Journal of Parallel and Distributed Computing 164 (2022) 55–68

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

www.elsevier.com/locate/jpdc

A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network

Randhir Kumar [a], Prabhat Kumar [a,*], Rakesh Tripathi [a], Govind P. Gupta [a], Sahil Garg [b,*], Mohammad Mehedi Hassan [c]

[a] Department of Information Technology, National Institute of Technology, CG, Raipur-492010, India
[b] Department of Electrical Engineering, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada
[c] Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

ARTICLE INFO

Article history:
Received 10 July 2021
Received in revised form 24 November 2021
Accepted 30 January 2022
Available online 18 February 2022

Keywords:
Blockchain
DDoS attacks
Fog computing
Internet of things (IoT)
Intrusion detection system
Mining pool

ABSTRACT

The Internet of Things (IoT) is emerging as a new technology for the development of various critical applications. However, these applications are still working on centralized storage architecture and have various key challenges like privacy, security, and single point of failure. Recently, the blockchain technology has emerged as a backbone for the IoT-based application development. The blockchain can be leveraged to solve privacy, security, and single point of failure (third-part dependency) issues of IoT applications. The integration of blockchain with IoT can benefit both individual and society. However, 2017 Distributed Denial of Service (DDoS) attack on mining pool exposed the critical fault-lines among blockchain-enabled IoT network. Moreover, this application generates huge amount of data. Machine Learning (ML) gives complete autonomy in big data analysis, capabilities of decision making and therefore is used as an analytical tool. Thus, in order to address above challenges, this paper proposes a novel distributed Intrusion Detection System (IDS) using fog computing to detect DDoS attacks against mining pool in blockchain-enabled IoT Network. The performance is evaluated by training Random Forest (RF) and an optimized gradient tree boosting system (XGBoost) on distributed fog nodes. The proposed model effectiveness is assessed using an actual IoT-based dataset i.e., BoT-IoT, which includes most recent attacks found in blockchain-enabled IoT network. The results indicate, for binary attack-detection XGBoost outperforms whereas for multi-attack detection Random Forest outperforms. Overall on distributed fog nodes RF takes less time for training and testing compared to XGBoost.

1. Introduction

Internet of Things (IoT) has emerged as a new technology that has merged with our daily lives as the Internet has progressed. The IoT-based applications such as supply chain management, healthcare, RFID based identity management system is directly empowering the individual and society [14]. The underlying technology is becoming promising for data analysis, and modeling by combining cloud computing and machine learning [34]. The advancement in the IoT-based development is causing growth in various sectors. However, the application built with IoT system mostly works on centralized storage and computing architecture [22,6]. The centralized storage model lacks various security and privacy breaches. The underlying working model has constraints to facilitate the expansion of IoT based system in near future [37]. Hence, there is a need of decentralized or distributed storage model that can address these issues. One of the emerging decentralize-based architecture is blockchain technology [24].

The blockchain is decentralized and immutable storage model that consists of all transactions details that have been initiated by the peer node in the network. The concept of decentralized storage is called distributed ledger [33]. Any transaction that is processed in the ledger is verified by the consent of the majority of network participants. Bitcoin is the most popular real time implementation of blockchain technology [21]. Blockchain and IoT integration can provide many benefits such as, decentralized blockchain storage model have the ability to synchronize the IoT devices and can provide real-time data to each IoT nodes. This underlying integrated model eliminates the third-party dependency and single point of failure [15]. In addition, IoT integration with blockchain can enable peer-to-peer messaging, real-time data and file sharing, and au-

* Corresponding authors.
E-mail addresses: rkumar.phd2018.it@nitrr.ac.in (R. Kumar), pkumar.phd2019.it@nitrr.ac.in (P. Kumar), rtripathi.it@nitrr.ac.in (R. Tripathi), gpgupta.it@nitrr.ac.in (G.P. Gupta), sahil.garg@ieee.org (S. Garg), mmhassan@ksu.edu.sa (M.M. Hassan).

https://doi.org/10.1016/j.jpdc.2022.01.030
0743-7315/

tonomous communication between IoT nodes without the need of a centralized client-server model [16].

Although blockchain is verifiable and immutable, yet it is vulnerable to different attacks [12]. The IoT and blockchain integration has experienced massive growth in revolutionizing the stand-alone IoT applications [27]. However, the number of attacks has also increased accordingly. DDoS attacks often caused by flooding on mempool/memory pool in blockchain network has severe consequences to legitimate users [20], [30].

In existing blockchain networks, DDoS attacks are aimed mostly on miners (mempool), users and their communication medium [29]. In the peer-to-peer system DDoS may be performed in different form like bootstrapping of blockchain network, users and miners towards a fake or counterfeit networks by denying of access to real network. This process can be performed by hacker by hijacking few (<100) Border Gateway Protocol (BGP) prefixes [2]. Another way to target the DDoS attack on the blockchain network is flood attack on memory pool (mempool) with spam transactions by the attacker in the network. In blockchain, the mempool work as a transaction repository where all shared transactions by a peers are initially logged and waits for the confirmations [17]. Once IoT node generates the transactions, it gets disseminated among all the synchronized IoT peer nodes. The raised transactions waits for the confirmation in the mempool. In November 11, 2017, the size of mempool exceeded by 115K spam transactions that causes loss of 700 million USD [18,23]. As the mempool size grows with unconfirmed transactions then real users have to pay further more mining fees to prioritize his/her unconfirmed transactions and this situation becomes opportunities for the attackers [41]. Thus in order to detect DDoS attacks, a secure and robust security mechanism is required.

On the other hand data generated by these applications are huge, causing big data related issue. Thus, to address this issue Artificial Intelligence (AI) works as an analytical tool and provides useful information to decision making, classification, prediction and detection of future actions in blockchain-enabled IoT network [25], [28]. Moreover, existing blockchain-based architecture uses cloud server for analysis. However, centralized server has various constraints such as low latency, less computational storage, low accuracy, low speed [19]. In order to fulfill these demands a new distributed paradigm, fog computing should get thoroughly investigated.

Fog computing is a decentralized architecture coined by Cisco in 2012, is an extension of cloud to the network edge [31]. The principle of fog computing such as support for heterogeneity, low latency, location awareness, geo-distribution, support for mobility presents a wide range support to blockchain-based applications [26]. Fog computing can also be used for load balancing, data collection in distributed manner [9]. Thus this paper integrates fog computing with AI to design a distributed security mechanism, that can detect DDoS attacks against mining pools in blockchain-enabled IoT network. Fig. 1 shows, secure detection model with blockchain, IoT, AI, and fog computing integration.

IDS is a tool that combines software and hardware to monitor network traffic or systems to identify malicious activities [3]. According to detection techniques IDS can be broadly categorized into two types. Signature-based IDS and anomaly-based IDS. In SIDS, a database of predefined rules or patterns with existing attacks are kept and the new traffic is matched against this pattern. When a suspicious activity is detected, administrator is alerted. On the other hand in AIDS, a typical normal user behavior is modeled. Any deviation between normal and incoming traffic is treated as malicious (intrusion) [10].

In the context of blockchain-enabled IoT network, SIDS gets confined as IoT systems are heterogeneous and diverse in nature [39]. In this paper, the proposed detection system is anomaly-based. Further, the effectiveness of the proposed model is evaluated by a benchmark dataset i.e., BoT-IoT [13]. This dataset is generated in a realistic IoT environment using a lightweight Message Queuing Telemetry Transport (MQTT) protocol and contains recent Botnet related attacks such as Dos, DDoS, Theft. Thus, BoT-IoT dataset is best suited for this research.

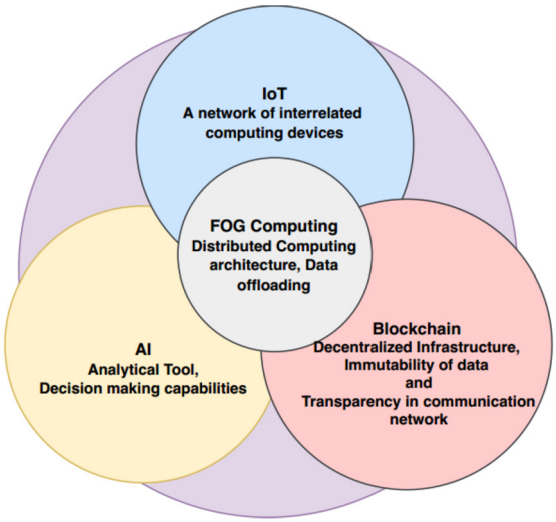


Fig. 1. Secure detection model with the integration og Blockchain, IoT, AI, and Fog computing.

In order to detect DDoS attacks, this paper proposes a fog computing based distributed IDS for mining pool in blockchain-enabled IoT network. The proposed IDS is trained using RF and XGBoost algorithm. Fig. 2, shows the overview of the proposed distributed model. The detection system consists of sensing nodes that are responsible to identify moving objects within their vicinity. All IoT sensors are identical and have same detection radius r_d based on which they are clustered into different groups. The data generated by this cluster group is send to local fog nodes. As fog node provides services such as gateway, access point to IoT devices, thus the security mechanism, IDS is integrated with every fog node. The incoming traffic is evaluated by IDS and respective measures are taken accordingly. If the incoming traffic (transactions) is normal then the transaction gets disseminated in the memory pool (mem pool) for the mining. The miners select the transactions for mining and block creation is done in blockchain network situated at cloud. If transactions are malicious or invalid then the IDS generates the alarm for administrator to take necessary actions.

1.1. Motivation

According to the literature, numerous security problems and challenges occur in the mining pool of blockchain-enabled IoT network. The growing DDoS attack in the blockchain-IoT ecosystem renders all blockchain-enabled IoT network vulnerable. The key challenges are listed below:

- Ensuring distributed security framework for blockchain-based IoT network is a challenging task.
- Ensuring a security mechanism uses appropriate analytical tool in a distributed working architecture and is capable of handling huge data generated by IoT devices in a distributed manner.
- Building an effective IDS that can differentiate normal and attack transactions is challenging task. There is not much research visible for security mechanism of mitigating DDoS attack against mining pools in blockchain-enabled IoT network after model deployment.

