Computer Communications 197 (2023) 173–185

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Impact on blockchain-based AI/ML-enabled big data analytics for Cognitive Internet of Things environment

Ankush Mitra [a], Basudeb Bera [a], Ashok Kumar Das [a], Sajjad Shaukat Jamal [b], Ilsun You [c,*]

[a] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[b] Department of Mathematics, College of Science, King Khalid University, Abha 614133, Saudi Arabia
[c] Department of Financial Information Security, Kookmin University, Seoul, South Korea

ARTICLE INFO

Keywords:
Cognitive Internet of Things (CIoT)
Blockchain
Security
Big data analytics
Artificial intelligence
Data poisoning attacks

ABSTRACT

Cognitive Internet of Things (CIoT) supports the organizations to learn from the information (data) arriving from various connected devices, sensors, machines and other sources, and at the same time it inspires intelligence into different business operations, products, customer experiences, and people. Data poising attacks are very serious concerns because they may play a significant factor for businesses and organizations for both financial terms and damaging their reputations, when the Big data analytics on the analyzed data is itself corrupted. To mitigate this issue, in this paper, we suggest a blockchain-based Artificial Intelligence(AI)/Machine Learning(ML)-enabled Big data analytics mechanism for CIoT environment. The comprehensive experimental results have been provided under two circumstances: (1) performance of the ML model under data poisoning attacks and (2) performance of the ML model without data poisoning attacks. In the first case, we show how the data poison attacks can effect the ML model when the data is on some cloud storage (i.e. not in the blockchains), whereas in the second case we show the effect when the data is in the blockchains (i.e., without data poisoning attacks). The experimental results demonstrate that we have significant gains in performance in terms of accuracy, recall, precision and F1 score when there are no data poisoning attacks on the data. Moreover, a detailed blockchain simulation has carried out to demonstrate the practical aspects of the proposed security framework.

1. Introduction

Internet of Things (IoT) has become an emerging technology due to a huge enhancement of Information and Communications Technology (ICT). IoT compromises a large number of smart devices, called IoT devices, which can be either physical or virtual objects [1]. The devices are interconnected among each other to communicate and exchange information among them. The devices are also assigned unique identity or an Internet address, such as Internet Protocol version 6 (IPv6) address, which is quite possible. IPv6 "Low power Wireless Personal Area Network (6LoWPAN)" IoT smart devices can work with very constrained resource environment, such as limited energy, low energy and low computation power [2]. Since the IoT devices typically communicate among each other and then to their nearby gateway nodes (access points) via wireless communication, there are several threats including impersonation, replay, physical device capture, man-in-the-middle and privileged-insider attacks [3]. To secure an IoT network, several IoT-based security protocols have been suggested by the researchers, such as authentication [4–10], key management [11–17], access control [18–27] and intrusion detection [28–31].

Cognitive Internet of Things (CIoT) has now become a new network model which is enhancement of IoT. Likewise IoT network, physical and virtual objects are part of CIoT, which work with minimum human intervention and they also communicate with each other based on a "context-aware perception–action cycle" [32]. CIoT applies understanding-by-building technique in order to learn from both social networks as well as physical environment for the purpose of storing the learned semantic and knowledge in kinds of databases. After that they adapt to uncertainties or changes with the help of resource efficient decision-making approaches. CIoT has two main objectives. The first objective is to bridge the physical world and the social world, in order to form an intelligent Physical Cyber Social (PCS) system. The second objective is towards permitting "smart resource allotment", "intelligent service provisioning" and also "automatic network operation".

CIoT is thus considered as a field of science where IoT and cognitive computing are applied to make the IoT systems smarter. It provides some kind of thinking ability to the IoT systems. Though the computers do not yet think like humans, but we can use the cognitive computing techniques to make the IoT system to understand, learn and reason the

* Corresponding author.
E-mail addresses: ankush.mitra@alumni.iiit.ac.in (A. Mitra), basudeb.bera@research.iiit.ac.in (B. Bera), ashok.das@iiit.ac.in (A.K. Das), shussain@kku.edu.sa (S.S. Jamal), ilsunu@gmail.com (I. You).

https://doi.org/10.1016/j.comcom.2022.10.010
Received 6 April 2021; Received in revised form 2 November 2021; Accepted 14 October 2022
Available online 1 November 2022

