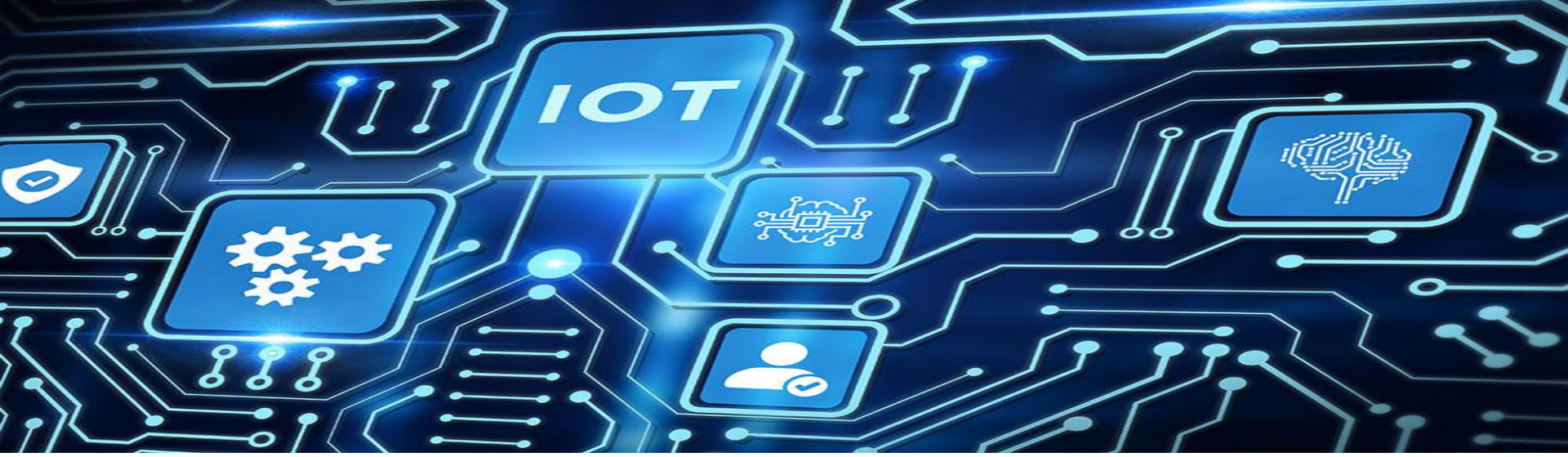


“Block Hunter: Federated Learning for Cyber Threat Hunting in Blockchain-Based IIoT Networks”

Abbas Yazdinejad , Ali Dehghantanha , Senior Member, IEEE, Reza M. Parizi , Senior Member, IEEE, Mohammad Hammoudeh , Senior Member, IEEE, Hadis Karimipour , Senior Member, IEEE, and Gautam Srivastava , Senior Member, IEEE

2022.10.17

Professor: 박종혁

Presented by:

YOTXAY SANGTHONG

Seoul National University of Science and Technology, Seoul, South Korea.

Table Contents

Abstract

I. Introduction

II. Related Work

III. Proposed Block Hunter Framework

IV. Methodology

V. Discussion and Evaluation

VI. Conclusion

Abstract

- In this paper, they use federated learning to build a threat hunting framework called block hunter to automatically hunt for attacks in blockchain-based IIoT networks.

- Block hunter utilizes a cluster-based architecture for anomaly detection combined with several machine learning models in a federated environment.

- The results prove the efficiency of the block hunter in detecting anomalous activities with high accuracy and minimum required bandwidth.

I. Introduction

- With the increasing use of IIoT devices, the world is inevitably becoming a smarter interconnected environment; especially factories are becoming more intelligent and efficient as technology advances.
- As the number of IIoT devices in smart factories increases, the main issue will be storing, collecting, and sharing data securely. Industrial, critical, and personal data are, therefore, at risk in such a situation.
- Blockchain technology can ensure data integrity inside and outside of smart factories through strong authentication and ensure the availability of communication backbones. Despite.
- Even though blockchain technology is a powerful tool, it is not protected from cyberattacks either.
- For example, a 51% cyberattack [2] on Ethereum Classic (ETC), and three consecutive attacks in August of 2020 [5], which resulted in the theft of over $5M worth of cryptocurrency, have exposed the vulnerabilities of this blockchain network.
- The main objective of this article is to detect suspicious users and transactions in a blockchain-based IIoT network specifically for smart factories.

I. Introduction (Cont.)

- They leveraged machine learning (ML) algorithms to identify out-of-norm patterns to detect attacks and anomalies on blockchain.

- ***The main contributions:***

  1) Used a cluster-based architecture to formulate an anomaly detection problem, to increased the hunting efficiency in terms of bandwidth reduction and throughput in IIoT networks.

  2) Federated design model to detect anomalous behavior in IIoT devices, to provided a privacy-preserving feature when using ML models.

