methodology

This section presents a high-level architecture for IoT, ML-based IDS, and XGBoost IDS, which are intended to address imbalanced multiclass classification. According to [28]. IoT high-level architecture consists of three layers: perception, network, and applications. Sensors, cameras, meters, and IoT devices are included in the perception layer. These devices serve primarily as sensors, environmental tracking devices, and raw material transporters. Lastly, we have the network layer, which may include Lora, 6LoWPAN,

Bluetooth, Wi-Fi, and IEEE 802.15.4 networks. As part of the sensory data transfer, these connectivity networks connect IoT devices at the perception layer with networks and processing layers [29]. An attack may be ransomware, DDoS, or Man-in-the-Middle in the network or processing layers. Data can be stolen, connections can be disrupted, or transmissions can be delayed [30].

A smart city, a smart home, an intelligent grid, a smart factory, and a VANET have application layers that ensure end-user requirements. Based on imbalanced multiclass classification, shows the XGBoost model used to classify IIoT-IDS. An IIoT IDS based on XGBoost is composed of an IIoT IDS, preprocessing, and classification evaluation.

- The data processing. In input datasets were normalized, labeled, and split into training and testing datasets. The input features are normalized by using a min-max scale:

$$X_i^A = \frac{x_i^A - min\left(x_i^A\right)}{max(x_i^A) - max(x_i^A)} \quad (1)$$

Using label encoding, we convert non-numerical data into numerical values so machine learning models can learn. The number of different values contained in the target labels (Y) is encoded with an encoding between 0 and $(n_{classes} - 1)$,, where $n_{classes}$ represents the total number of possible values. The task was completed using Sclera's Label Encoder library in Python. To determine how many samples were collected for each step, the training data ratio was taken into account. Our data split for training and testing was 70:30.

- A sequence model based on XGBoost is XGBoost. In addition to learning from mistakes, fine-tuning hyperparameters, scaling imbalanced data, and handling null values, XGBoost is a powerful algorithm. Sequential ensemble methods such as boosting correct previous models' errors in sequence. Both classification and regression problems can be boosted with XGBoost, a sort of boosting algorithm. In XGBoost, trees are built by considering previous predictions and then maximising the predicted gain. An IIoT IDS dataset based on XGBoost is presented.

Adding new trees to the training process is iterative, allowing errors and residuals from prior trees to be fixed. To generate the final prediction, the previous trees are combined. Equation (2) shows the prediction value.

$$Y_{PT}(X) = Y_{PT-1}(X) + \alpha * f_n(X, wT) \quad (2)$$

In the case of T^{th}, Y_{PT} represents the prediction output, a represents the learning rate parameter, while f_n represents the function that predicts wT as the weight. During training, the model calculates the maximum gain or loss after continuously measuring node loss. The training process produces a tree of T nodes, each with a corresponding score at its leaf. Each tree's related score is added to yield the predicted value. It is important to avoid overfitting in XGBoost by balancing complexity decline and objective function. Regularizing the Taylor expansion boosts its performance. XGBoost is thus predicted by Equation (3).

$$Y_{Pt} = \sum_{t=1}^{T} f_n(X_t) \quad (3)$$

Assuming a t-th decision tree, $f_n(X_t)$ represents the function of the input, and Y_{Pt}represents the predicted result.

In Equation (4), the training error and regularisation terms comprise the training objective function of XGBoost.

$$Y_{Pt} = \sum_{i=1}^{n} L(Y_i, Y_{Pi}) \sum_{n=1}^{T} re(f_n) \quad (4)$$

A loss function's prediction and real value are measured by $\sum_{i=1}^{n} L(Y_i, Y_{Pi})$. There are three regularization terms for weak learners: $\sum_{n=1}^{T} re(f_n)$, $re(f_n) = \gamma^N + \frac{1}{2}\lambda|s^2|$,, and λ, which ensures that node scores are not too high.

As a delta change, $f_n(X)$ is used to approximate the T^{th} step object function with a second order Taylor equation. The previous $t-1$ predictor function is converted into a variable T^{th} weak learner function and $f_n(X)$ is used as a delta change.