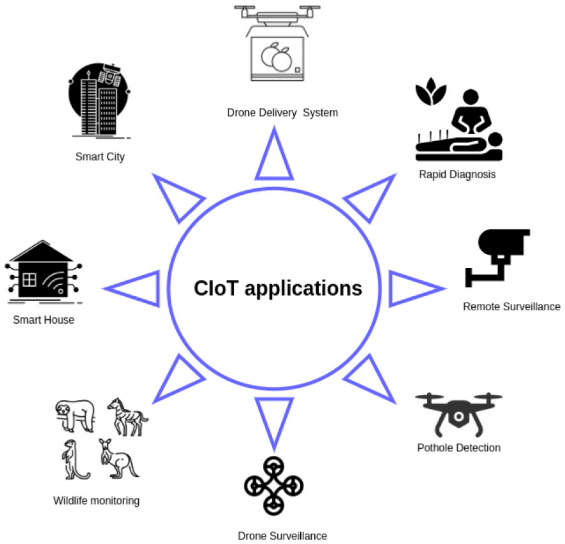


Fig. 1. Various CIoT-enabled applications.

data coming from the IoT systems. The IoT system is nothing but a group of sensors (smart devices) which are connected via the internet to collect the data from a real time environment. We can apply cognitive computing techniques on these data to learn, reason and predict certain things.

The shortage of spectrum resources makes the development of an Industrial Internet of Things (IIoT) to be limited. With the help of cognitive radio, the cognitive IIoT (CIIoT) can improve the spectrum utilization via sensing as well as accessing the idle spectrum [33]. For improving the sensing along with transmission performance of CIIoT, Liu and Zhang [33] suggested a cluster-based CIIoT, wherein the cluster heads are responsible for executing cooperative spectrum sensing in order to obtain the available spectrum, and the nodes also transmit via "non-orthogonal multiple access (NOMA)". In this architecture, it was shown that the "spectrum access probability" and "average total throughput" of the CIIoT are reduced.

Liu and Zhang [34] investigated a 5th generation mobile network (5G)-based IoT for accessing the 5G spectrum in order to transfer 5G and IoT data concurrently. An IoT node can utilize the 5G network for transmitting both voice and video data, where that node can use the IoT network for forwarding the sensing information. In addition, they also provided another optimization problem for maximizing the IoT transmission rate under the 5G transmission rate constraint.

1.1. Motivation

There are several potential applications using CIoT as shown in Fig. 1. CIoT is a technology in which we can combine both IoT and machine learning (ML) together in order to help us in many ways including automatic smart home application, automatic drone surveillance, automatic traffic detection, automatic pothole detection, etc. However, this technology has potential threats because it opens a window for data poison attack (i.e., corrupting the ML model with malicious data). This may cause severe consequences as this technology works in the real time applications. Instead, if the data is stored in a blockchain network, the data becomes tamper proof and reliable. As in blockchain the blocks are linked with each other with hashing technique, changing/updating/deleting the data could result in breaking the chain. In blockchain, we have consensus mechanisms by which we can ensure the reliability of the data. But, if the data is stored in any other system, such as cloud or fog computing servers except blockchain, the attacker can corrupt the raw data on which the ML model has to be trained. This can cause a significant drop of accuracy score of the ML model trained on corrupted data. In this paper, we aim to show the impact on the corrupted data which is not in the blockchain and also on the genuine data that is stored in the blockchain. For this purpose, we show that how a small change of data can cause a substantial accuracy score drop of the ML model.

1.2. Research contributions

The primary research contributions are enumerated as follows:

- We provide the network and threat models that are based on blockchain-based framework for Big data analytics purpose using AI/ML techniques in CIoT environment. We consider the data poising attacks on the data that can be stored without blockchain in cloud servers.
- Next, we discuss the overall mechanism in the proposed framework which considers how the data are being transmitted and collected by the IoT smart devices and gateway nodes. The securely collected in form of transactions are formed with the blocks and then added into a public blockchain using a well-known voting-based consensus algorithm.
- We provide extensive experimental results into two cases: (1) impact on Big data analytics using AI/ML technique on the data stored in cloud servers with different data poisoning attacks and (2) impact on Big data analytics using AI/ML technique on the data stored in blockchain-enabled cloud servers. The experimental results demonstrate that with blockchain-enabled solution the performance of ML-based deep learning algorithm, namely Convolution Neural Networks (CNN) [35], is significantly improved as compared to that without blockchain-enabled solution with respect to CIoT-based pothole data sets.
- Finally, we carry out a detailed blockchain-based simulation on the suggested security framework to show its impact on the required computational time in seconds when the number of transactions in each block and the number of blocks mined into the blockchain are varied with fixing the number of nodes in a Peer-to-Peer (P2P) cloud servers network.