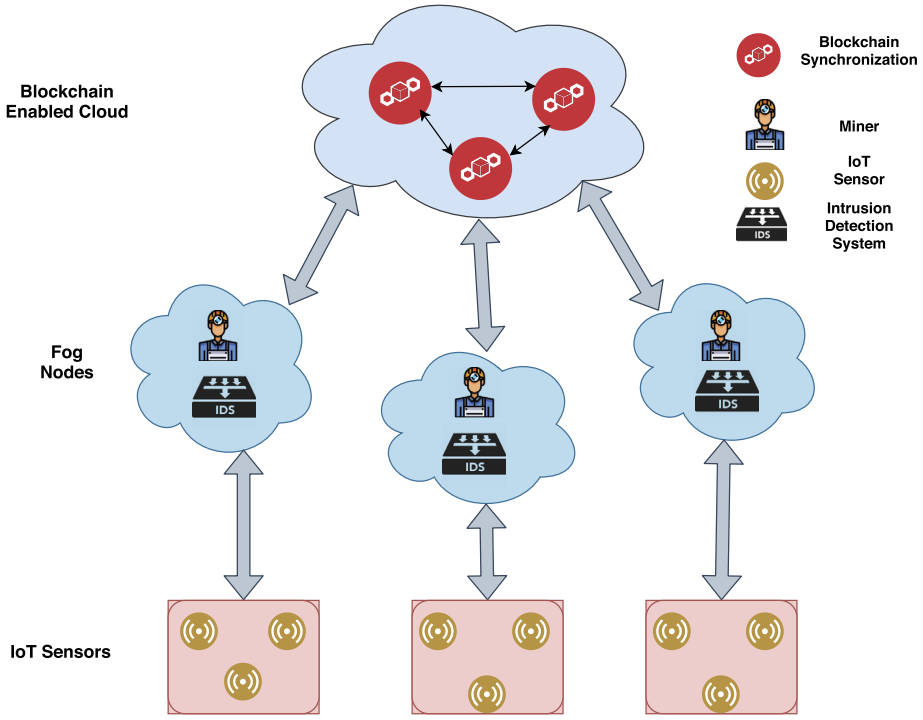


Fig. 2. Proposed IDS framework for blockchain based IoT Network.

1.2. Contribution

To address the aforementioned challenges and issues, this paper aims to construct a distributed IDS based on AI-enabled fog computing with integration of mining-pool to detect DDoS attack with blockchain-enabled IoT network. We are the first to design a framework for the deployment of a anomaly-based IDS for mining pool in blockchain-IoT environment. The main contributions of our research are as follows:

- The devices and sensors in blockchain-enabled IoT network produce a lot of data, so AI is applied as analytical tool to provide consistent results in decision-making.
- Fog computing paradigm is used to decentralize cloud based centralized security mechanism, thus data analysis and security related concerns are handled at edge of networks.
- A distributed IDS is designed using fog computing to detect DDoS attacks against memory pool in blockchain-enabled IoT network.
- To evaluate the proposed detection system two well known machine learning algorithms, random forest and XGBoost are used in distributed architecture.
- An actual IoT based BoT-IoT dataset is used to analyze the performance of the model. As it contains various recent Botnet related attacks such as DoS, DDoS, theft.
- Different evaluation metrics such as accuracy, detection rate, false alarm rate, and precision are used to thoroughly investigate performance of the proposed IDS.

2. Related work

In this section, we review the various attacks on blockchain based network including mining attack, transaction attack, and DDoS attack. The DDoS attack is centerpiece of our study. The work in [7,38] discusses about the selfish mining problem, where miners do not publish their block after mining computation, hoping to get more reward by mining subsequent blocks. Eyal et al. [7] proposed strategy to prevent selfish mining problem in blockchain network. Rosenfeld et al. [35] explored the withholding attack, where miners submit partial proof-of-work rather than full proof-of-work, as a result miners gets rewarded for the mining participation although mining pool suffers from partial solutions. Similarly Yujin et al. [32] introduces new attack called fork attack after withholding attack. In this attack rewards are always higher than the withholding attacks.

The 51% attack can be initiated, if an attacker gains more than 50% of the network hashing power. With more than 50% of the hashing power attacker can prevent from transactions verifications and mining a block in the blockchain network. To address this attack, Eyal et al. [8] proposed the Two-Phase Proof-of-Work (2P-PoW) and examined by Bastiaan et al. [5]. The transaction attack in blockchain includes double spending attack, when a peer nodes generates two transactions and disseminate with two recipient [11]. To address this problem one-time signatures technique is approached [36].

The Distributed Denial-of-Service (DDoS) have been quite prevailing attack in blockchain network [40]. The information exchange between two peers in the blockchain network has been frequently targeted by attackers reported in the various studies [25]. However, none of these studies have given the mitigation procedures for proposed attacks. Another form of distributed denial of service attack on blockchain network includes generating low price transactions in the mempool [27]. This attack is known as penny-flooding attack. Kumar et al. [15] discusses the blockchain stress by analyzing the blockchain networks and how the respective attackers can exploit them.

Kumar et al. [20] designed an IDS for smart contract-based blockchain IoT system. Mothukuri et al. [34] proposed a federated learning-based attack detection in IoT environment. This approach was implemented using PySyft library. However, important evaluation metrics such as false alarm rate was not used in this work. Bakhsh et al. [4] suggested an adaptive IDS for IoT devices that uses agent technology to support portability, rigidity, and self-starting characteristics. This was a hybrid system that used both

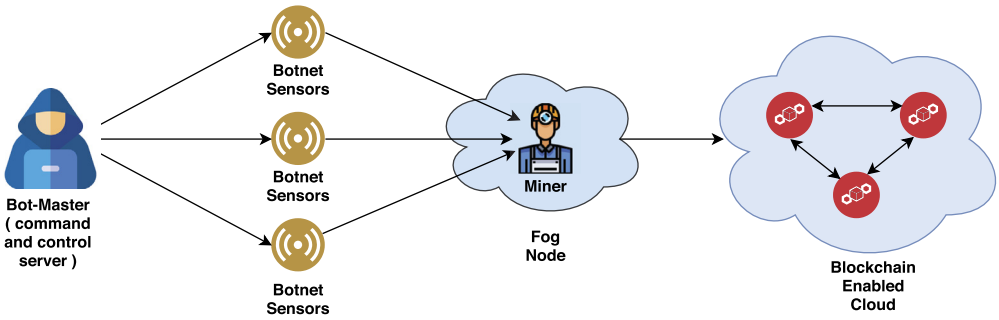


Fig. 3. The scenario of DDoS attacks by miners in a blockchain-enabled IoT network.

host-based and network-based capability to identify both misuse and anomalies. However, there were no performance measurements for the IDS or its running platforms available in this study. Anthi et al. [1] suggested a network-based IDS system that is predictive and adaptable for IoT ecosystems based on signature and anomaly-based detection. However, the model specifics were unclear in the study, which yielded moderate attack detection outcomes.

From the literature we conclude the expected threat model and attack objective in blockchain-enabled IoT network.

2.1. Threat model

In this work, we consider two different notions in terms of attack on blockchain-based IoT network. First, the attacker can generate large volume of transactions by Bot-IoT devices with spending minimum mining fee. Second, the attacker can generate group of Sybil accounts which consists of multiple public addresses. Both the attackers and Sybil account works mutually in the network with the knowledge of their respective public addresses. The attackers and Sybil accounts execute client side scripting, which facilitate them to generate flood of raw transactions by Bot-IoT devices in short period of time.

2.2. Attack objective

While generating the flood of raw transactions in mempool, the attacker objective is to maximize the size of mempool and reduce the cost of the attack. The attack cost is the fee paid to the miners which includes relaying and mining fees. If the transactions get mined with minimal fee then by making higher fees of transactions increases priority and the chances of transactions to be mined in the blockchain network. To produce flood attack in the mempool, the attackers initiate low fee transactions that are less likely to be prioritized and stay in the mempool as long as possible.

In this article we analyze and mempool flooding attack by the different counters measures. Moreover, we explored the mempool flooding attack in terms of blockchain-enabled IoT network. We also discussed that how to detect IoT devices that has became BoT devices and generating spam transactions with the supervision of attackers. To identify the flooding attack, we deployed the IDS on mempool. The IDS is completely tested and trained on the real time IoT datasets namely BoT-IoT. We are the first to design the AI enabled IDS in mempool for prevention of DDoS attack.

2.3. DDoS attack scenario on miners in a blockchain-enabled IoT network

The miners in blockchain-enabled IoT network are responsible to verify the transaction by validating the signatures contained within the transaction. Miner appends the verified transaction to a blockchain network. In a blockchain network robustness is maintained by having multiple miners processing a single transaction. However, mining same transaction by multiple miners increases delay in the network. For example, to perform DDoS attacks with in smart home application, a bot-master can perform malicious activities in the entire system such as verification of identity, data storage model between smart thermostat with cloud and local fog nodes.

Fig. 3 shows the DDoS attack scenario on miners with blockchain based IoT applications. The bot-master(controls and command the server) injects malicious code within IoT sensors through Internet, making them botnet sensors. The malicious activities aim to interrupt the normal miners working and further it leads to fog nodes utilization for unusual resources and activities. As a result, miner stuck with large numbers of illegal transaction requests. Therefore, to reduce the DDoS attacks against mining pool in blockchain-IoT application, this paper proposes a robust and effective distributed IDS that integrates AI and fog computing.

3. Our proposed model

This section discusses an IDS integration with mining pool along with working of distributed IDS, to detect the DDoS attack on AI-enabled fog computing. The steps describe the preprocessing of data, and also the detailed about AI-enabled ML techniques for blockchain-based IoT networks deployments.