The step t-th object function is approximated by a second order Taylor expression in XGBoost:

$$X_{Pt} = \sum_{i=1}^{N}\left(L\left(Y_i, Y_{Pt-1} + gif_n(X_i) + \frac{1}{2}h_i f_n^2(X_i)\right) + \sum_{i=1}^{t} re(f_i)\right) \quad (5)$$

The first derivative of this function is gi, and the second derivative is hi.

$$gi = \partial\gamma_{P_{t-1}} L(Y_i, Y_{Pt-1}) \quad (6)$$

$$h_i = \partial^2 Y_{Pt-1} L(Y_i, Y_{Pt-1}) \quad (7)$$

At the current t^{th} step, all values and predictions from the previous steps are known, so they are constant. If the constant terms are removed from the object function (since they do not affect it), then we obtain:

$$X_{Pt} = \sum_{i=1}^{n}\left(g_i, f_n(X_i) + \frac{1}{2}h_i, {f_n}^2(X_i)\right) + re(f_n) \quad (8)$$

This is how tree mapping is defined:

$$l_j = \{i \backslash q(X_i) = j\} \quad (9)$$

Tree leaf j corresponds to l_j, a set of all elements composing the j^{th} leaf, and $q(X_i)$ is the function mapping X_i to the leaf j.

$$f_n(X) = w_q(X) \quad (10)$$

The i^{th} leaf score is determined by the leaf w_i weight or output. A tree output is represented by $f_n(X)$, for example X

Next, we will regularize the object function:

$$X(t) = \sum_{i=1}^{n}\left(g_i, f_n(X_i) + \frac{1}{2}h_i, {f_n}^2(X_i)\right) + \gamma^T + \frac{1}{2}\lambda\sum_{i=1}^{T} w_j^2 \quad (11)$$

$$= \sum_{j-1}^{T}\left(\left(\sum_{i\epsilon I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\epsilon I_j} h_i + \lambda\right)w_j^2\right) + \gamma^T \quad (12)$$

A weak learner tree $f_t(X)$ has T leaves. There are two hyperparameters for regularization, γ and λ. Due to the loss purpose and step $t-1$prediction, the optimized object function for the t-th step depends on , w_i, g_i, and h_i,. As a result of this equation, we can determine the best w_i for minimising the object function:

$$\partial_{wi} Xt = 0 \quad (13)$$

It is optimal to have w :

$$w_j = -\frac{\sum_{i\epsilon I_j} g_i}{\sum_{i\epsilon I_j} h_i + \lambda} \quad (14)$$

In addition, the minimal object value is as follows:

$$Xt = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\epsilon I_j} g_i\right)^2}{\sum_{i\epsilon I_j} h_i + \lambda} + \gamma^T \quad (15)$$

To slit the weak learner, we first obtain the object function of the t-th step. The t-th tree is built next. Trees should be constructed in a manner that reduces as much value as possible from the object function. This tree is built by only allowing nodes to split and searching for the most efficient split. As a result, each split reduces the objective function value by (After Node Split) minus (Before Node Split).

$$G = \frac{1}{2}\left(\frac{\left(\sum_{i\epsilon I_j} g_i\right)^2}{\sum_{i\epsilon I_j} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_j} g_i\right)^2}{\sum_{i\epsilon I_j} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_j} g_i\right)^2}{\sum_{i\epsilon I_j} h_i + \lambda}\right) - \gamma \quad (16)$$

Split objects' gain (G) describes the reduction in function values. Splitting occurs on leaf I_L; leaf I_R; and leaf I. The Leaf Similarity Score (SS) is calculated as follows:

$$SS = \frac{\left(\sum_{i \epsilon I_j} g_i\right)^2}{\sum_{i \epsilon I_j} h_i + \lambda} \quad (17)$$

It is possible to express the splitting gain as follows:

It is calculated using the 1-order and 2-order derivatives of the chosen loss function, g_i and h_i, respectively. Log loss functions are used for classification (Equation (18)).

$$L = Y_i log(P_i) + (1 - y_i) log(1 - P_i) \quad (18)$$

Based on tree output prediction Y_i, $P_i = Sigmoid(Y_i)$ is the probability of the output class.

- The performance metric for mitigating IIoT attacks was energy consumption during transmission as part of the classification evaluation metrics. The ML approach in this study, however, accounted for other performance metrics. In CMs, there are four types of results: true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs). In addition to CM values, precision, recall, and F1 metrics were calculated based on these values.