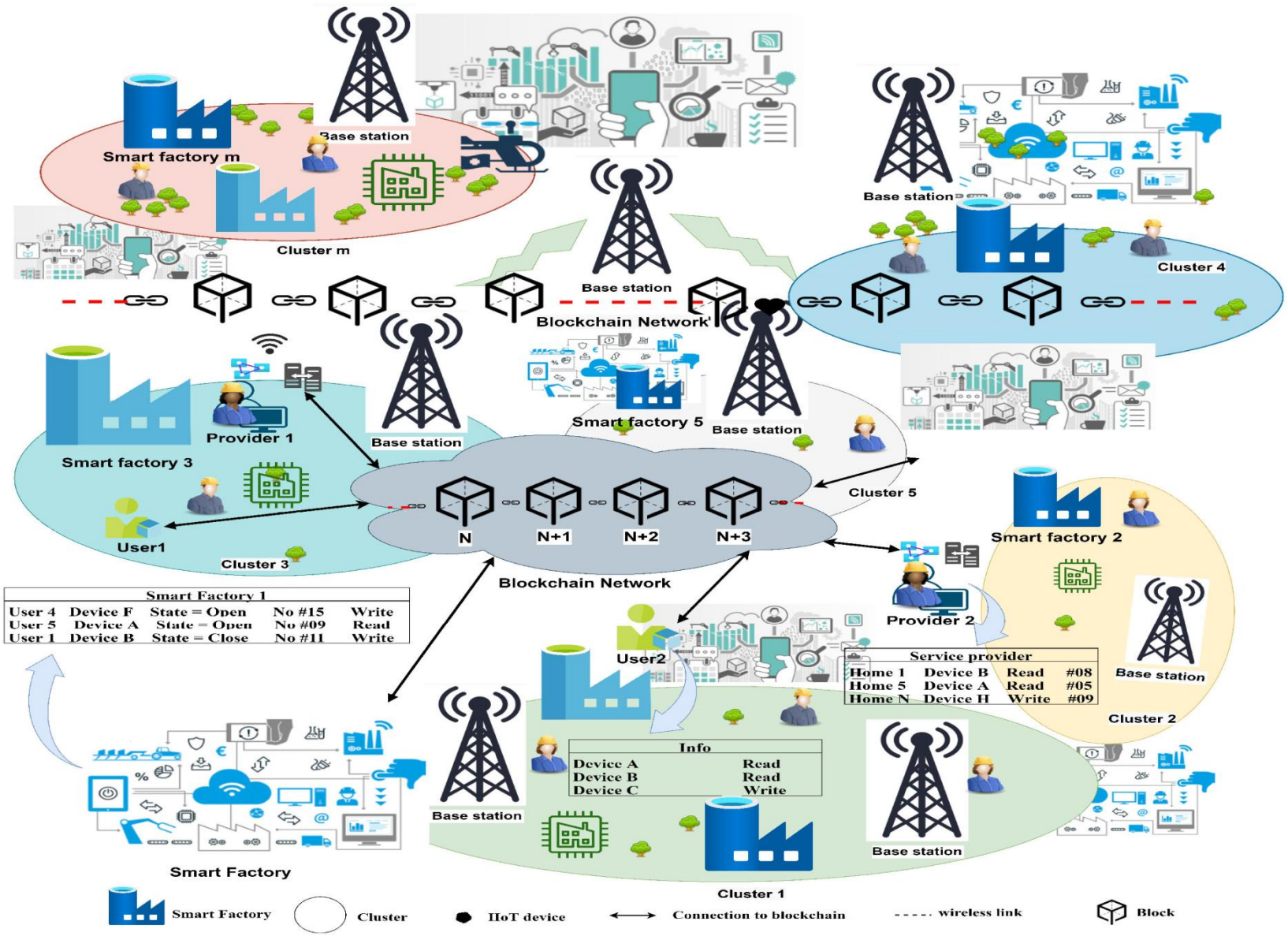
  3) Implementation of various anomaly detection algorithms such as clustering-based, statistical, subspace-based, classifier-based, and tree-based for efficient anomaly detection.

  4) The impact of block generation, block size, and miners on the Block Hunter framework and the performance measurements like Accuracy, Precision, Recall, F1-score, and True Positive Rate anomaly detection.

II. Related Work

- The research relating to anomaly detection, especially in relation to blockchain and FL.
- Authors [13], proposed a framework as blockchain anomaly detection (BAD) to detect anomalies in blockchain based systems. In [14], suggested blockchain and anomaly detection systems that recognize frauds when IoT meter data are tampered with.
- Sayadi et al. [15] proposed an algorithm for anomaly detection over bitcoin electronic transactions. They examined the one-class support vector machines and the K-means algorithms to group outliers similar in both statistical significance and type.
- Also [16], suggested an approach based on the semantics of anomalies in blockchain-based IoT networks. In [7], provided an FL approach to anomaly detection for smart buildings that made use of an additional recurrent neural network for a privacy-by-design approach.
- Nguyen et al. [19], presented a self-learning federated system for detecting anomalies in IoT networks.
- Authors in [20] put forward an approach via FL for detecting abnormal client behavior. In [21] involved the use of Deep Learning and blockchainbased FL to detect COVID-19.
- Chai et al. [22] proposed a hierarchical blockchain framework and FL to learn and share environmental data. Furthermore, the authors in [23] investigated on how FL could supply better cybersecurity and prevent various cyberattacks in real time.

III. Proposed Block Hunter Framework in IIoT Networks



Smart Factory 1				
User 4	Device F	State = Open	No #15	Write
User 5	Device A	State = Open	No #09	Read
User 1	Device B	State = Close	No #11	Write

Service provider			
Home 1	Device B	Read	#08
Home 5	Device A	Read	#05
Home N	Device H	Write	#09

Info	
Device A	Read
Device B	Read
Device C	Write

Fig. 1. Overview of the blockchain-based IIoT network for smart factories.

III. Proposed Block Hunter Framework in IIoT Networks

- The cluster-based architecture combines users, base stations, WiFi, service providers, and smart factories connected to the blockchain network. Smart factories include several smartconnected devices.
- The service provider can collect sensor data in smart factories and use them based on their applications and services.
- There are several inputs and outputs in a transaction. Blocks consist of a list of transactions, a reference to the previous block, and a hash. Every block is made up of transactions that the block creator, referred to as the miner, has accepted into its memory pool from the previous block.
- Detecting anomalous activities is a significant contributor to automatically protecting a system from unexpected attacks. Anomalies in blockchain must be detected by sending each block of data to a central server for each block update.
- This is not efficient and also imposes privacy concerns. FL solutions are promising in tackling this issue. They used FL to update the model frequently and to obtain a global model for detecting an anomaly.
- After learning about each smart factory's data, devices, and service provider, the model's parameters will be sent to the parameter server for aggregation and to update our general model. They provided the details of implementing the Block Hunter framework in the following subsections.