3.1. IDS integration at mining pool in IoT environment

Mining pool is combined with the IDS to detect suspicious transactions at blockchain-based IoT systems. Fig. 4, demonstrates an IDS integration with the mining pool. Our aim is to protect miners in mining pool from the DDoS attacks, once they are successfully deployed with IoT ecosystems. The incorporation of the IDS within mining pool works as a final defense line. Incoming traffic is analyzed by the IDS in the proposed method. If packet arrives normal, the transaction packets are mined by miners and appended to chain of the blockchain network. In comparison, if the new traffic or transactions contains unusual behavior, administrator gets alerted and permitted to perform necessary action. In addition, for generating false alarm the proposed detection system, administrator has the authority to forward the transaction in the mining pool, where the miner mines it, and adds the transaction to the blockchain network. As a consequence, the IDS will alerts administrators a second chance to take action and target their adversary.

3.2. Working of distributed IDS in blockchain-enabled IoT network by integrating ML and fog computing for mining pool

The proposed detection system has decentralized the current centralized security and data storage close to the network edge.

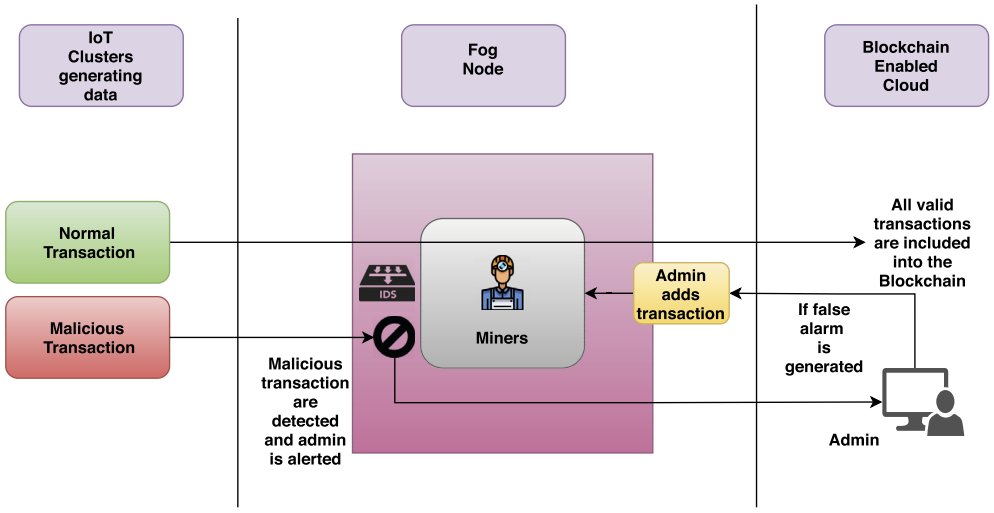


Fig. 4. Proposed IDS integration with mining pool.

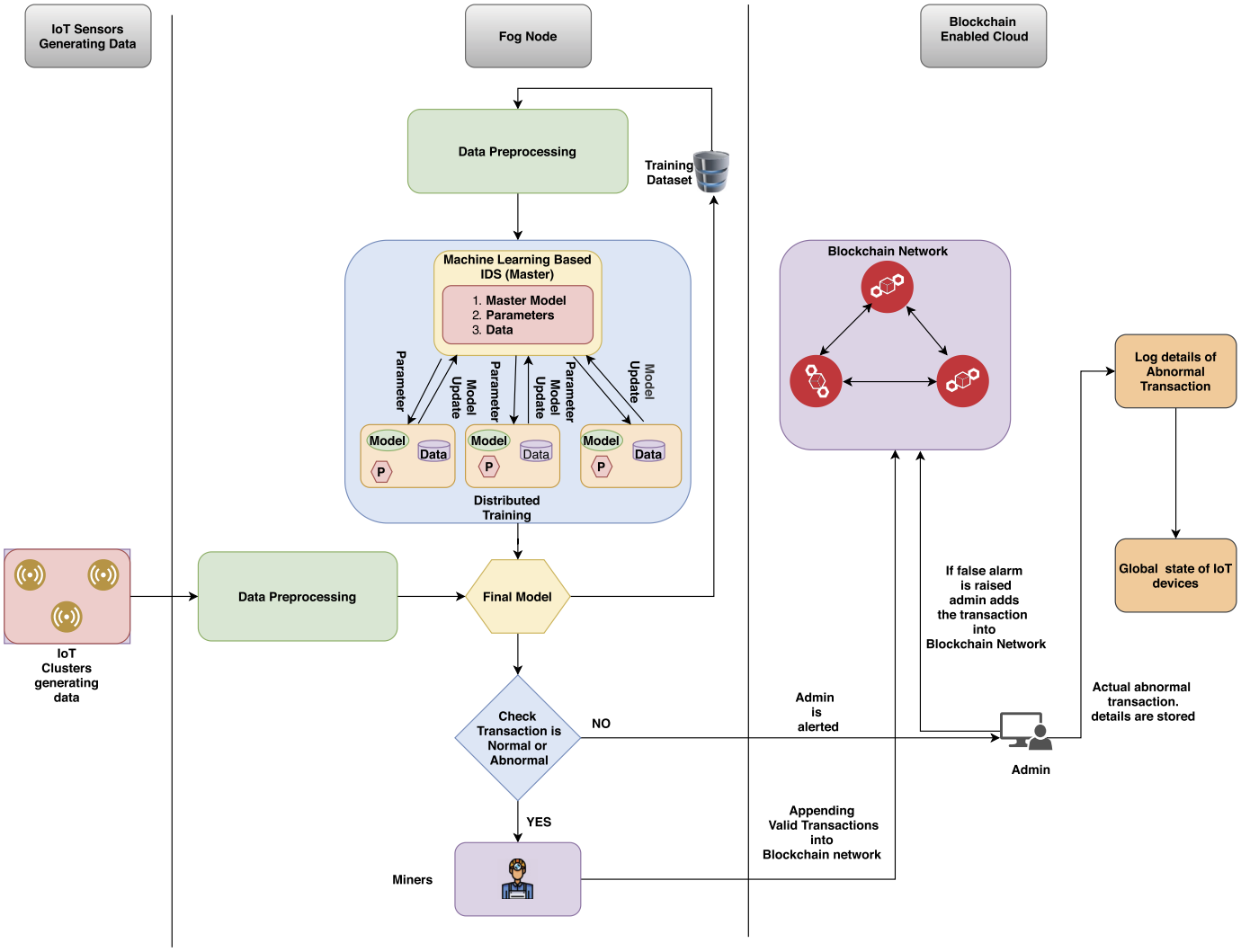


Fig. 5. Proposed distributed IDS working structure model for mining pool in blockchain-enabled IoT network using fog computing.

Fig. 5, demonstrates the working of distributed IDS. The proposed detection system consists of three main engines:

(i) Traffic Processing Engine: In this phase, fog nodes are applied to process traffic of network, and to deploy intrusion detec-

tion system at fog network edge, as it gets closer to the systems (IoT devices). This step involves the pre-processing of the training dataset. StandardScaler technique is used to normalize the dataset. Next, AI-based two different machine learning (ML) techniques are applied in the distributed structure i.e., random forest and XGBoost. The master node coordinates with a place for both the strategies for sharing and optimizing mutual parameters. The benefit of this approach is that it allows local attack detection power by local training and parameter optimization by speeding up data training near the source and acquiring modified parameter values from neighbors. The master node gives update to every cooperative (worker) node's with certain parameters and corrects the updated values with worker nodes. It plays an prominent role while the storage offloading as well as computing on costs with IoT models, for quick response of actual data with various parameters.

(ii) Intrusion Detection Engine: A predictive model is created, that is used to evaluate the efficiency of the detection system. IoT sensors in blockchain-enabled IoT network are responsible for generating large volumes of data. This incoming traffic follows pre-processing step and finally it is analyzed against the predictive model.

(iii) Transaction Handling Engine: Depending on the behavior of traffic, transaction gets categorized into normal and malicious type. If the transaction is normal or legitimate, the miner would execute the transaction in the mining pool and add the transaction to the blockchain network stored in the cloud. On the other hand, the administrator is alerted to suspicious transactions and is allowed to take the appropriate intrusion protection steps. Log details of these transactions are submitted to the cloud on the basis of which the global status are maintained successfully of IoT devices. In addition, the extra benefits are offered to the administrator, if the IDS generates a false alarm, an administrator can transfer the transactions to miner for further addition into the blockchain network. The next sub-section discusses data preprocessing steps in detail.

3.3. Preprocessing of data

The IoT network traffic consists of variant magnitude values. Data preprocessing helps in transforming raw data into more suitable form for modeling purpose. This technique assists in reducing the detection system's training and testing time. It leverages the entire detection system's efficiency by identifying attack observations in an IoT environment.

3.3.1. Data normalization

The extent of blockchain-based IoT network traffic varies. This model uses the StandardScaler normalization approach for scaling the feature values. This technique transforms observations of feature such as distribution of incoming traffic with mean value 0 and standard deviation as value 1 [20]. This method incorporates the feasibility of the proposed detection system as it eliminates incoming traffic bias without changing its mathematical properties. The Eq. (1) shows transformation.

$$s_k = \frac{val_k - \mu_k}{\sigma_k} \tag{1}$$

where s_k, denotes features standard score within the detection system, $k \in \{k1, k2, k3.., kn\}$. The val_k denotes the IoT traffic features. μ_k and σ_k denotes mean and standard deviation of features that is evaluated with following expression below:

$$\mu_k = \frac{\sum_{k=1}^{K} val_k}{N} \text{ and } \sigma_k = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(val_k - \mu_k)^2}$$

3.3.2. AI methods deployed for classifying attack instances

The processing of big-data within a blockchain-based IoT systems is a major challenge. The volume of data sent by the various sensor nodes that continues to rise. Sensors are usually limited in terms of length, memory, and processing power. One of the major problems and pitfalls of this rapidly expanding field is data storage and privacy. To monitor and analyze traffic and predict potential attacks, an IoT-based IDS is required. To overcome the aforementioned challenges, the proposed IDS includes AI technologies such as machine learning (ML). To distinguish regular and attack observations, two machine learning methods, Random Forest and XGBoost, are used. From large data sets of training sensors, machine learning techniques can automatically differentiate between natural and malicious patterns. The identification of accurate threats is needed due to the high volume of traffic and the need for real-time response. This section briefly explains why two classification algorithms were selected and how they operate in a blockchain-based IoT infrastructure.