Precision (P) is a measure of how accurate a model is at classifying.

$$P = \frac{TP}{TP+FP} \quad (19)$$

Detection of positive samples is measured by recall (R).

$$R = \frac{TP}{TP+FN} \quad (20)$$

F1 is derived by summing P and R using Equation (21).

$$F1 = 2 \times \frac{P \times R}{P+R} \quad (21)$$

A model's efficiency can be measured by how the learning curve performs during training. It represents the learning curve evaluation's performance through cross-validation.

3.1 Hyperparameter Tuning Using the HAFSO Algorithm

The detection rate of a DSAE model can be improved by optimising its hyperparameters using HAFSO. According to the proposed algorithm, the AFSO comprises two essential components: parameters and functions connected to fish behavior. The X represents a condition of the fish population, and the Y represents an objective function or feed focus. Among the four parameters, Step and Visual are crucial. Compared to fish, AFSO reaches the global optimal more quickly as the parameter's value increases. There are three essential functions in AFSO derived from fish behaviour: Search, Follow, and Swarm. Whenever a fish finds a high concentration of food, it goes directly there.

The fish will share food when they discover a lot. With AFSO adaption, feed concentrations are higher than they are now. In other situations, the fish move toward them, ensuring their survival. To prevent risks, they swarm, apparently. The Step and Visual values heavily influence a fish's behavior. It is possible to control search and swarm behaviour when the Visual size is narrow. There is a high focus on feed in the Visual. Based on whether the fish move towards it, the swarming performance would be determined. Global optima were obtained faster with higher Step and Visual values. Some optimization problems have been solved using AFSO as a result of this concept. Below is a mathematical expression for fish swarming:

$$F = (F_1, F_2, F_3 \dots, X_n) \quad (22)$$

Eq. (22), where F represents the fish, represents the visual place like this:

$$F_v = (F_{v1}, F_{v2}, F_{v3} \dots, F_{vn}) \quad (23)$$

The fish's visual location is indicated by F_v in Eq. (23). In regards to task (7), the following has been accomplished:

$$F_{vi} = F_{vi} + Rand(\,) \times step\ if\ f(F_{vi}) > f(F_v) \tag{24}$$

$i \in 1,2,..,n$ In Eq. (24), F_{vi} denotes the fish condition in Visual.

$$F_{next} = F + \left[\frac{(F_v - F)}{(\|F_v - F\|)}\right] \times Step \times Rand(\,) \tag{25}$$

The following fish in Visual is represented by F_{next} in Eq. (10). A fish's state is indicated by equations (22) and (23); equation (24) shows how equations (22) and (23) work together and illustrates the cowardly characteristics of AFSO. According to equation (25), the next fish varies with the distance between the previous two fish. Search behavior is represented by the Search Function in Eq. (24). When $F_i < F_j$ then (12) is applied. F_i and F_j indicate the concentrations of food in the present and in the future.

$$F_i^{(t+1)} = F_i^t + \left[\frac{(F_v - F)^{(t)}}{(\|F_v - F\|^{(t)})}\right] \times Step \times Rand(\,) \tag{26}$$

Alternatively, you can perform (9) by picking a state $F_i$arbitrarily and inspecting the outcome using equation (26).

$$F_i^{(t+1)} = F_i^t + Visual \times Rand(\,) \tag{27}$$

$$F_i^{(t+1)} = X_i^t + \left[\frac{(F_v - F)^{(t)}}{(\|F_c - F_c\|^{(t)})}\right] \times Step \times Rand(\,) \tag{28}$$

This equation (29) is called the Swarm Function and represents swarming behaviour. Once all conditions are met, carry out (13) as soon as possible. A point is currently in state $F_i(d_{ij} < Visual)$,, or else use the Search Function:

i. $(n_f/n) < \delta$

ii. $F_c > F_i$; F_c signifies the central food concentration

$$F_i^{(t+1)} = F_i^t + \left[\frac{(F_j - F_i)^{(t)}}{\left(\|F_j - F_i\|^{(t)}\right)}\right] \times Step \times Rand(\,) \tag{29}$$

The Follow Function represents this behavior. Perform (14) if the entire condition is met, or search function: Currently, the point is at $F_i(d_{ij} < Visual)$, $(n_f/n) < \delta$ and $F_j > F_i$. Eqs. (27)- (29) represent swarm behaviour, following behaviour, and search behaviour, respectively. This would be followed by achieving the related state. Then Eqs. (26) and (27) are applied. Eq. (24) is repeated until the condition is met. Repeat the process until the optimal point is reached. By meeting these criteria, the existing optimal value based on the obtained results will be improved. After meeting the ending condition, the result is recorded.