III. Proposed Block Hunter Framework in IIoT Networks

A. Role of FL to Detect Anomalies in Block Hunter

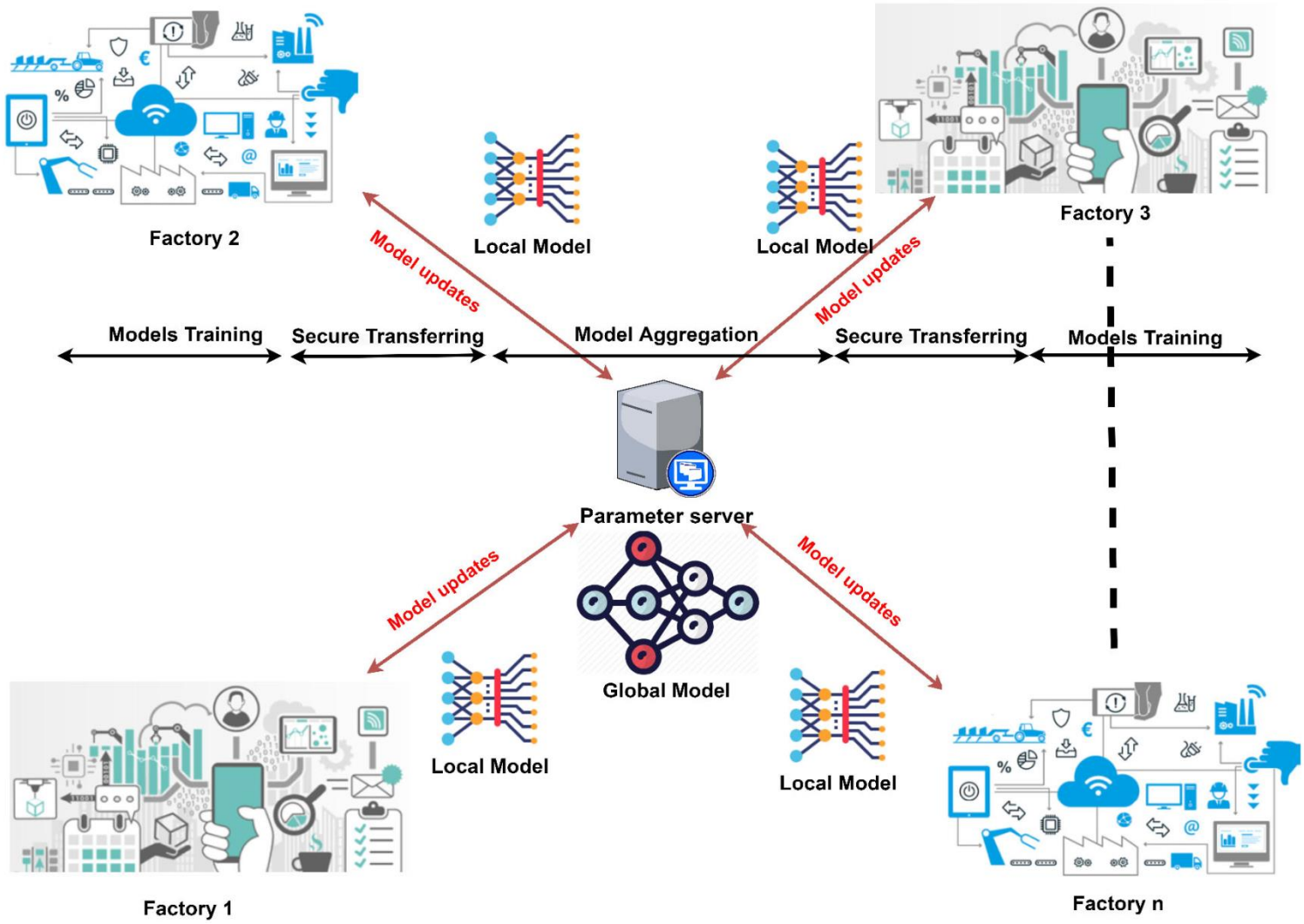


Fig. 2. Proposed federated anomaly detection in the Block Hunter framework.

III. Proposed Block Hunter Framework in IIoT Networks

A. Role of FL to Detect Anomalies in Block Hunter (Cont.)

- To summarize, steps for FL in the Block Hunter framework are categorized as follows.

  1) ***Federation construction:*** The subset of smart factory members, cluster, is selected to receive the model locally.
  2) ***Decentralized training:*** When a cluster of smart factories is selected, it updates its model using its local data.
  3) **Model accumulation:** Responsible for accumulating and merging the data models. Data are not sent and integrated from the federation to the server individually.
  4) ***Model aggregation (FedAvg):*** Parameter server aggregates model weights to compute an enhanced global model.

- At runtime, pretrained models as local models are sent to clusters in the Block Hunter framework from the Parameter Server, considering blockchain-based IIoT networks environments.

- The local models are sent to smart factories for training based on local epochs. Then, the parameters and hyperparameters will be forwarded to the Parameter Server for aggregating model weights to compute the global model.

- The global model is an ML model that holds in the Parameter Server to update its parameters. When a new cluster joins our framework, the latest global model will be sent to that cluster as a pretrain model.

III. Proposed Block Hunter Framework in IIoT Networks

A. Role of FL to Detect Anomalies in Block Hunter (Cont.)

- They proposed federated anomaly detection algorithm for smart factories in IIoT networks is shown in Algorithm 1.

Algorithm 1: Federated Anomaly Detection Model.

Input: Pre-trained model
Output: Global anomaly-detection model
Initializing: ($t = 0$) // start to set values
    Parameter server update (W)
    Define initial values = B, E, h, C, K
        initial local models ()
    Model (m_i) = set parameters ($w_1, \ldots, w_n$)
    Client model update
    Local models→ smart factories' clusters
Federated Training: // beginning FL method
        While ($K > 0$)
            Get (local models)
            For local epoch E
                For each batch 1 to b
                    Run local models
                Obtain and set model parameters
            Return (Model parameters)
            Parameter server = `FedAvg` (w)
    Update (w, m)
      Decrypted (w, m)
      D = split Data into batches of size B
        for local epoch E from 1 to B do
            for batch $x <= B$ do
      return w and D to server
    Return updated Model
FedAvg: // model aggregation
        Server (Initialize w_0)
        encryption (Homomorphic)
            P = compute loss (w, b)
        for round $c = 1$ to k do
                Server: Send w_{t-1} to Smart factory $i - 1$
                $E = E + 1$
            Parameter server = Update ($w_{k-1}, \ldots, w_k$)

III. Proposed Block Hunter Framework in IIoT Networks

B. Anomaly Detection in Blockchain-Based IIoT Network

- The development of an anomaly detection framework for the blockchain-based IIoT networks in providing a new decentralized system based on FL that leverages all smart factory data while protecting their privacy.
- Indeed, a global MLmodel can use all of the collected information from previous forks to detect anomalies during training. This approach has the advantage that while attacks may only happen once, they behave the same way when repeated against other smart factories over time.
- Hence, information on past attacks may help us blacklist them and prevent them from occurring in the future. The advantage of FL is clear since it will train the global ML model for anomaly detection. Their solution considers smart factories' data and chain forks.
- They collect, enrich, and share such information with other local MLmodels across the network. They used the specific information for training anomaly detection in each local ML model that contains sensitive smart factory data, the features of previous forks, and the number and type of malicious transactions that occurred.
- As a result, they can hunt an anomaly by the Block Hunter in a blockchain-based IIoT network for smart factories. To protect the privacy of the data, we only share the parameters of trained models instead of the original data from smart factories and their blockchain.
- This work aimed to train a global anomalous detection model through locally trained sub-protocol models based on the Block Hunter framework.