In order to justify why these classification methods are selected in the proposed detection system, IoT network constraints such as capability of generating huge data in distributed and parallel manner, datasets include both numerical and categorical values, and the sparse nature of IoT traffic is a key factor in their selection. This has intensified the need to assess which classification techniques can be used in a distributed detection scheme based on potential procedures for developing techniques that can manage massive, traffic and sparse input with in IoT environment and can be used with distributed and parallel context. Thus, RF and XGBoost gets selected because they can easily distinguish these IoT network traffic. Further, big data and sparse data can be easily managed owing to the supports of parallel and distributed computing. The working of both the techniques are explained below:

1. Random Forest (RF) [27]: RF is a technique of machine learning that employs ensemble bagging to create a big number of unrelated decision trees from a collection of random features. When building decision trees, a random list of u individual characteristics is selected according to split candidates subsets from the entire set of independent attributes when split is used within tree (in parallel). As a result, each split produces a unique set u with new and independent features; typically, $u = sqrtL$ is used, meaning that each split u is roughly equal to the total number of independent features. As a consequence, weakly coupled classifiers are combined to form a strong classifier. Since it is sensitive with outlier, missing values, over-fitting, and has the capacity to handle a huge volume of incoming traffic, random forest is ideal for anomaly detection in blockchain-based IoT architecture. The training and testing for parallel distributed near fog nodes for random trees is managed by algorithms 1 and 2.
2. XGBoost [20]: Extreme Gradient Boosting is an ML Ensemble technique based on boosting. It is a part of boosting algorithm that transforms weak student to strong one. This approach is sequential that expands the tree one after another, with each repetition attempting to minimize the misclassification rate. It is accomplished by higher weight assignments to the previous trees point of misclassification. Further, it is more effectively accommodate different forms of sparsity patterns in input data, as well as regularization terms to prevent overfitting. Regularization is a compensation word that penalizes abstract models while rewarding simpler ones. Least Absolute Shrinkage & Selection Operator(L1) and Ridge Regression are two regularization parameters available in XGBoost (L2). The aim is to create a paradigm that is less complex in order to reduce bias and possible overfitting. In the s^{th} estimation, XG-

Algorithm 1: Random Forest based distributed parallel training.

Input:
f : Parallel system fog nodes
t^r : Random forest (RF) with total number of trees
D_s : D is the datasets of training and s denotes size
A : training datasets independent features
Output:
Tree of RF ($\mathbb{R}$)
if ($t^r > f$) **then**
  $k_t = t^r/f$ //where each node is capable of generating k_t trees
else
  f=1 //t<=f
end
// parallel environment with iteration of each fog nodes
for $j \leftarrow 1$ *to* K **do**
  Generate bootstrap by computing dj of s size for dataset
  Perform Random sampling using substitutions from D_s
  $OOB_j = D_s - d_j$ //calculating the error (out of bag)
  $z = \sqrt{A}$
  z denotes set of attributes from A with initial attribute $\mathbb{A}_{\mathbb{j}} = \{\mathbb{a}_{\mathbb{1}}, \mathbb{a}_{\mathbb{2}}, \mathbb{a}_{\mathbb{3}}, \ldots, \mathbb{a}_{\mathbb{z}}\}$
  D_t=build_decision_tree(d$_j$, $\mathbb{A}_{\mathbb{j}}$)
end

Algorithm 2: Random Forest-based Distributed Parallel Testing.

Input:
f : Fog nodes on Parallel system
t : Random Forest (RF) tree
k_t: Each node consisting number of tree
D_{st} : Dataset (D) with size st
C_l : Dataset D_{st} classes
Output:
distinguish the normal(0) and attack(1) class
for $j \leftarrow 1$ *to* st **do**
  for $i \leftarrow 1$ *to* k_t **do**
    for each record j traavers tree T_i
    local copy j is classified as $\{c_1, c_2, c_3, \ldots, C_{cl}\}$
    // each classified local copy maintains notation of 2D array i.e., [t_i, c_j], the t_i denotes objects and c_j denoted as class. if an instance is obtained successfully within class, 2D array will be filled with 1, else it will be filled with 0. The obtained local copy gets disseminated to every node at every iteration.//
  end
end
Finally, master copy will be updated by obtained result of local copy of indivisual with attack (1) and normal (0)

Boost uses exponential learning methods, which combine the optimal model with the existing classification model.

$$v_k^s = v_k^{(s-1)} + f_s r(k) \tag{2}$$

As shown in Eq. (2), f$sr(k)$ denotes the best tree prediction in the s^{th} prediction $vk^{(s-1)}$ denotes the current classification model, and the next prediction is vk^s new classification model.

$$W^s = \sum_{k=1}^{N} L_{XGBoost}(v_k, p_k^s) + \sum_{k}^{s} \Psi(f_k) \tag{3}$$

To measure the loss value in XGBoost algorithm, we use XGBoost $LXGBoost(vk, vk^s)$. The $\Psi(fk)$ is the regularization that avoids overfitting. From the Eq. (2) and Eq. (3), we obtained again a form of XGBoost algorithm objective functions as

$$W^s = \sum_{k=1}^{N} L_{XGBoost}(v_k, v_k^{s-1} + f_s r(k)) + \Theta(f_s) + C \tag{4}$$

As shown in Eq. (4), the $\Theta(fs)$ is a term of regularization which states complexity fs tree and constant term C. The expansion of Taylor series can be seen in Eq. (5)

$$f(r + \Delta r) \cong f(r) + f'r\Delta r + \frac{1}{2} f''r\Delta r^2 \tag{5}$$

Next, objective function is represented by Taylor expansion that is shown in Eq. (6)

$$\begin{aligned} W^s &\cong \sum_{k=1}^{N} [L_{XGBoost}(v_k, v_k^{s-1}) + a_i f_s r(k) \\ &\quad + \frac{1}{2} b_i f_s^2(r_k)] + \Omega(f_s) + C \\ s.t\ \ a_k &= \partial v_k^{(s-1)} L_{XGBoost}(v_k, v_k^{(s-1)}), \\ b_i &= \partial^2 v_k^{(s-1)} L_{XGBoost}(v_k, v_k^{(s-1)}) \end{aligned} \tag{6}$$

The loss function is defined by second level derivatives using b_k. The process of scoring computation is evaluated using XGBoost functions a_k and b_k. The new representation of the objective function is evaluated by removal C (constant term). Further, Eq. (7) shows objective function.

$$\begin{aligned} &\sum_{k=1}^{N} [a_k f_s r(k) + \frac{1}{2} b_k f_s^2(r_k)] + \Theta(f_s) \\ s.t\ \ a_k &= \partial v_k^{(s-1)} L_{XGBoost}(v_k, v_k^{(s-1)}), \\ b_k &= \partial^2 v_k^{(s-1)} L_{XGBoost}(v_k, v_k^{(s-1)}) \end{aligned} \tag{7}$$

Each tree is redefined as $f_s(r) = \theta_{q(r)}$, where $\theta \in \Re^S$, q:$\Re^d \rightarrow$ (1,2,3, ..., S). The term q(r) indicates the leaf node. The $\theta q(r)$ defines score of leaf nodes and also the value prediction of current model. The $\Theta(f_s)$ is denoted as

$$\Theta(f_s) = \beta S + \frac{1}{2} \gamma \sum_{k=1}^{N} \theta_l^2 \tag{8}$$

The Eq. (8) is divided in two parts, first part (L1) denotes the selection of leaf nodes and the second part (L2) evaluates the normalization value to avoid the overfitting. Each node of leaf nodes is further defined as $I_l = k \mid q(r_k) = l$. Thus, new objective function is shown in Eq. (9).

$$\begin{aligned} W^s &\cong \sum_{k=1}^{N} [a_k f_s r(k) + \frac{1}{2} b_i f_s^2(r_k)] + \Theta(f_s) \\ &= \sum_{k=1}^{N} [a_k \theta_{q(r_k)} + \frac{1}{2} b_k \theta_{q(r_k)}^2] + \beta S + \frac{1}{2} \beta \sum_{l=1}^{S} \theta_l^2 \\ &= \sum_{l=1}^{S} \left[(\sum_{k \in I_l} a_k) \theta_l + \frac{1}{2} (\sum_{k \in I_l} b_k + \gamma) \theta_l^2 \right] + \beta S \end{aligned} \tag{9}$$

The term θ_l^* is used for optimal value computation of XGBoost algorithm

$$\theta_l^* = -\frac{\sum_{k \in I_l} a_k}{\sum_{k \in I_l} b_k + \gamma} \tag{10}$$

Further, XGBoost objective function is represented as Eq. (10) and Eq. (11)

$$W^{(s)} = -\frac{1}{2}\sum_{l=1}^{S}\frac{(\sum_{k\in I_l} a_k)^2}{\sum_{k\in I_l} b_k + \gamma} + \beta S \tag{11}$$

we set $A_l = \sum_{k\in I_l} a_k$ and $B_l = \sum_{k\in I_l} b_k$, and the objective function can be rewritten as shown in Eq. (12)

$$W^{(s)} = \sum_{k=1}^{S}\left[A_l\theta_l + \frac{1}{2}(B_l+\gamma) + \theta_l^2\right] + \beta S \tag{12}$$

We set $\theta_l^* = -\frac{A_l}{B_l+\gamma}$, and the objective is rewritten as mentioned in Eq. (13)

$$W^{(s)} = -\frac{1}{2}\sum_{l=1}^{S}\frac{A_l^2}{B_l+\gamma} + \beta S \tag{13}$$

The greedy algorithm is applied to find split in XGBoost algorithm. The value of $\epsilon = 0.1$ is set to the parameters, to avoid the overfitting. This predefined parameter value makes degradation in value of prediction. The final model is represented as:

$$v_k^{(s)} = v_k^{(s-1)} + \epsilon f_s(r_k) \tag{14}$$

Both the Algorithm 3 and Algorithm 4 show, parallel training and testing approach in local fog nodes. The working of proposed model of detection is discussed in next section.