As part of the HAFSO algorithm, a chaotic approach is used. Chaotic logistic systems exhibit complex dynamics and are most commonly used in practice. Depending on the initial conditions, the chaotic system exhibits several properties. Signals produced by deterministic systems exhibit genus-randomness, and the signal curve depends on the initial value and the chaos-mapping parameter.

$$\lambda_{i+1} = \mu \times \lambda_i \times (1 - \lambda_i) \tag{30}$$

Where $\lambda \in [0,1], i = 0,1,2,\cdots,\mu$ is within [1, 4]. According to the study, should be closest to 4, and should be closest to the average distribution within [0,1].

Conversely, a value of 4 results in chaos. Optimal solutions and convergence rates in intelligent optimisation techniques are affected by the initial population. AFSO exploits solution space to improve its efficiency, which is achieved through logistic chaotic mapping.

HAFSO algorithm relies heavily on fitness selection. An assessment of a candidate solution's aptitude (quality) is made using solution encoding. As a primary condition, the accuracy of the proposed fitness function is considered.

$$Fitness = \max(P) \quad (31)$$

$$P = \frac{TP}{TP+FP} \quad (32)$$

According to the expression, TP corresponds to true positives, and FP to false positives.

4. Result and Discussion

Figure 2 presents model accuracy across six ID configurations (ID 10-ID 60) over training epochs 0-50. Initially, accuracy values are low and vary significantly across IDs, with ID 20 and ID 40 starting at just 0.25 and 0.2, respectively.

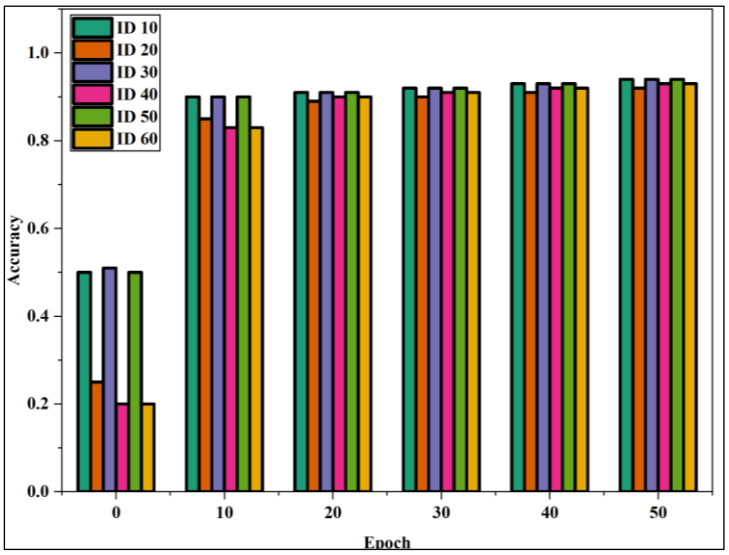

Figure 2: Model Accuracy Across Epochs for Different ID Configurations.

The True Positive Rate (TPR) is shown in Figure 3 as a function of False Positive Rate (FPR). At FPR = 0, the model fails to identify any true positives (TPR = 0). As FPR increases to 0.2 and 0.4, TPR rises sharply to 0.88 and 0.85, respectively, indicating improved detection at the cost of more false alarms. From FPR = 0.6 onward, the model achieves perfect sensitivity (TPR = 1), suggesting that all actual positives are correctly identified, albeit with increasing false positives. In this example, sensitivity versus specificity are traded off in a typical classification process.

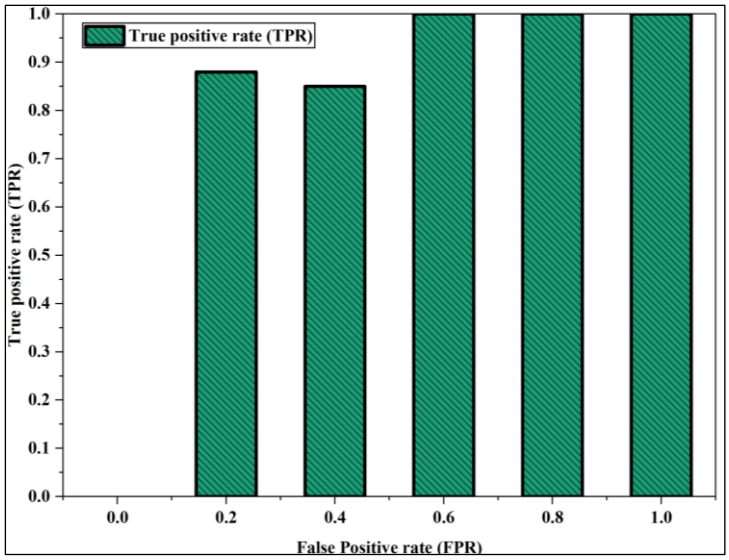

Figure 3: A comparison of true positives and false positives for the purpose of evaluating a model.