III. Proposed Block Hunter Framework in IIoT Networks

C. Network Model and Topology Design

- Wireless sensor networks have a variety of topologies, which affect their performance and behavior. Some of the metrics include throughput, reliability, energy consumption, and latency [26].
- Therefore, they proposed blockchain technology's cluster-based formation model for smart factories. Cluster-based architecture provides more efficient use of resources [27] and throughput during the blockchain run in each smart factory.
- Clustering reduces the computational complexity in the creation of the underlying network through a hierarchical approach [26]. It is especially so with blockchain-based IIoT networks that are expected to encompass large numbers of individual devices.
- Also, they believe that cluster-based architecture will enable us to hunt and manage anomalies better in each smart factory zone and increase the network's throughput.
- Based on the target Block Hunter framework, which can be represented as a directed graph G = (S,D) with D being the set of IIoT devices, representing smart devices, D = {d1, d2, . . . , dn}, and S = {s1, s2, . . . , sn} is the set of smart factories in each cluster.

$$D_f(D_{ki}, D_{kj}) = \sqrt{\sum_{i,j=1}^{K} (S_{nj} - S_{ni})^2 \times (D_{kj} - D_{ki})^2}. \quad (1)$$

III. Proposed Block Hunter Framework in IIoT Networks

C. Network Model and Topology Design (Cont.)

Algorithm 2: Cluster Formation Strategy in Block Hunter.

Input:
  Get $= (S, D)$
Initialize:
  Define initial values
  Set Number of Cluster = K
  Get loc $= S = \{s_1, s_2, \ldots, s_n\}$ // location of smart factories
  Get loc $= D = \{d_1, d_2, \ldots, d_n\}$ // location of devices
  $s_n = \bigcup_j^k s_j \times D_{kj}$ // Deployed clusters and smart factories
Main():
  Get (K, S, D)
  While $(K > 0)$
  {
    For z **each** K
      Comput $= D_{fz}$
        $(D_{ki}, D_{kj}) = \sqrt{\sum_{i,j=1}^{K}(S_{nj} - S_{ni})^2 \times (D_{kj} - D_{ki})^2}$ // Calculating distance for finding neighboring devices. The presence or absence of devices in the space of (i, j)
      Set_area $= K_z$
    Client distance update
    Marge neighbor
  }
Return K

III. Proposed Block Hunter Framework in IIoT Networks

C. Network Model and Topology Design (Cont.)

- The clustering part is shown in Algorithm 2, and it can be considered a piece of the overall algorithm in the Block Hunter. Algorithm 2 collects the locations of smart factories and their IIoT devices.
- Based on (1), they measured each smart factories' distance and their devices and record it until they obtained the cluster-based architecture.
- Afterward, the cluster calculates a collection of S nearest smart factories for each IoT device, Setting model parameters in the parameter server and sending pretrained models to clusters happen during initializing.
- Next, local models are trained by clusters in the training step to aggregate models and update the global model parameters.

IV. Methodology

A. Neural Encoder–Decoder (NED) Model

- The proposed anomaly detection framework develops an NED model that summarizes the information about the blockchain's status and transactions, and then, rebuilds the initial data from this space.

- Encoding/decoding preserves the data's basic properties when the current status is consistent. Differently, anomalous situations exhibit inconsistent values, ultimately leading to a failing reconstruction.

- In an encoder– decoder, this quantity would be paraphrased as noise, and therefore, would be failing when reconstructing.

- Therefore, the difference between the initial and reconstructed values would highlight the anomalous and abnormal situation, thereby triggering an alert.

IV. Methodology (Cont.)

B. Isolation Forest (IF)

- The Isolation Forest (IF) model falls under the Tree-based anomaly detection algorithms category. The approach has gained much universal acceptance in recent years because it is unsupervised.

- IF is a concept based on the idea that it is more prudent to isolate data anomalies rather than generalize the norms. It is a recursive and random partitioning process to isolate the anomalous data point in the dataset until it simply describes the stored data.

- A tree structure represents the recursive partition. A forest of isolation trees is the foundation of the IF algorithm, where cells in the dataset are randomly selected from the data to form a forest of normal and outlier cells.

IV. Methodology (Cont.)

C. Cluster-Based Local Outlier Factor (CBLOF)

- This model belongs to the classifier-based algorithm based anomaly detection category.
- Within this algorithm's anomaly detection methodology, the data are clustered into clusters, based on which anomaly scores can be computed similar to those of the local outlier factor algorithm and so on.
- This algorithm's underlying principle of anomaly detection is based on clustering datasets together and created clusters using groups in a dataset by arbitrary clustering algorithms that assign a specific observation to a cluster.
- The clusters are sorted in each case corresponding to their respective sizes of |F1| > |F2| > . . . > |Fk| where F1, F2, . . . , Fk all represent the cluster for which number k is the cluster number [8].

$$\mathrm{CBLOF}(t) = \begin{cases} |Fk|.\mathrm{dis}(t, Fi) & \mathrm{t} \in \mathrm{Fi} \\ |Fk|.\min(\mathrm{dist}(t, Fi) & \mathrm{t} \in \mathrm{Fi}. \end{cases} \quad (2)$$

IV. Methodology (Cont.)

D. Principal Component Analysis (PCA)

- A PCA model is a subsequence-based anomaly detection algorithm. PCA is commonly considered a method to reduce the dimensionality algorithm.
- The variance-covariance of dataset characteristics can be used to construct new variables known as principal components, which are functions of original variables.
- The PCA algorithm detects anomalies by getting rid of any outliers. The outliers are determined by the Mahalanob is distance that is carried out repeatedly to eliminate all data points with high Mahalanob is distance values.
- Where S is a covariance matrix, xi is the measure of an observation of the I th feature in data, and x is the mean of all observations, Mahalanobis distance is denoted as follows:

$$D = \sqrt{(Xi - X)^T s^{-1} (Xi - X)}. \qquad (3)$$

IV. Methodology (Cont.)

E. K-means

- In the cluster-based detection algorithms category, K-means is a clustering-based algorithm. As one of the most popular clustering algorithms, K-means is also commonly used as an anomaly detection algorithm.

- It has been introduced as an unsupervised learning scheme. The data are divided into k different clusters, with each sample belonging to the cluster with the closest mean value within each cluster.

- Across clusters, there is a cluster centroid c, which is the mean of observations from each cluster in that cluster.

- When assessing the similarities among independent observations, the similarity measure employed is Euclidean distance, where Xi is the measurements and Ci is the centroids, and n outlines the number of independent measurements.

$$d^2(X, c) = \sum_{i=1}^{k} (Xi - ci)^2. \qquad (4)$$

V. Discuss and Evaluation

- This section evaluates the performance of the Block Hunter and provides results and discussion.