Algorithm 3: XGBoost-based Parallel training in distributed model.

Input:
$\mathbb{f}$: Parallel system fog nodes
t_r : Number of trees produced in XGBoost
D_s : Dataset (D) and the training sample size s
Output:
Generated Tree in XGBoost
// iteration evaluated at each parallel nodes
for $i \leftarrow 1$ *to* $\mathbb{f}$ **do**
  if $(t_r > \mathbb{f})$ **then**
    initiate with s^{th} tree $f_s(r_k)$
    Compute $a_k = \partial_{v_k^{(s-1)}} L_{XGBoost}(v_k, v_k^{(s-1)})$
    Compute $b_k = \partial^2_{v_k^{(s-1)}} L_{XGBoost}(v_k, v_k^{(s-1)})$
    New tree $(f_s(r_k))$ formation using statistics to greedy grow
    $W^{(s)} = -\frac{1}{2}\sum_{l=1}^{S}\frac{A_l^2}{B_l+\gamma} + \beta S$
    As shown in Eq. (14), add the best tree $f_s(r_i)$ to the present model
    The dataset with size s is distributed by round robin where master node uses dataset firstly and the first worker nodes used in secondly, and so on. The identical dataset is used at individual fog nodes.
  else
    f=1 //tr <= f
    initialize the s^{th} tree $f_s(r_k)$
    Compute $a_k = \partial_{v_k^{(s-1)}} L_{XGBoost}(v_k, v_k^{(s-1)})$
    Compute $b_k = \partial^2_{v_k^{(s-1)}} L_{XGBoost}(v_k, v_k^{(s-1)})$
    Apply the statistics with greedy grow approach for a new tree $f_s(r_k)$:
    $W^{(s)} = -\frac{1}{2}\sum_{l=1}^{S}\frac{A_l^2}{B_l+\gamma} + \beta S$
    The Eq. (14) shows the best tree $f_s(r_i)$ of the local fog nodes
    The distributed datasets is disseminated first at master node, first worker node and so on using round robin fashion. Each fog nodes uses same instances and assignment of the datasets.
  end
end

Algorithm 4: XGBoost parallel testing algorithm in distributed model.

Input:
D_{st} :used dataset(D) for testing with size st
C_l : dataset D_{st} and the number of classes
Output:
classification of normal(0) and attack(1)
1. extract the required features from dataset (D_{st})
2. the required feature (f1,f2,f3, ...,f_m) is obtained dataset (D_{st})
3. features are combined with set {f1,f2,f3, ...,f_d}, where d <= features in dataset (D_{st})
4. Repeat
5. Evaluate the probability $k_{i,j}$ for the features of dataset (D_{st}) using XGBoost
6. Until each required features is being evaluated from dataset (D_{st})
7. Evaluate the weight of dataset (D_{st}) features and probability is computed of $k_{i,j}$ by Eq. (15)

$$L_{log}(Y, P(y|z)) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j} log\, k_{i,j} \tag{15}$$

where N denotes the number of actual features whereas C signifies the attack or normal classes. When it belongs to class 1 (attack), the yi, j shows the ith features; otherwise, it is 0. (normal).

4. Experimental results and discussion

The fog computing and distributed IDS is applied, to reduce DDoS attacks against mining pools in blockchain-enabled IoT network is the subject of this section's experimental research. The experiment is carried out using the Python programming language. The performance of our proposed model is assessed using the BoT-dataset. Experiments are carried out on Tyrone PC run by Intel(R) Xeon(R) Silver 4114 CPU @ 2.20 GHz (2 processors), 128 GB RAM and 2 TB hard disk.

4.1. Description of BoT-IoT dataset

A particular-dataset is used to validate the proposed distributed detection method. The BoT-dataset [13] was developed at the UNSW Canberra Cyber Range Center by developing a realistic network environment. The Message Queuing Telemetry Transport (MQTT) protocol is used to create this data collection that connects device-to-device communications used as an alternative for blockchain-based IoT solutions. The numerous features are shown in the Table 1 of BoT-IoT dataset. The various attacks are briefly discussed below:

1. DoS Attacks [14]: Denial-of-Service (DoS) attacks arise as malicious cyber threat devices prevent authenticated IoT devices from accessing information systems, servers, or other network services. A DoS attack gets created by flooding the target particular host or a entire network as sending traffic until it becomes unresponsive or crashes, blocking valid IoT devices from accessing it.
2. DDoS Attacks [20]: When multiple IoT sensor devices perform working together to reach an attack a single target, it's called a Distributed Denial-of-Service (DDoS) attack. For large-scale attacks, botnet is a way through attackers apply DDoS attack scenarios, to compromise the network of IoT devices connected with internet. By controlling the devices using several commands to hack IoT machines, attackers profit from security vulnerabilities or system deficiencies. When more and more IoT devices come online, DDoS attacks are becoming more serious. IoT systems often use default keys that aren't protected through tone, that makes the system vulnerable and chances of attacks and exploitation. IoT system penetration often goes unnoticed by consumers, and an attacker can potentially hack hundreds of thousands of these networks to launch a large-scale attack without the knowledge of network owners.

Table 1
BoT-IoT dataset features description.

Feature number	Features	Description
F1	pkSeqID	rows Identification
F2	proto	protocols of Transaction shows network flow
F3	saddr	Source IP address
F4	sport	port number of source
F5	daddr	Destination IP address
F6	dport	port number of destination
F7	seq	Record total duration
F8	stddev	aggregated records with its standard deviation
F9	N_IN_Conn_P_SrcIP	source ip inbound connection
F10	min	minimum time duration for aggregate transactions
F11	state_number	numerical state features selection
F12	mean	aggregate record average time duration
F13	N_IN_Conn_P_DstIP	destination ip inboudn connection
F14	drate	per second time for destination to source packet
F15	srate	per second time for source to destination packets
F16	max	aggregated record maximum duration time
F17	attack	level 1 is for attack and level 0 for normal
F18	category	it represent traffic types
F19	subcategory	It shows traffic sub types

Table 2
BoT-IoT dataset distribution for normal and attack instances in training set.

Class wise division	No. of occurrences	Class frequency (in %)
DDoS	15,41,315	52.5183%
DoS	13,20,148	44.9823%
Recon	72,919	2.4846%
Normal	370	0.0126%
Theft	65	0.0022%
Total	29,34,817	100%

Table 3
BoT-IoT dataset distribution for normal and attack instances in testing set.

Class-wise division	No. of occurrences	Class frequency (in %)
DDoS	3,85,309	52.5155%
DoS	3,30,112	44.9925%
Recon	18,163	2.4755%
Normal	107	0.0146%
Theft	14	0.0019%
Total	7,33,705	100%

3. Reconnaissance Attacks [20]:Reconnaissance (Recon) is the process of gathering or probing knowledge in order to assess a network's vulnerabilities, and is then used to initiate an effective assault. Traffic analysis, packet sniffers, network port inspection, and IP address queries are examples of reconnaissance attacks.
4. Theft Attacks [20]: This is a type of attack in which the hacker attempts to make control over the IoT system's protection in order to obtain access to confidential data. Data leakage and keylogging are examples of theft attacks. During data stealing attacks, an attacker attempts to hack a remote IoT device, obtaining unauthorized access to data that can be sent to a remote attack computer. During keylogging operations, the attacker takes advantage of the remote host to record user keystrokes and extract confidential data.

The distribution, cumulative facts of attack, and typical attributes present in the training and testing sets of the BoT-IoT dataset are seen in Tables 2 and 3. There are a total of 29,34,817 instances in the training dataset. There are 15,41,315 DDoS instances, 13,20,148 DoS instances, 72,919 Recon instances, 370 regular instances, and 65 instances of Theft attacks. After that, there are 7,33,705 instances in the Testing dataset. There are 3,85,309 DDoS attacks, 3,30,112 DoS attacks, 18,163 Recon attacks, 107 regular attacks, and 14 theft attacks among them. The measurement metrics used in the proposed model are defined in the following sub-section.

4.2. Description of evaluation metrics

The identification performance of the proposed model was assessed using a variety of measurement criteria. Accuracy, Detection Rate, False Alarm Rate, Precision, True Negative Rate, and $\mathbb{F}\mathbb{1}$ score are the most specific assessment metrics. These metrics are based on the four criteria mentioned below [27]:

- True Positive ($\mathbb{TP}$): The graph shows that correct classification of malicious behavior with number of malicious observational activities in datasets.
- True Negative ($\mathbb{TN}$): The graph shows normal classification of behavior in the model with number of normal activities within the datasets.
- False Positive ($\mathbb{FP}$): This graph depicts how much the detection model incorrectly classifies normal IoT network traffic and findings as suspicious behavior.
- False Negative ($\mathbb{FN}$): This graph depicts how many malicious instances of IoT network traffic the identity model wrongly classifies as common operation.