Figure 4 shows a consistent decrease in latency as the number of training epochs increases. Starting at 60 ms with 0 epochs, latency drops progressively—reaching 55 ms at 10 epochs, 45 ms at 20, and ultimately 25 ms by epoch 50. This trend suggests that continued training improves model efficiency, resulting in faster response times.

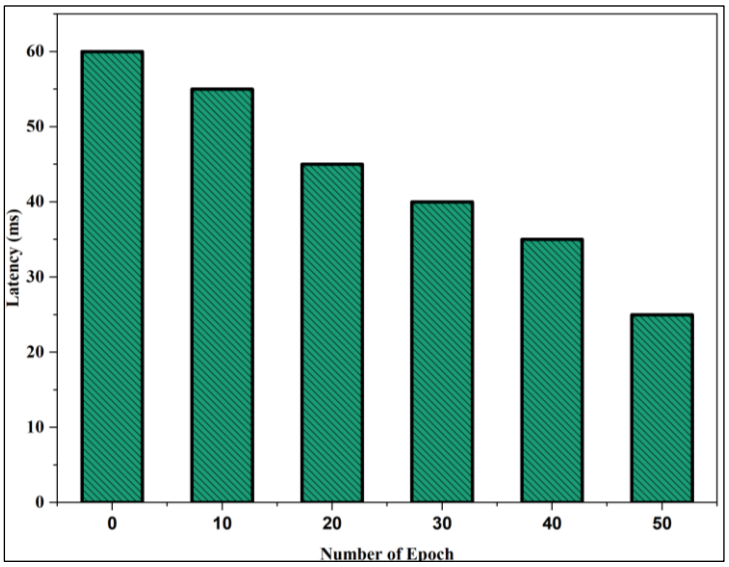

Figure 4: Latency Reduction Over Training Epochs.

Figure 5 highlights the model's performance across Normal, Attack, and Average categories using four key metrics. Normal consistently scores the highest, with 98% accuracy, 97% precision, 99% recall, and a F1-score of 97.98%. Attack performance is slightly lower but still strong, while the Average category reflects a modest drop across all metrics. As a result, the model performed most effectively in identifying normal instances and slightly less effectively in detecting attacks.

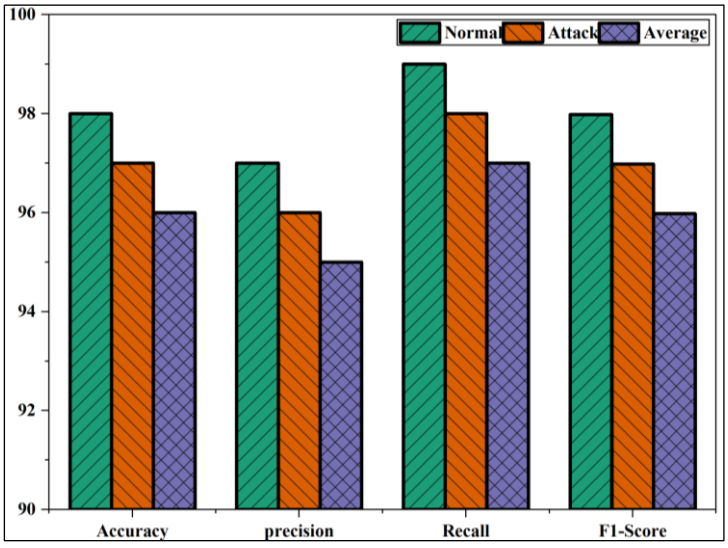

Figure 5: Comparative Performance Metrics Across Normal, Attack, and Average Categories.