A. Experimental Setup

- Experimental setup on Intel(R) Core(TM) i7- 10700KF CPU at 3.80 GHz 3.79-GHz, Linux 64-bit operating system (Ubuntu 20.04), and equipped it with 16-GB DDR4 memory.

- To evaluate their network model and cluster-based topology design in the proposed framework, they applied the Bitcoin Simulator.1

Parameters	Description
Simulators	Bitcoin simulator/ NS3 / 5G-LENA
Operating system	Ubuntu 20.04
libraries	PySyft / Pythorch
Number of Clusters	50
Optimization method	SGD
local epoch	E = 4
Fraction of smart factories	$6e-3$
Mobility model	Random waypoint model
Traffic Type	Constant Bit Rate (CBR)
Number of IoT devices	5000
Block Size	1 MB, 2 MB, 4 MB, 8 MB, 16 MB
Number of Miners	16, 32, 64
Local epochs in each cluster	4
Learning rate	$3e-2$
Local mini-batch size	15
Federated approach	FedAvg

Table I. Federated Framework parameters

V. Discuss and Evaluation (Cont.)

B. Datasets

- The proposed framework is evaluated by two datasets on the blockchain side, providing conditions for blockchain adoption in smart manufacturing systems, and also two IIoT-related datasets for assessing Block Hunter for smart factories.

- The Bitcoin Transaction Dataset (BTD)4 designed for research on blockchain anomaly and fraud detection. It has been donated to the IEEE data port online community for academic exploration.

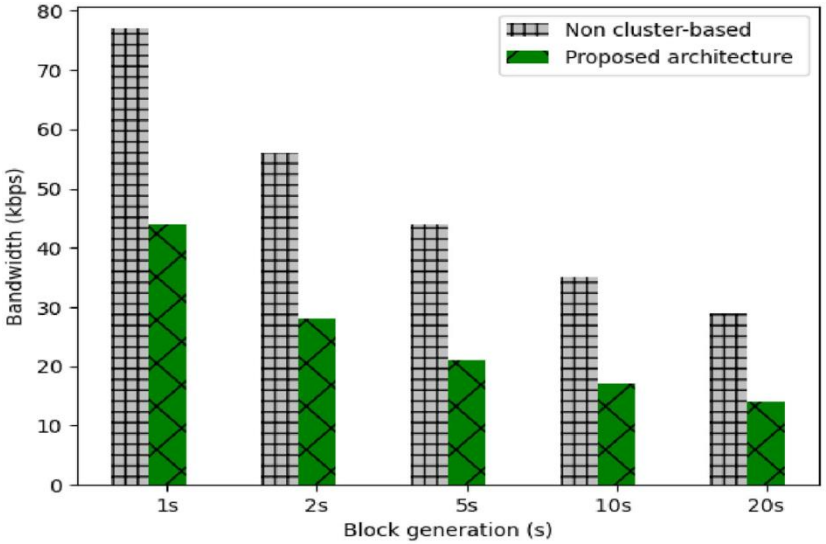
- Because the dataset is imbalanced and contains roughly 30 million transactions, it presents a challenge in creating an anomaly detection model that captures all of them.

- The dataset is an implementation of a research project that presents anomaly detection within the context of blockchain technology and its applications in the monetary domain.

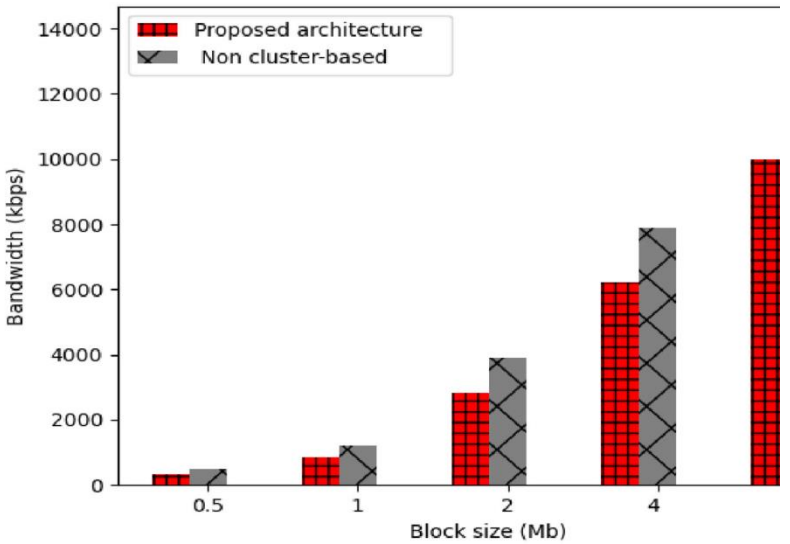
- It extracts blockchain data and uses ML techniques to hunt potentially malicious transactions.

V. Discuss and Evaluation (Cont.)

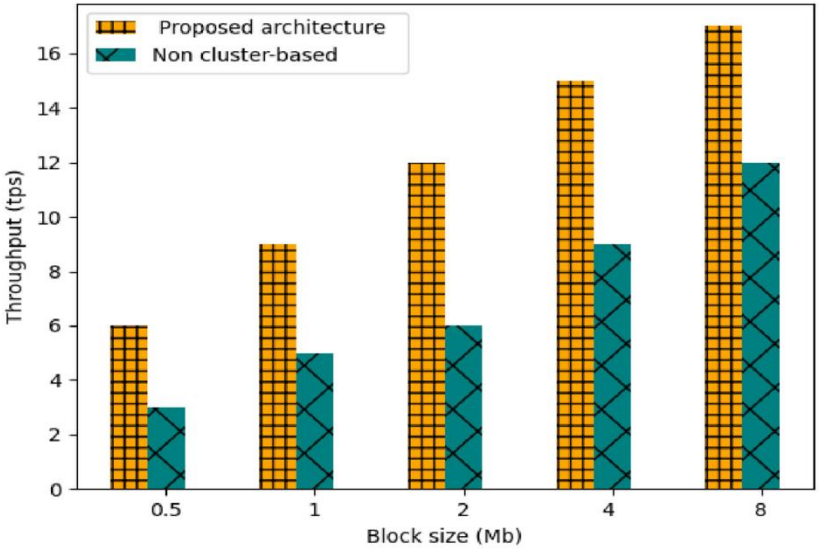
B. Datasets (Cont.)