1.3. Paper outline

The rest of the paper is sketched as follows. In Section 2, discussion on mathematical preliminaries is provided. The system models consisting of both the network and threat models are discussed in Section 3. The related work is then discussed in Section 4. A blockchain-based AI/ML-enabled Big data analytics security framework for CIoT environment is provided in Section 5. After that the experimental setup and discussion on various results are provided in Sections 6 and 7, respectively. In Section 8, we provide a detailed implementation of the blockchain in the proposed framework. Finally the paper is wound up in Section 9.

2. Preliminaries

In this section, we discuss briefly some mathematical preliminaries that are essential for discussion on the proposed framework and also for measuring the impact on the framework using AI/ML based Big data analytics in CIoT.

2.1. Artificial intelligence and machine learning

Artificial Intelligence (AI) is a computer science field where it is used to simulate human intelligence in computers so that they can think and act like human beings. It is the ability of a computer to perform tasks that are commonly associated with human beings. The task may be anything, like image recognition and speech recognition. Generally, the applications of AI involve in developing systems to mimic characteristic of humans, such as the "ability to reason, discover meaning, generalize, or learn from past experience". Now a days, AI is used in various other purposes including e-commerce, bio-medical engineering, automobile industry, smart home, smart city, etc.

Machine Learning (ML) is regarded as a subset of AI, which enables the machines to learn from past history or experiences without being programmed explicitly. Formally, ML is defined as follows: "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves experience E" [36].

ML is classified into three categories:

- *Supervised learning*: In this case, the computer is provided with the training data along with their desired outcomes, given by a supervisor. The goal is then to learn a general rule that assigns the inputs to the outputs.
- *Unsupervised learning*: It allows the machines to learn themselves without human intervention. In other words, under this category, no labels are supplied to the learning algorithm, and it is left on its own in order to find the appropriate structure in its inputs.
- *Reinforcement learning*: Here, the goal is to maximize the notion of cumulative rewards. Under this scenario, the training data is provided only as "feedback to the program's actions in a dynamic environment" (for instance, driving a vehicle).

The 21st century becomes the era of AI/ML. Now a days, ML is applied in almost every where. Some of the ML applications include image recognition, image detection, speech recognition, computer vision, smart IoT, smart medical environment, etc.

2.2. Attacks related to AI/ML

Like other networking environments, AI/ML is also prone to various attacks. Recently, AI/ML security becomes an emerging topic in the computer science field in order to make correct and accurate predictions on non-poisonous (corrupted) data. For instance, typically a huge volume of data generated by the IoT smart devices in CIoT can be stored in cloud server(s). As the cloud servers are semi-trusted, there is a possibility by the insider attackers of the cloud servers to perform several attacks as discussed below:

- *Adversarial input attacks*: These are specially crafted on inputs that have been developed with the aim of being reliably mis-classified in order to evade detection [37].
- *Data poisoning attacks*: An attacker may try to pollute the training data set on which the various machine learning models have to be learned [38]. The attacker can then insert false data, alter the label of the data, and also remove the data or insert random noise to the data to poison the training data.
- *Model attacks*: In such type of attack, an attacker may pollute the model's hyper-parameters (that is, the parameters that are learned using AI/ML) [39]. The model performance heavily depends on the hyper-parameters which it has learned during the learning phase. Hence, modifying the hyper-parameters can significantly drop the model performance.
- *Model stealing attacks*: Such kinds of attack scenarios are used to steal or duplicate models or recover training data membership via black-box probing [40].

2.3. Convolution neural networks (CNN)

Artificial neural network is a set of algorithms that endeavor to admit elementary relationships in a set of data through a mechanism that mimics the strategy similar to what the human brain works. In a neural network, a neuron is modeled as a node that takes the inputs and produces the output, which may be non-linear function of inputs.

Convolution Neural Networks (CNN) is a deep neural network that is generally used in computer vision [35]. CNN is a multi-layer perception where some of the layers are just convolution layers that are designed to extract features from the visual objects. A typical CNN architecture is illustrated in Fig. 2.

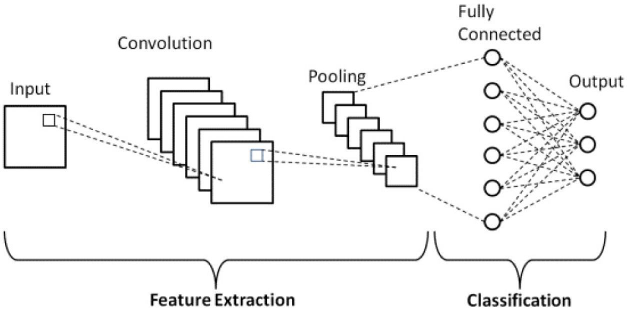


Fig. 2. CNN architecture [35].