5. Conclusion

By integrating blockchain and machine learning technologies, this research presents an intelligent, secure intrusion detection system for IIoT environments. In addition to addressing multiclass imbalance, XGBoost and HAFSO optimization enhance detection precision. In addition to minimizing single points of failure and ensuring data integrity, blockchain technology enables automated responses through smart contracts. Performance evaluations confirm the model's high accuracy and low latency, with superior classification results across normal and attack scenarios. As a whole, the proposed hybrid framework provides scalable, resilient, and trustworthy infrastructure for the next generation of IIoT security.

REFERENCES

1. G. Wang, "Sok: Applying blockchain technology in industrial internet of things," *Cryptology ePrint Archive*, 2021, Accessed: Oct. 14, 2025. [Online]. Available: https://eprint.iacr.org/2021/776
2. M. Yli-Ojanperä, S. Sierla, N. Papakonstantinou, and V. Vyatkin, "Adapting an agile manufacturing concept to the reference architecture model industry 4.0: A survey and case study," *Journal of industrial information integration*, vol. 15, pp. 147–160, 2019.
3. H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus Inf Syst Eng*, vol. 6, no. 4, pp. 239–242, Aug. 2014, doi: 10.1007/s12599-014-0334-4.
4. P. Kumar, R. Kumar, G. P. Gupta, and R. Tripathi, "A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing," *Trans Emerging Tel Tech*, vol. 32, no. 6, p. e4112, Jun. 2021, doi: 10.1002/ett.4112.
5. M. Borhani, M. Liyanage, A. H. Sodhro, P. Kumar, A. D. Jurcut, and A. Gurtov, "Secure and Resilient Communications in the Industrial Internet," in *Guide to Disaster-Resilient Communication Networks*, J. Rak and D. Hutchison, Eds., in Computer Communications and Networks. , Cham: Springer International Publishing, 2020, pp. 219–242. doi: 10.1007/978-3-030-44685-7_9.
6. M. Shuaib *et al.*, "Identity Model for Blockchain-Based Land Registry System: A Comparison," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 5670714, Jan. 2022, doi: 10.1155/2022/5670714.
7. S. Wadhwa, S. Rani, G. Kaur, D. Koundal, A. Zaguia, and W. Enbeyle, "HeteroFL Blockchain Approach-Based Security for Cognitive Internet of Things," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 5730196, Jan. 2022, doi: 10.1155/2022/5730196.
8. M. Saeed, M. Aftab, R. Amin, and D. Koundal, "Trust Management Model in IoT: A Comprehensive Survey," in *Innovations in Bio-Inspired Computing and Applications*, vol. 419, A. Abraham, A. M. Madureira, A. Kaklauskas, N. Gandhi, A. Bajaj, A. K. Muda, D. Kriksciuniene, and J. C. Ferreira, Eds., in Lecture Notes in Networks and Systems, vol. 419. , Cham: Springer International Publishing, 2022, pp. 675–684. doi: 10.1007/978-3-030-96299-9_64.
9. J. M. Müller, D. Kiel, and K.-I. Voigt, "What Drives the Implementation of Industry 4.0? The Role of Opportunities and Challenges in the Context of Sustainability," *Sustainability*, vol. 10, no. 1, p. 247, Jan. 2018, doi: 10.3390/su10010247.
10. R. F. Mansour, "Blockchain assisted clustering with Intrusion Detection System for Industrial Internet of Things environment," *Expert Systems with Applications*, vol. 207, p. 117995, Nov. 2022, doi: 10.1016/j.eswa.2022.117995.
11. W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Future Generation Computer Systems*, vol. 96, pp. 481–489, Jul. 2019, doi: 10.1016/j.future.2019.02.064.
12. R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 55–68, Jun. 2022, doi: 10.1016/j.jpdc.2022.01.030.