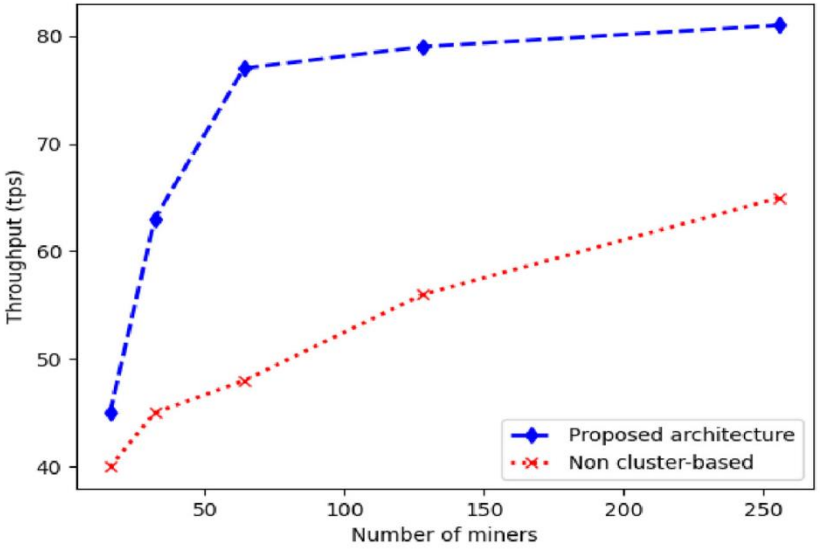
(a) Impact of the **Block generation on bandwidth consumption**



(b) Impact of the **Block size on bandwidth**.



(c) Impact of the **Block size on throughput**.



(d) Impact of number of **miners.**

Fig. 3. Cluster-based architecture evaluation.

V. Discuss and Evaluation (Cont.)

C. Experimental Analysis

- A cluster-based architecture provides more efficient use of resources and throughput during the blockchain run in smart factory applications. To evaluate the performance of the Block Hunter, cluster-based architecture, the simulation parameters are presented in Table I.

- To accomplish more realistic results, they did the simulation 20 times and designed another scenario as a noncluster model to compare the architectural models' performance during the simulation.

- The noncluster model combined blockchain technology with the standard network model and did not consider and divide it into cluster architecture. It has no features and typologies of cluster-based architecture such as adjacencies with other clusters or part of the network, flexibility, and scalability during run time.

- In the following, they addressed the impact of Block generation, the impact of the Block size, and the impact of the number of miners in the evaluation.

- In the proposed framework, the public blockchain network is deployed among clusters that include smart factories.

- They needed a public blockchain to allow any smart factories to join and keep the system completely decentralized. Additionally, public blockchains give all participants equal access to the chain.

V. Discuss and Evaluation (Cont.)

C. Experimental Analysis (Cont.)

1) Impact of the Block Generation: Block generation interval is regarded as an important metric for measuring the performance of blockchain networks.

- They have an organized topology and structure, the block generation will be more efficient to support nodes than in a distributed network with a solid and organized structure.

- Further, since blocks are generated more frequently in individual clusters instead of generated in batches that consume a considerable amount of bandwidth, they can better manage and use the bandwidth.

- Fig. 3(a) shows the bandwidth efficiency of the cluster-based design (proposed architecture) compared to the noncluster-based design.

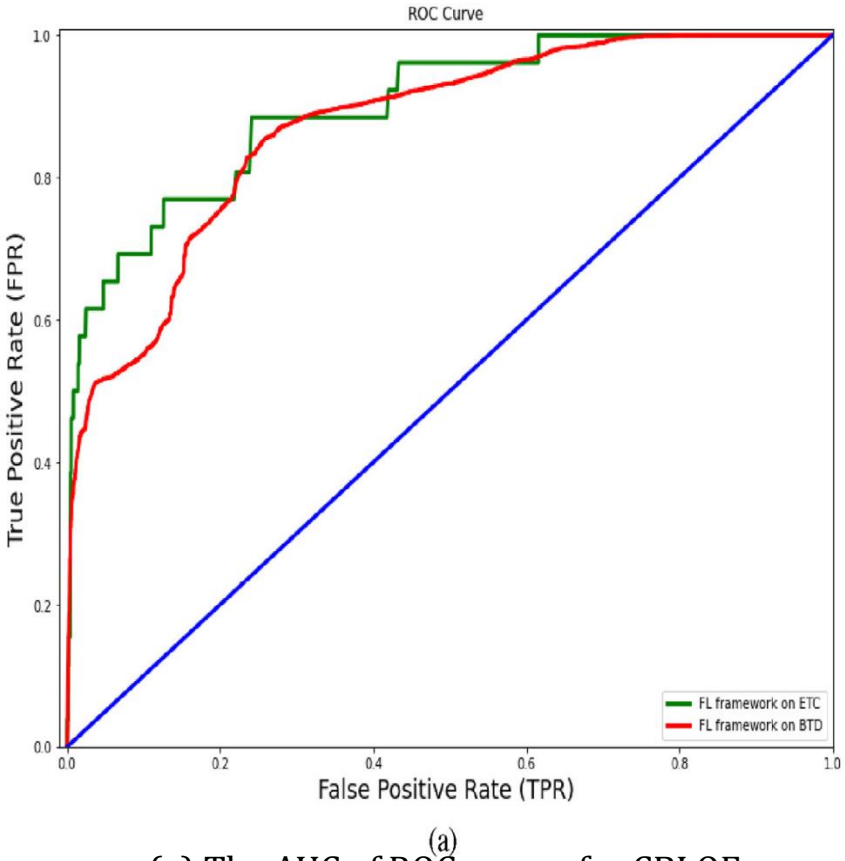
2) Impact of the Block size: Block size has a significant impact on the performance of blockchain. The block size determines the highest number of transactions that can be approved within a block.

- This size, thus, controls the throughput (transactions/ second) obtained by the proposed design. Larger blocks cause more sluggish propagation in each cluster than smaller blocks.

- Fig. 3(b) and (c) shows that the bandwidth consumption and throughput increase with the increasing block size from 0.5 to 8MB. This directly impacts both the bandwidth and throughput of the proposed model.