CNN mainly contains three types of layers:

- *Convolution layer*: It is a convolution tool that splits various features of the input images for analysis.
- *Pooling layer*: The main goal of this layer is that it decreases the size of the convoluted feature map in order to reduce computation costs.
- *Fully connected layer*: This layer applies the output of the convolution layer for making prediction about the best description for the inputs (images).

The architecture of a CNN can be thought as a similar to that of the connectivity pattern of Neurons containing in a human brain. The individual neurons have the ability to stimuli only in a confined region of the visual field, which is the "receptive field". As a result, a collection of such fields overlap to enclose the whole visual portion.

2.4. Blockchain and its evolution in CIoT

Blockchain builds on the idea of a Peer-to-Peer (P2P) distributed network and offers a universal data set that every entity can trust, even if they might be not known or trust each other. It offers a shared as well as trusted ledger of various transactions. The immutable and encrypted copies of data (transactions) are stored on each node across the network. In other words, blockchain is treated as a distributed, immutable and transparent ledger. The blocks of transaction are linked via the hash value of the blocks (maintained by a linked list with hash pointers) such that no one can tamper the blocks once the blocks are inserted into the chain. If there is any tamper incidents on the blocks, it will cause different hash value which breaks the link.

The consensus mechanism is an important step towards verifying and adding the blocks by the peer nodes in a P2P network. Consensus mechanism is an algorithm which helps the miners to validate a transaction and come to the conclusion of adding or dropping a block in the blockchain. It ensures a tamper free environment where one version of the truth should be agreed upon. It solves the problem of trust in blockchain, as all the non trusted miners participating in the process undergo a similar algorithm to agree on the validity of the block. Consensus algorithm also mitigates the effect of presence of faulty nodes in the network. Several consensus mechanisms are available in blockchain, such as *Proof of Work (PoW)* [41], *Proof of Stake (PoS)* [42,43], *Proof of Credit (PoC)* [44], *Alternative to PoW (Alt-PoW)* [45], *Proof of Authentication (PoAh)* [46], *Proof of Elapsed Time (PoET)* [47], *Proof of Space* [48], *Byzantine Fault Tolerance (BFT)* [49], *Practical Byzantine Fault Tolerance (PBFT)* [50,51], etc.

Blockchains are categorized into three types: (1) public blockchain, (2) private blockchain and (3) consortium blockchain. *Public blockchain*, also known as permission-less blockchain, works in an open environment, such as Bitcoin, where anyone is permitted to join and write the shared blocks. In *private blockchain* like Hyperledger, multichain fabric

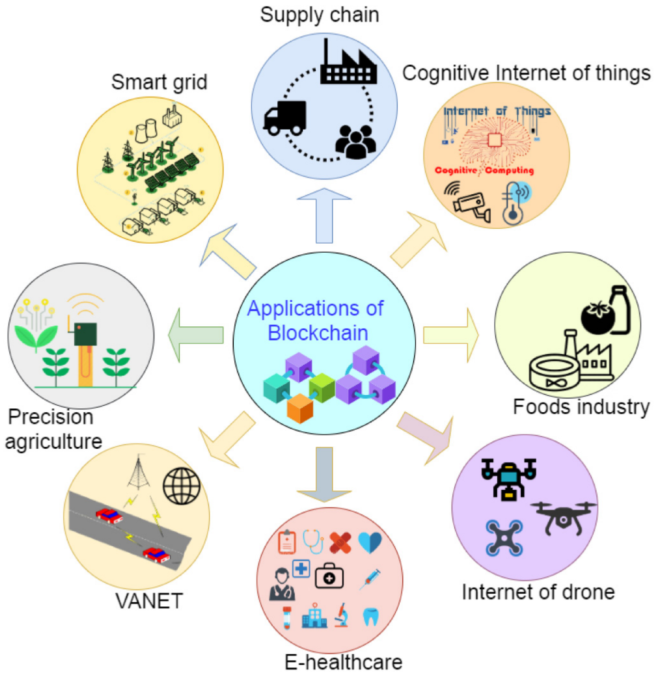


Fig. 3. Various applications of Blockchain of Things (BoT).

works in a closed environment where all the participants permitted in the process are well-known. Finally, in *consortium blockchain*, a hybrid approach is followed which combines both public and private blockchains to reach consensus in a P2P consortium blockchain.