The above parameters are used to compute the evaluation metrics which is discussed below:

(a) Accuracy ($\mathbb{AC}$): It measures how many times the model correctly identified out of the total number of findings in the research samples. When calculating the precision of the model, it takes into account both TP and *mathbbmTN* [27].

$$\mathbb{AC} = \frac{\mathbb{TP} + \mathbb{TN}}{\mathbb{FN} + \mathbb{TP} + \mathbb{FP} + \mathbb{TN}} \tag{16}$$

(b) Detection Rate ($\mathbb{DR}$): By splitting the cumulative number of attacks in the research sets, it reflects the observations of the model's number of detected attacks. Recall [27] is the name given to it.

$$\mathbb{DR} = \frac{\mathbb{TP}}{\mathbb{FN} + \mathbb{TP}} \tag{17}$$

(c) Precision ($\mathbb{PR}$): In the model [27], it reflects the number of observed attacks and their observations detected by dividing the total numbers of classified attack observation.

$$\mathbb{PR} = \frac{\mathbb{TP}}{\mathbb{TP} + \mathbb{FP}} \tag{18}$$

(d) False Alarm Rate ($\mathbb{FAR}$): The total number of normal observations in the dataset [27] is divided by the total numbers of normal observation to reflect with normal observations specified as an attack.

$$\mathbb{FAR} = \frac{\mathbb{FP}}{\mathbb{FP} + \mathbb{TN}} \tag{19}$$

(e) $\mathbb{F}\mathbb{1}$ Score: The $\mathbb{PR}$ and $\mathbb{DR}$ weighted average is calculated. It is mostly used when class distribution is skewed, and it is more valuable than precision since it accounts for $\mathbb{FP}$ and $\mathbb{FN}$ when calculating [27].

$$\mathbb{F}\mathbb{1} = 2 * \frac{\mathbb{RC} * \mathbb{PR}}{\mathbb{RC} + \mathbb{PR}} \tag{20}$$

Table 4
Binary classification confusion matrix for BoT-IoT dataset using Random Forest algorithm.

Actual class	Predicted class	
	Attack	Normal
Attack	732196	1402
Normal	0	107

Table 5
Binary classification confusion matrix for BoT-IoT dataset using XGBoost algorithm.

Actual class	Predicted class	
	Attack	Normal
Attack	733589	9
Normal	0	107

(f) True Negative Rate ($\mathbb{TNR}$): It specifies the amount of real normal instances in the detection model [27] that is predicted as normal.

$$\mathbb{TNR} = \frac{\mathbb{TN}}{\mathbb{TN} + \mathbb{FP}} \tag{21}$$

4.3. Experimental environment

We used binary and multi-class classification categories to investigate the detection model's results. In the blockchain-based IoT setting, unknowing test datasets are taken to reflect botnets in output evaluation. The experiment is carried out by spreading machine learning techniques to several coordinated nodes in order to detect DDoS attacks. We have taken different number of devices used for testing the network, to measure the reliability of parallelism and dissemination. The following hyper-parameter optimization is used in the ML technique for random forest: the Gini index for estimation of impurity, the maximum tree depth is set to 200, the lowest samples number at the leaf node is set to 1, the lowest samples number are used to divide the inner leaf nodes and set to 6, and the forest is set to 50 for number of trees generation. The learning rate for XGBoost is also set to 0.1, the maximum tree depth is 200, the L1 regularization alpha is 0, and the L2 regularization alpha is 1. The following section investigates and responds to classification results using a variety of assessment techniques.

4.4. Evaluation and discussion of results

In this part, various performance assessment criteria are used to measure the outcomes. In the evaluation process, a Confusion matrix and class-prediction results are used. The confusion matrix, also known as an error matrix, is created and used to assess the efficacy of the machine learning technique. The binary classification results are shown in Table 4 and Table 5. The mathematical parameter true positive is shown around the uncertainty matrix's main diagonal, while the others are determined using the formula stated earlier. The confusion matrix for Random forest is shown in Table 4. The actual classes are represented by rows in the uncertainty matrix, while the predicted classes are represented by columns in the classification algorithm predictions. True Positive samples are 7,32,196 for attack and 107 for normal cases, according to the main diagonal. False negatives (FN) of a given class can be computed from the preceding row without taking into account the true positive of that class, for example: attack class false negatives (FN) is 1402 when true positive is removed. Table 5 shows the confusion matrix for XGBoost in binary classification. The predicted model has detected attack instances 7,33,589 and has FN of 9 observations.

In Table 6 and 7 multi-class classification confusion matrix (CM) for BoT-IoT dataset is shown. Table 6, shows results of CM for random forest. It is worth noting that, true positive (TP) for DDoS attack is 3,85,251 and it has overall 58 instances of FN. In Table 7 performance result of XGboost algorithm is shown through CM. It can be seen that the diagonal shows less TP instances compared to random forest. However, the detection system has FN of 58 for DDoS, 30 for Recon and 22 instances of normal. Further, we have compared class wise prediction results for random forest and XGBoost algorithm. Table 8 and 9, shows class wise prediction results for multi-class classification of BoT-IoT dataset. It is worth noting that, the detection system using random forest for DDoS attack has 0.99992095% ACcuracy (AC) and 0.99984947% Detection Rate (DR). Similarly for DDoS attack using XGBoost the IDS has AC of 0.99992095% and DR of 0.99984947%. False alarm rate (FAR) is also, a important parameter to evaluate the performance. In Tables 8 and 9, it can be seen that the model has 0.014371% FAR for DoS attack, 0% for other attack and normal instances using random forest. Whereas FAR with XGBoost for DDoS has 0%, DoS has 0.014371%, Recon attack has 0%, normal observation has 0.004193% and Theft attack has 0.002999%.

Table 6
Multi-class classification confusion matrix for BoT-IoT dataset using Random Forest algorithm.

Actual class	Predicted class				
	DDoS	DoS	Recon	Normal	Theft
DDoS	385251	58	0	0	0
DoS	0	330112	0	0	0
Recon	0	0	107	0	0
Normal	0	0	0	18163	0
Theft	0	0	0	0	14

Table 7
Multi-class classification confusion matrix for BoT-IoT dataset using XGBoost algorithm.

Actual class	Predicted class				
	DDoS	DoS	Recon	Normal	Theft
DDoS	385251	58	0	0	0
DoS	0	330112	0	0	0
Recon	0	0	77	30	0
Normal	0	0	0	18141	22
Theft	0	0	0	0	14

Table 8
Multi-class wise prediction results for BoT-IoT dataset using Random Forest algorithm.

Attack	Performance metrics				
	AC	PR	DR	TNR	FAR
DDoS	0.99992095	1.00	0.99984947	1.00	0.0
DoS	0.99992095	0.99982433	1.00	0.99985629	0.014371
Recon	1.00	1.00	1.00	1.00	0.0
Normal	1.00	1.00	1.00	1.00	0.0
Theft	1.00	1.00	1.00	1.00	0.0

Table 9
Multi-class wise prediction results for BoT-IoT dataset using XGBoost algorithm.

Attack	Performance metrics				
	AC	PR	DR	TNR	FAR
DDoS	0.99992095	1.00	0.99984947	1.00	0.0
DoS	0.99992095	0.99982433	1.00	0.99985629	0.014371
Recon	0.99995911	1.00	0.71962617	1.00	0.0
Normal	0.99992913	0.99834902	0.99878875	0.99995807	0.004193
Theft	0.99997002	0.38888889	1.00	0.99997001	0.002999

The overall performance evaluation of RF and XGBoost in terms of AC, PR, DR, F1 score and processing time is discussed using BoT-IoT dataset as demonstrated in Fig. 6 and Fig. 7. In the case of binary classification, the proposed detection system using RF has achieved 99.8089% AC, DR of 99.8088%, PR of 100% and F1 of 99.9043%. Similarly for XGBoost the model has AC of 99.9987%, PR

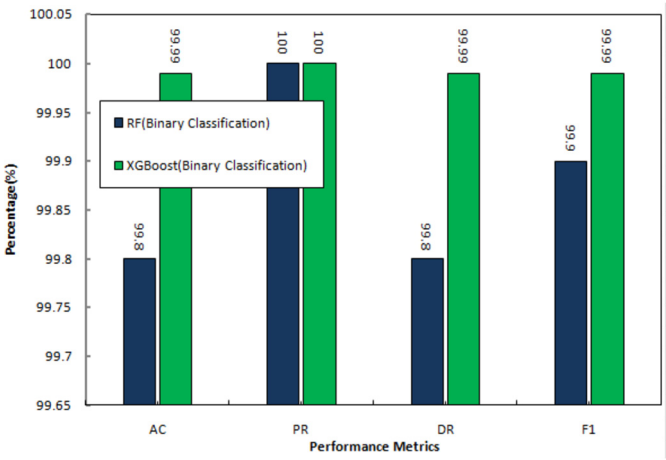


Fig. 6. Overall performance of RF and XGBoost for binary classification.

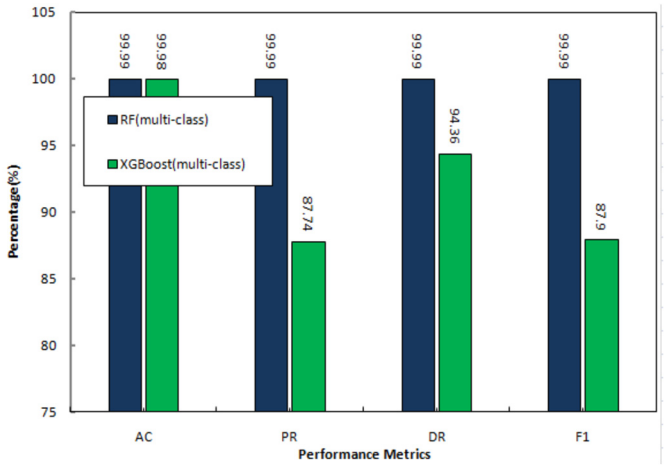


Fig. 7. Overall performance of RF and XGBoost for multi-class classification.

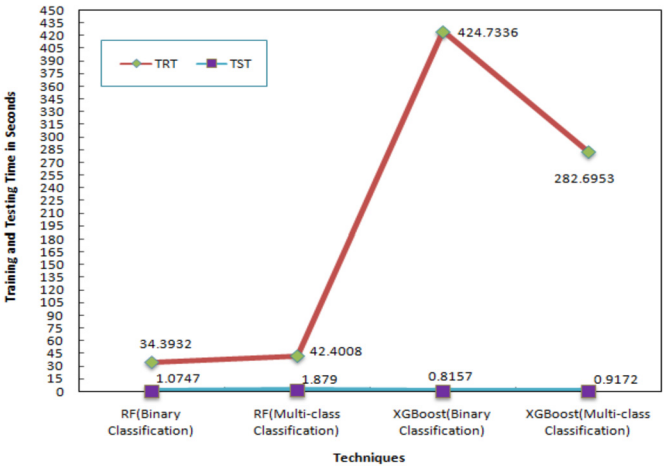


Fig. 8. Training and Testing Time comparison for RF and XGBoost.

of 100%, DR of 99.9993% and F1 of 99.9993%. On the other hand, for multi-class classification the proposed detection system using RF has achieved 99.985% AC, DR of 99.997%, PR of 99.996% and F1 of 99.997%. Finally, the XGBoost the model has AC of 99.985%, PR of 87.741%, DR of 94.365% and F1 of 87.907%.