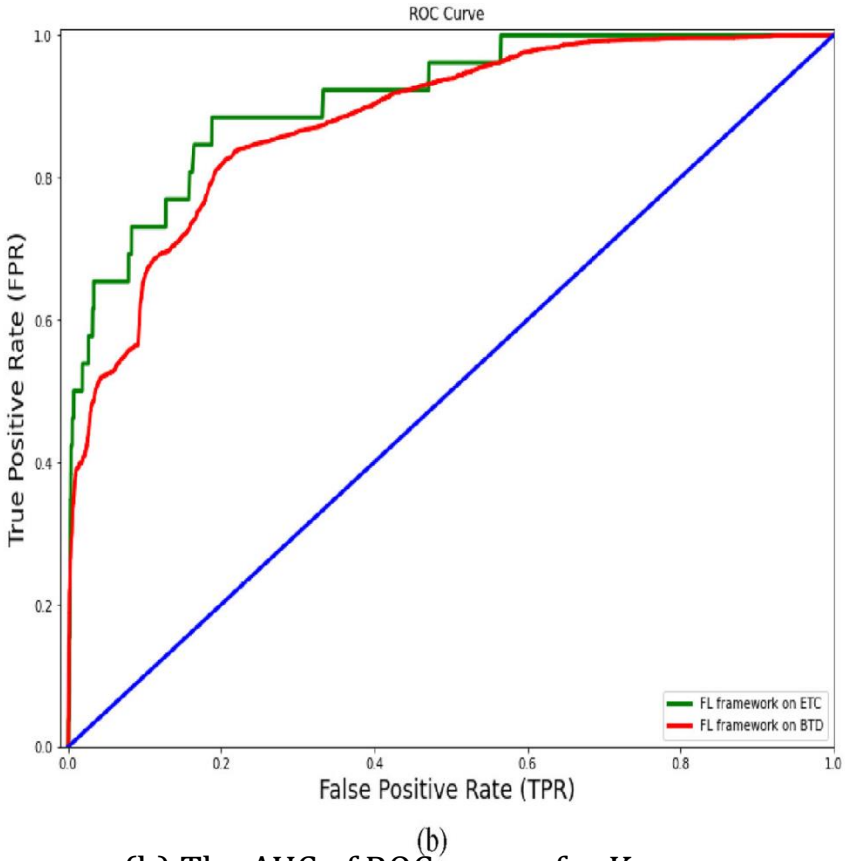
V. Discuss and Evaluation (Cont.)

C. Experimental Analysis (Cont.)



(a)

(a) The AUC of ROC curves for CBLOF.



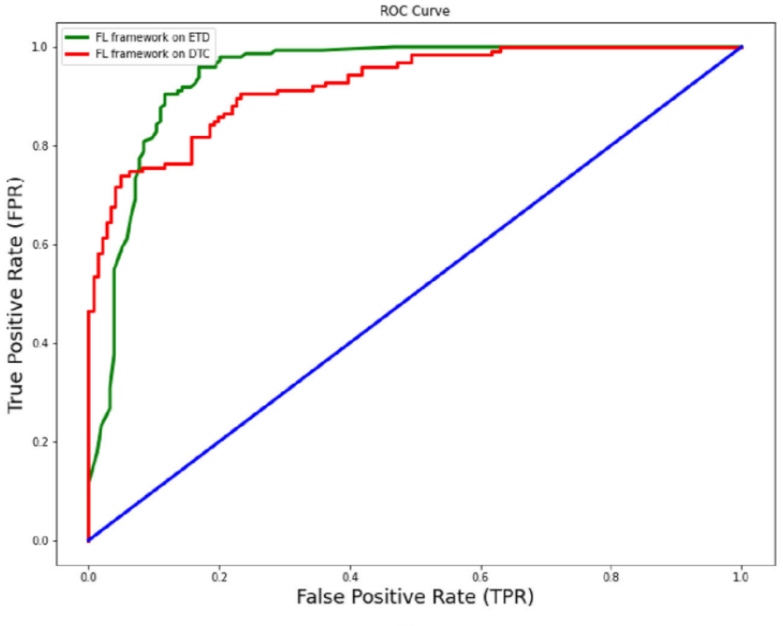
(b)

(b) The AUC of ROC curves for K-means.

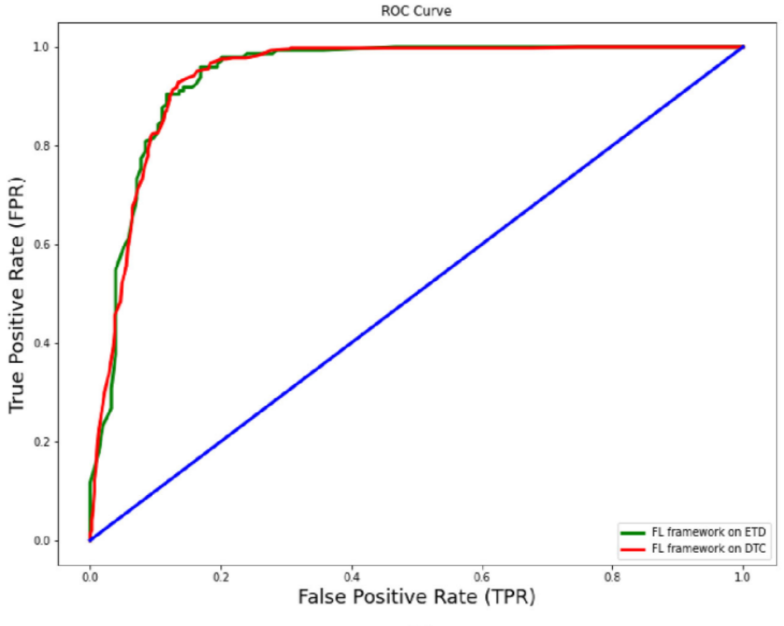
Fig. 4. AUC of ROC curves in FL framework for the Block Hunter.

V. Discuss and Evaluation (Cont.)

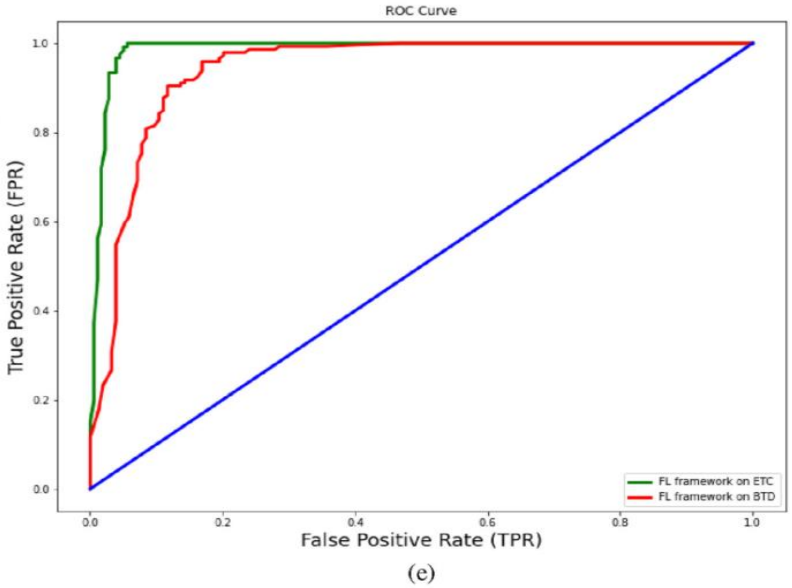
C. Experimental Analysis (Cont.)