13. R. Kumar and R. Tripathi, "Data Provenance and Access Control Rules for Ownership Transfer Using Blockchain:," *International Journal of Information Security and Privacy*, vol. 15, no. 2, pp. 87–112, Apr. 2021, doi: 10.4018/IJISP.2021040105.
14. S. A. Latif *et al.*, "AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems," *Computer Communications*, vol. 181, pp. 274–283, Jan. 2022, doi: 10.1016/j.comcom.2021.09.029.
15. S. V. N. Santhosh Kumar, M. Selvi, and A. Kannan, "A Comprehensive Survey on Machine Learning-Based Intrusion Detection Systems for Secure Communication in Internet of Things," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, p. 8981988, Jan. 2023, doi: 10.1155/2023/8981988.
16. N. I. Che Mat, N. Jamil, Y. Yusoff, and M. L. Mat Kiah, "A systematic literature review on advanced persistent threat behaviors and its detection strategy," *Journal of Cybersecurity*, vol. 10, no. 1, p. tyad023, Jan. 2024, doi: 10.1093/cybsec/tyad023.
17. P. Alaeifar, S. Pal, Z. Jadidi, M. Hussain, and E. Foo, "Current approaches and future directions for Cyber Threat Intelligence sharing: A survey," *Journal of Information Security and Applications*, vol. 83, p. 103786, Jun. 2024, doi: 10.1016/j.jisa.2024.103786.
18. S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2020, doi: 10.1109/ACCESS.2019.2963077.
19. P. Rani, S. Sachan, A. A. Alwan, J. Jagtap, N. Kaur, and H. Alabdeli, "EECP2-IOT: Energy-Efficient Clustering Protocols for Performance Optimization of Internet of Things-Based Heterogeneous and Homogeneous Wireless Sensor Networks," in *Proceedings of the 3rd International Conference on Internet of Things, Communication and Intelligent Technology*, vol. 1366, J. Dong, L. Zhang, and T. Zheng, Eds., in Lecture Notes in Electrical Engineering, vol. 1366. , Singapore: Springer Nature Singapore, 2026, pp. 575–590. doi: 10.1007/978-981-96-2771-4_50.
20. R. Rehyadd and P. Rani, "Sign-Based Encryption-Enabled Reliable Data Communication for Mobile Ad Hoc Networks," in *2025 International Conference on Next Generation of Green Information and Emerging Technologies (GIET)*, Gunupur, India: IEEE, Aug. 2025, pp. 1–7. doi: 10.1109/GIET65294.2025.11234764.
21. P. Rani *et al.*, "Federated Learning-Based Misbehavior Detection for the 5G-Enabled Internet of Vehicles," *IEEE Trans. Consumer Electron.*, vol. 70, no. 2, pp. 4656–4664, May 2024, doi: 10.1109/TCE.2023.3328020.
22. S. Mishra, A. Albarakati, and S. K. Sharma, "Cyber Threat Intelligence for IoT Using Machine Learning," *Processes*, vol. 10, no. 12, p. 2673, Dec. 2022, doi: 10.3390/pr10122673.
23. P. Rani, J. Kumar, S. Singh, P. Dey, and L. H. Jasim, "Decoding the Aspects of Intelligent Traffic Management," in *Artificial Intelligence Technologies for Smart and Sustainable Urban Transportation*, 1st ed., P. Raj, S. Yadav, M. K. Mishra, S. P. Yadav, and V. H. C. Albuquerque, Eds., Wiley, 2025, pp. 287–300. doi: 10.1002/9781394346776.ch17.
24. K. Yang, J. Ren, Y. Zhu, and W. Zhang, "Active Learning for Wireless IoT Intrusion Detection," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 19–25, Dec. 2018, doi: 10.1109/MWC.2017.1800079.
25. J. H. Jeon, K.-H. Kim, and J.-H. Kim, "Block chain based data security enhanced IoT server platform," in *2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand: IEEE, Jan. 2018, pp. 941–944. doi: 10.1109/ICOIN.2018.8343262.
26. S. Suhail, C. S. Hong, M. A. Lodhi, F. Zafar, A. Khan, and F. Bashir, "Data trustworthiness in IoT," in *2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand: IEEE, Jan. 2018, pp. 414–419. doi: 10.1109/ICOIN.2018.8343151.
27. Z. Shahbazi and Y.-C. Byun, "Integration of Blockchain, IoT and Machine Learning for Multistage Quality Control and Enhancing Security in Smart Manufacturing," *Sensors*, vol. 21, no. 4, p. 1467, Feb. 2021, doi: 10.3390/s21041467.
28. P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Computers and Electrical Engineering*, vol. 105, p. 108543, Jan. 2023, doi: 10.1016/j.compeleceng.2022.108543.

29. J. Sengupta, S. Ruj, and S. Das Bit, "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT," *Journal of Network and Computer Applications*, vol. 149, p. 102481, Jan. 2020, doi: 10.1016/j.jnca.2019.102481.
30. A. Singh *et al.*, "Resilient wireless sensor networks in industrial contexts via energy-efficient optimization and trust-based secure routing," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 3, p. 132, May 2025, doi: 10.1007/s12083-025-01946-5.