Processing time is an important factor in the design of IDS. In real time IoT network generates huge data at regular intervals. Therefore, detection system with less processing time is needed in such environments. In this experiment, we have compared training (TRT) and testing (TST) time for binary vs multi-class classification used by the distributed IDS. Fig. 8, shows TRT and TST in seconds for both algorithms. In the case of binary classification using RF, it can be seen that the TRT is 34.3932 and TST is 1.0747 seconds. On the other hand, for XGBoost TRT is 424.7336 and TST is 0.8157 seconds. However, for multi-class classification, the proposed model using RF, takes TRT of 42.4008 and TST of 1.879 seconds. Similarly using XGBoost, the detection system takes TRT of 282.6953 and TST of 0.9172 seconds.

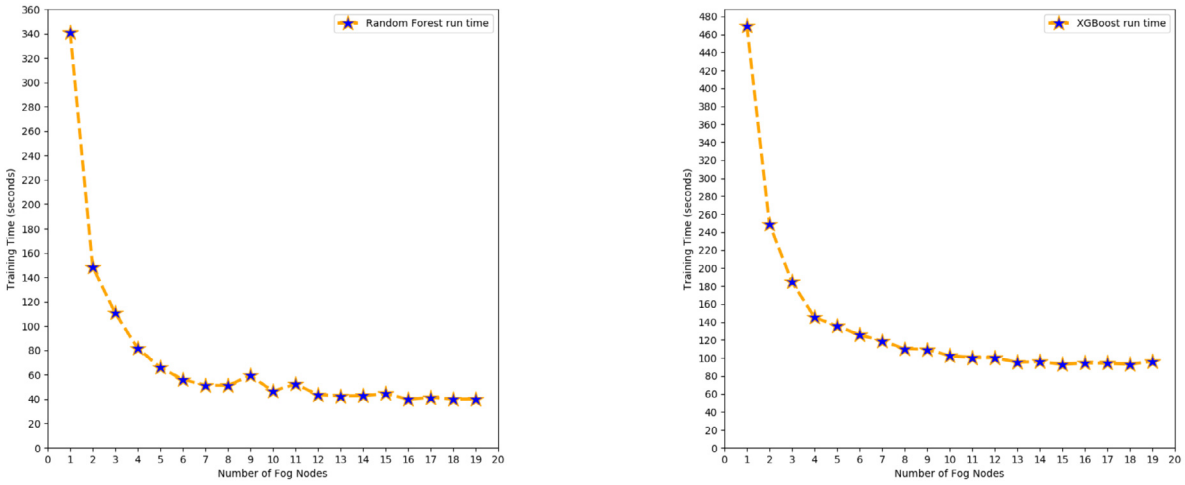
We have trained the proposed distributed detection method on a variety of fog nodes ranging from one to twenty. TRT in seconds for random forest(RF) and XGboost, for binary and multi-class grouping, as shown in Figs. 9 and 10. It can be noticed that, while RF has no effect with TRT after 12 fog nodes, XGBoost's TRT stays the similar behavior after 13 fog nodes when complete features are used. On the other hand, for multi-class grouping, there is no significant effect on TRT after 14 fog nodes for RF, but after 13 fog nodes for XGBoost. It's worth noting that the dispersed IDS trains quicker in binary grouping, as the TRT for RF steadily decreases from 340 to 155 seconds using two fog nodes, 155 to 110 seconds, and so on. Similarly, when using two fog nodes, the TRT is reduced from 470 to 250 seconds, and when using three fog nodes, the TRT is reduced from 250 to 190 seconds, and so on. Multi-class grouping, on the other hand, cuts RF TRT from 360 to 190 seconds when using two fog nodes, 190 to 130 seconds when using three fog nodes, and so on. Finally, TRT for XGBoost for two fog nodes is reduced from 1480 to 920 seconds, 920 to 530 seconds for three fog nodes, and so on. In binary classification, AI-based all ML strategies require less time to learn than multi-class classification using distributed fog nodes.

4.5. Overview of two classifiers and discussion of performance

Overall, the experimental results indicate high accuracy in both binary and multi-class classifications. Training time for binary classification is less compared to multi-class classification. Additionally, the overall performance of XGboost in binary and random forest in multi- class shows significant performance. This also indicates that ML techniques have enormous potential to transform the cybersecurity direction as attack detection in distributed blockchain-IoT/Fog environment.

5. Conclusion

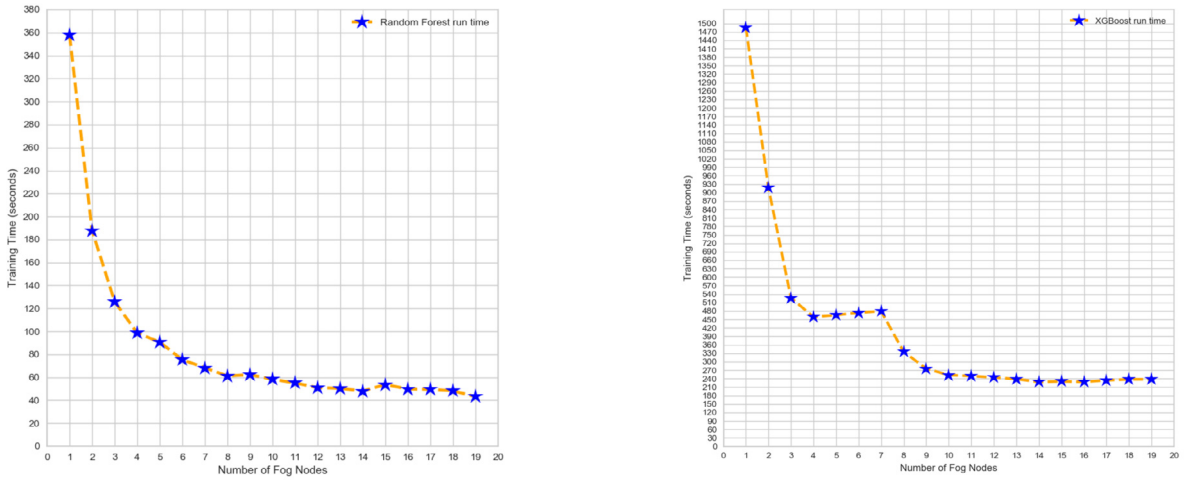
This paper illustrates the numerous limitations and vulnerabilities of stand-alone IoT systems and how blockchain can provide a



(a) Distributed Training time for RF

(b) Distributed Training time for XGBoost

Fig. 9. Distributed Training time for Binary Classification on multiple fog nodes.



(a) Distributed Training time for RF

(b) Distributed Training time for XGBoost

Fig. 10. Distributed Training time for multi-class Classification on multiple fog nodes.

decentralized network to fulfill these requirements. In this paper, we proposed a distributed IDS that integrates AI and fog computing. In order to detect DDoS attacks, the detection system was integrated with mining pool in a blockchain-enabled IoT network. The proposed distributed detection system works on three main engines. The first, traffic processing engine, includes fog nodes for preprocessing of network traffic by normalizing features using StandardScaler, that scale features to a specific scale. Two AI-based ML techniques, random forest and XGBoost are deployed in distributed blockchain-IoT environment. The second, intrusion detection engine that follows data preprocessing step and finally IoT incoming traffic was analyzed for the detection of normal and abnormal transactions. The third, transaction handling engine, based on detection results transactions are categorized into normal and malicious instances. Normal transactions are executed by miners in mining pool and then gets added to blockchain network. The results using BoT-IoT dataset indicate that the proposed model was effective in detecting IoT-based attacks and has high performance by taking less processing time on multiple fog nodes. In future, we plan to extend this work by applying different deep learning techniques, which could improve the performance of the distributed detection system.

CRediT authorship contribution statement

All the authors have participated sufficiently in the paper entitled “A Distributed Intrusion Detection System to Detect DDoS Attacks in Blockchain-enabled IoT Network” submitted to the Journal of Parallel and Distributed Computing (Elsevier) and take responsibility of the content of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP 2021/18.

References

[1] E. Anthi, L. Williams, P. Burnap Pulse, An adaptive intrusion detection for the Internet of things, in: Living in the Internet of Things: Cybersecurity of the IoT - 2018, 2018, pp. 1–4.

[2] M. Apostolaki, A. Zohar, L. Vanbever, Hijacking bitcoin: routing attacks on cryptocurrencies, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 375–392.

[3] P. B, N. Deepa, Q.-V. Pham, D.C. Nguyen, P.K.R. M, T.R. G, P.N. Pathirana, O. Dobre, Toward blockchain for edge-of-things: a new paradigm, opportunities, and future directions, IEEE Int. Things Mag. 4 (2) (2021) 102–108, https://doi.org/10.1109/IOTM.0001.2000191.

[4] S.T. Bakhsh, S. Alghamdi, R.A. Alsemmeari, S.R. Hassan, An adaptive intrusion detection and prevention system for Internet of things, Int. J. Distrib. Sens. Netw. 15 (11) (2019) 1550147719888109, https://doi.org/10.1177/1550147719888109.

[5] M. Bastiaan, Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin, available at http://referaat.cs.utwente.nl/conference/22/paper/7473/preventingthe-51-attack-a-stochasticanalysis-oftwo-phase-proof-of-work-in-bitcoin.pdf, 2015.

[6] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J.C.-W. Lin, G. Fortino, Deep learning for pedestrian collective behavior analysis in smart cities: a model of group trajectory outlier detection, Inf. Fusion 65 (2021) 13–20.