(c) The AUC of ROC curves for PCA.



(d) The AUC of ROC curves for IF.



(e) The AUC of ROC curves for NED.

V. Discuss and Evaluation (Cont.)

C. Experimental Analysis (Cont.)

3) Impact of Number of Miners: The number of miners in a given architecture directly impacts throughput (transactions/ second).

- According to Fig. 3(d), an increasing number of miners from 16 to 256 and the block size to 1 MB in all clusters increased the model throughput. The increase in the number of miners makes it easier for smart factories to reach a consensus.

- Additionally, the proposed cluster-based architecture can handle more transactions in each block by increasing the block size. Consequently, it will grow the proposed architecture's throughput rate and offer a better performance.

4) Nomaly Detection Rate: This subsection aims to assess some well-known ML models such as K-means, PCA, CBLOF, IF, and NED to hunt anomalies in the Block Hunter framework.

- They evaluate these models by comparing their average performance, such as Accuracy, Precision, Recall, and F1-score as follows. These include,

$$\text{Accuracy (Acc)} = \frac{TP + TN}{TP + TN + FN + FP},$$

$$\text{Recall (Rec)} = \frac{TP}{TP + FP}, \text{Precision (Pre)} = \frac{TP}{TP + FP}$$

and

$$\text{F1-score (F1)} = \frac{2 * TP}{2 * TP + FN + FP}.$$

V. Discuss and Evaluation (Cont.)

C. Experimental Analysis (Cont.)

TABLE II
PERFORMANCE COMPARISON OF THE ML MODELS BASED ON BITCOIN TRANSACTION DATASET (BTD)

Model	Acc (%)	Pre (%)	F1 (%)	Rec (%)
NED	96.7%	70.4%	62.5%	80.2%
CBLOF	82.4%	55.5%	78%	66.1%
K-means	87.6%	55.2%	65.2%	89.5%
PCA	89.4%	64.2%	76.2%	76.2%
IF	95.3%	65.1%	61.2%	75%

TABLE III
PERFORMANCE COMPARISON OF THE ML MODELS BASED ON ETHEREUM CLASSIC BLOCKCHAIN BLOCKCHAIN (ETC)

Model	Acc (%)	Pre (%)	F1 (%)	Rec (%)
NED	97.8%	74%	66.2%	86.2%
CBLOF	85.6%	60.2%	82.1%	72.1%
K-means	89.7%	59.1%	70.2%	93.1%
PCA	91.6%	70.3%	81.3%	82.1%
IF	96.8%	70.5%	67.6%	81.1v

V. Discuss and Evaluation (Cont.)

C. Experimental Analysis (Cont.)

- Tables II and III display the measured performance of the Block Hunter during applying ML models, K-means, PCA, CBLOF, IF, and NED in terms of Precision, Accuracy, F1-score, and Recall based on BTD and ETC Blockchain datasets. To minimize the loss function, all hyperparameters are maximized.

- They can see that NED and IF have the highest accuracy during the anomaly detection while their accuracy is almost similar.

- By examining the visuals and using the highest level of accuracy metric, the AUC for ROC curves show a comparable ROC curve for all algorithms.

- The AUC for CBLOF, K-means, PCA, IF, and NED are based on BTD and ETC datasets, respectively.

- While running the Block Hunter framework with each ML model, they obtained a global model whose parameters are frequently updated via the FedAvg approach [24].

V. Discuss and Evaluation (Cont.)

C. Experimental Analysis (Cont.)

TABLE IV
ANOMALY HUNTING IN THE BLOCK HUNTER

Number of cluster (K)	Existing anomaly	Max number of transactions (per second)	Detected
30	2	35	2
40	4	35	4
50	4	35	4
30	5	70	4
40	4	70	3
50	4	70	4

TABLE V
SUMMARY OF PERFORMANCE COMPARISON OF THE BLOCK HUNTER IN IIOT RELATED DATASETS

Datasets	Model	Acc (%)	Pre (%)	F1 (%)	Rec (%)
GP	NED	99.1%	98%	99%	98%
	CBLOF	87.8%	82%	78%	70%
	K -means	89.5%	88%	90%	87%
	PCA	95.2%	90%	88%	89%
	IF	96.8%	93%	94%	91%
SWaT	NED	98.8%	97%	99%	99%
	CBLOF	85.9%	77%	78%	75%
	K -means	87.1%	78%	80%	76%
	PCA	92.1%	89%	88%	90%
	IF	94.6%	95%	90%	92%

V. Discuss and Evaluation

C. Experimental Analysis (Cont.)

- Table IV presented the hunting of anomalies in global models using NED as the local model. This table shows the moment where the Block Hunter framework can hunt an anomaly while doing transactions.
- This consists of K = 30, 40, 50 clusters and 1 to 35 transactions per second for 100 s. Based on the cluster-based structure in the Block Hunter, it is almost certain that this system's accuracy is acceptable during anomaly hunting.
- The Block Hunter framework also works perfectly as the number of transactions and clusters increases. They also evaluated the performance of the Block hunter on several IIoT standard datasets as shown in Table V.
- The model performance was evaluated using different ML models namely K-means, PCA, CBLOF, IF, and NED on GP and SWaT datasets. NED has the highest accuracy as it preserves data encoding/decoding.
- Blockchain-based IIoT networks are the underlying technology for the future smart factories, hence, an emerging attack target, which shows the significance of this work.

VI. Conclusion

- In this article, they developed the Block Hunter framework to hunt anomalies in blockchain-based IIoT smart factories using an FL approach.
- The Block Hunter used a cluster-based architecture to reduce resources and improve the throughput of blockchain-based IIoT networks hunting. The framework was evaluated using a variety of ML algorithms (NED, IF, CBLOF, K-means, and PCA) to detect anomalies.
- They also examined the impacts of block generation interval, block size, and different miners on the performance of the Block Hunter.
- Using generative adversarial networks to design and implement a block hunter-like framework would be an interesting future research work.
- Furthermore, designing and applying IIoT-related blockchain networks with different consensus algorithms would also be worth investigating in the future.

Thank You

For your Attention!

연락처: yotxaysangthong@seoultech.ac.kr
+82 10-8999-3151