[7] I. Eyal, E.G. Sirer, Majority is not enough: bitcoin mining is vulnerable, in: International Conference on Financial Cryptography and Data Security, Springer, 2014, pp. 436–454.

[8] I. Eyal, E.G. Sirer, How to disincentivize large bitcoin mining pools, Blog post: http://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools.

[9] T.R. Gadekallu, Q.-V. Pham, D.C. Nguyen, P.K.R. Maddikunta, N. Deepa, B. Prabadevi, P.N. Pathirana, J. Zhao, W.-J. Hwang, Blockchain for edge of things: applications, opportunities, and challenges, IEEE Int. Things J. 9 (2) (2022) 964–988, https://doi.org/10.1109/JIOT.2021.3119639.

[10] A.R. Javed, S.u. u. Rehman, M.U. Khan, M. Alazab, T.R. G, Canintelliids: detecting in-vehicle intrusion attacks on a controller area network using cnn and attention-based gru, IEEE Trans. Netw. Sci. Eng. 8 (2) (2021) 1456–1466, https://doi.org/10.1109/TNSE.2021.3059881.

[11] G.O. Karame, E. Androulaki, S. Capkun, Double-spending fast payments in bitcoin, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, 2012, pp. 906–917.

[12] R. Kaushal, Bitcoin: vulnerabilities and attacks, Imp. J. Interdiscip. Res. 2 (7) (2016) 944–946.

[13] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, Benjamin Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.

[14] P. Kumar, G.P. Gupta, R. Tripathi, Design of anomaly-based intrusion detection system using fog computing for iot network, Autom. Control Comput. Sci. 55 (2) (2021) 137–147.

[15] P. Kumar, G.P. Gupta, R. Tripathi, Tp2sf: a trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning, J. Syst. Archit. 115 (2021) 101954.

[16] P. Kumar, G.P. Gupta, R. Tripathi, A distributed ensemble design based intrusion detection system using fog computing to protect the Internet of things networks, J. Ambient Intell. Humaniz. Comput. 12 (10) (2021) 9555–9572.

[17] P. Kumar, G.P. Gupta, R. Tripathi, Pefl: deep privacy-encoding based federated learning framework for smart agriculture, IEEE MICRO (2021) 1, https://doi.org/10.1109/MM.2021.3112476.

[18] P. Kumar, G.P. Gupta, R. Tripathi, Toward design of an intelligent cyber attack detection system using hybrid feature reduced approach for iot networks, Arab. J. Sci. Eng. 46 (4) (2021) 3749–3778.

[19] P. Kumar, G.P. Gupta, R. Tripathi, S. Garg, M.M. Hassan, Dltif: deep learning-driven cyber threat intelligence modeling and identification framework in iot-enabled maritime transportation systems, IEEE Trans. Intell. Transp. Syst. (2021) 1–10, https://doi.org/10.1109/TITS.2021.3122368.

[20] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, A distributed framework for detecting ddos attacks in smart contract-based blockchain-iot systems by leveraging fog computing, Trans. Emerg. Telecommun. Technol. 32 (6) (2021) e4112.

[21] P. Kumar, R. Kumar, G. Srivastava, G.P. Gupta, R. Tripathi, T.R. Gadekallu, N. Xiong, Ppsf: a privacy-preserving and secure framework using blockchain-based machine-learning for iot-driven smart cities, IEEE Trans. Netw. Sci. Eng. (2021) 1, https://doi.org/10.1109/TNSE.2021.3089435.

[22] P. Kumar, R. Tripathi, G.P. Gupta, P2IDF: A Privacy-Preserving Based Intrusion Detection Framework for Software Defined Internet of Things-Fog (SDIoT-Fog), Association for Computing Machinery, New York, NY, USA, 2021, pp. 37–42.

[23] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, G. Srivastava, P2tif: a blockchain and deep learning framework for privacy-preserved threat intelligence in industrial iot, IEEE Trans. Ind. Inform. (2022) 1, https://doi.org/10.1109/TII.2022.3142030.

[24] R. Kumar, R. Tripathi, Data provenance and access control rules for ownership transfer using blockchain, Int. J. Inf. Secur. Priv. (IJISP) 15 (2) (2021) 87–112.

[25] R. Kumar, R. Tripathi, Dbtp2sf: a deep blockchain-based trustworthy privacy-preserving secured framework in industrial Internet of things systems, Trans. Emerg. Telecommun. Technol. 32 (4) (2021) e4222.

[26] R. Kumar, R. Tripathi, Large-scale data storage scheme in blockchain ledger using ipfs and nosql, in: Large-Scale Data Streaming, Processing, and Blockchain Security, IGI Global, 2021, pp. 91–116.

[27] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, T.R. Gadekallu, G. Srivastava, Sp2f: a secured privacy-preserving framework for smart agricultural unmanned aerial vehicles, Comput. Netw. 187 (2021) 107819.

[28] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, S. Garg, M.M. Hassan, Bdtwin: an integrated framework for enhancing security and privacy in cybertwin-driven automotive industrial Internet of things, IEEE Int. Things J. (2021) 1, https://doi.org/10.1109/JIOT.2021.3122021.

[29] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, N. Kumar, P2sf-iov: a privacy-preservation-based secured framework for Internet of vehicles, IEEE Trans. Intell. Transp. Syst. (2021) 1–12, https://doi.org/10.1109/TITS.2021.3102581.

[30] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, N. Kumar, M.M. Hassan, A privacy-preserving-based secure framework using blockchain-enabled deep-learning in cooperative intelligent transport system, IEEE Trans. Intell. Transp. Syst. (2021) 1–12, https://doi.org/10.1109/TITS.2021.3098636.

[31] R. Kumar, R. Tripathi, N. Marchang, G. Srivastava, T.R. Gadekallu, N.N. Xiong, A secured distributed detection system based on ipfs and blockchain for industrial image and video data security, J. Parallel Distrib. Comput. 152 (2021) 128–143.

[32] Y. Kwon, D. Kim, Y. Son, E. Vasserman, Y. Kim, Be selfish and avoid dilemmas: fork after withholding (faw) attacks on bitcoin, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 195–209.

[33] S.A. Latif, F.B.X. Wen, C. Iwendi, F.W. Li-li, S.M. Mohsin, Z. Han, S.S. Band, Ai-empowered, blockchain and sdn integrated security architecture for iot network of cyber physical systems, Comput. Commun. 181 (2022) 274–283.

[34] V. Mothukuri, P. Khare, R.M. Parizi, S. Pouriyeh, A. Dehghantanha, G. Srivastava, Federated learning-based anomaly detection for iot security attacks, IEEE Int. Things J. (2021) 1, https://doi.org/10.1109/JIOT.2021.3077803.

[35] M. Rosenfeld, Analysis of bitcoin pooled mining reward systems, arXiv preprint, arXiv:1112.4980.

[36] M. Saad, J. Choi, D. Nyang, J. Kim, A. Mohaisen, Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions, IEEE Syst. J..

[37] M. Shabbir, A. Shabbir, C. Iwendi, A.R. Javed, M. Rizwan, N. Herencsar, J.C.-W. Lin, Enhancing security of health information using modular encryption standard in mobile cloud computing, IEEE Access 9 (2021) 8820–8834.

[38] M. Shafiq, Z. Tian, A.K. Bashir, X. Du, M. Guizani, Corrauc: a malicious bot-iot traffic detection method in iot network using machine-learning techniques, IEEE Int. Things J. 8 (5) (2021) 3242–3254, https://doi.org/10.1109/JIOT.2020.3002255.

[39] S. ur Rehman, M. Khaliq, S.I. Imtiaz, A. Rasool, M. Shafiq, A.R. Javed, Z. Jalil, A.K. Bashir, Diddos: an approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (gru), Future Gener. Comput. Syst. 118 (2021) 453–466.

[40] M. Vasek, M. Thornton, T. Moore, Empirical analysis of denial-of-service attacks in the bitcoin ecosystem, in: International Conference on Financial Cryptography and Data Security, Springer, 2014, pp. 57–71.

[41] Willy Woo, Charts: determining the ideal block size for bitcoin, https://www.coindesk.com/charts-determining-ideal-block-size-bitcoin, 2017. (Accessed 10 January 2020), Online.

Randhir Kumar is working towards the Ph.D. degree in Department of Information Technology, National Institute of Technology, Raipur. He has published more than 20 research articles in the areas of Blockchain Technology and It's Framework. His research interests include Blockchain Technology, Cryptography Techniques, Information Security, Web Mining, and Image Processing.

Prabhat Kumar is working towards his Ph.D. degree in Information Technology, National Institute of Technology, Raipur, India. He earned his Ph.D. scholarship position as a talented student. He has over 15 publications in high-ranked Journals and Conferences. His research interests are Security and Privacy of the Internet of Things, Software-defined Networking, and Blockchain. He is also an IEEE Student Member.

Rakesh Tripathi received his Ph.D. degree in computer science and engineering from the Indian Institute of Technology Guwahati, India. He is an Assistant Professor with the Department of Information Technology, National Institute of Technology, Raipur, India. He has over ten years of experience in academic. He has published over 20 referred article and served as a Reviewer of several journals. His research interests include Mobile-Adhoc Networks, Sensor Networks, Data Center Networks, Distributed Systems, Network Security, Blockchain and Game Theory in Networks.

Govind P. Gupta received his Ph.D. degree from Indian Institute of Technology, Roorkee, India, in 2014. He is currently an Assistant Professor in the Department of Information Technology at National Institute of Technology, Raipur, India. His current research interests include efficient protocol design for Wireless Sensor Networks and Internet of Things, Network Security and Software-defined Networking. He is a professional member of the IEEE and ACM.

Sahil Garg (S'15, M'18) received his Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He is currently a postdoctoral research fellow at École de Technologie Supérieure, Université du Québec, Montréal, Canada. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing.

Mohammad Mehedi Hassan received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in February 2011. He is currently an Associate Professor with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.