In recent years, the researchers have shown that the blockchain technology embedded with IoT (specifically in CIoT context) serves various advantages, such as confidentiality, decentralization, transparency, immutability, etc. [52]. Fig. 3 shows different applications that are possible due to Blockchain of Things (BoT) (for examples, smart grid [53], supply chains [54], foods industry [55], Internet of Drones (IoD) [56], e-healthcare [57–59], Internet of Vehicles (IoV) [60], precision agriculture and smart farming [61–63], and CIoT [64]).

In 2017, Kotobi and Bilen [65] suggested a decentralized database using the blockchain technology in order to verify the spectrum sharing among cognitive radios. Their proposed medium access scheme is secure as well as decentralized in nature. In addition, their scheme performs in both moderate and severe fading situations for sharing available unexploited spectrum as compared to the traditional system. Their mechanism can be also applied to access available licensed spectrum without having constant spectrum sensing.

In 2021, Xie et al. [66] designed an active and passive reputation mechanism using blockchain technology for secure wide-band spectrum sensing, called APR_SWSS. In APR_SWSS, users' reputation can be categorized into two types: (1) "active sensing reputation" and (2) "passive sensing reputation". It was shown that APR_SWSS can further reduce the burden of sampling when it is combined with compressed sensing technology. Moreover, this mechanism can also find the illegitimate nodes based on the reasonable weight allocation of each node, and efficiently protect the "spectrum sensing data falsification (SSDF) attack" of the malicious users (MUs). Note that under SSDF attack, a malicious user can send the "false spectrum sensing results" to the fusion center, which in turn can significantly degrade the network performance parameters, like "detection accuracy" and "energy efficiency" [67].

3. System models

The purpose of this section is to discuss about network and threat models used in this paper.

3.1. Network model

Fig. 4 illustrates the overall network model without blockchain for CIoT, whereas the network model with blockchain-enabled CIoT is provided in Fig. 5. In the network models, a registration authority (RA) (also known as the trusted party) registers different IoT applications, such as moving vehicle, smart home, IoD, and so on, prior to functioning in their respective applications. The RA registers all other entities like gateway node which is associated with a particular CIoT application. Once the registration is over, the smart devices for an IoT application and the associated gateway can start the communication over public channel. In such circumstances, an IoT smart device sends the sensed data to its corresponding gateway node, and the gateway collects the data, makes a transaction and sends it to the cloud server. The cloud server forms a P2P cloud servers distributed network, where each cloud server replicates its database. The transactions are broadcasted to the entire cloud server network and all the peer nodes maintain their local transactions pool with these transactions. After that, a block is created whenever the pool reaches to a threshold value, and the block will be broadcasted to the cloud server network. After executing a consensus mechanism, the block is added into the blockchain. The data in the blockchain is used to offer the cognitive services based on AI/ML based Big data analytics (for example, CNN [68]). In addition, the decision making, cognitive selection, reasoning, and planning can made to provide the cognitive services. The actions or adaptations also form the cognitive services center that can send to the related IoT application via the gateway. After receiving the decisions, the IoT devices react accordingly.

3.2. Threat model

Data plays a major role in an CIoT environment. In CIoT system, multiple IoT applications are connected to each other, and for each IoT application there is a gateway node. The smart devices in an IoT application collects the real time data and send it to associated gateway over the public channel (insecure channel). Therefore, there exits always security threats for the communicating data. We accept two security threats: (1) "Dolev–Yao (DY) threat model" [69] and (2) "Canetti and Krawczyk's model (CK-adversary model)" [70], which are broadly-applied in this field. In the DY threat model, an adversary has a provision to access or intercept the transmitted messages from an insure channel and may also modify, delete, and inject some malicious contents in the transmission media. According to the CK-adversary model, the adversary not only can eavesdrop the communicated messages, modify, and delete some contents of the messages, but may also compromise a session. This results that the adversary can reveal the session keys and session states if the secret credentials are stored insecure memory of the communicated parties.

It is not always possible to monitor the deployed smart devices in CIoT applications for 24 × 7 time, and hence, there is a risk for a device physical capture by an adversary from such a hostile environment. By applying device physical capture attacks, the adversary can then extract the stored credentials from the insecure memory of the captured device through power analysis attacks [71]. Furthermore, it is assumed that the cloud servers are semi-trusted, whereas the gateways are also trusted. However, the gateways will be put under a physical locking system as it was the case in [72,73] in order to resist physical capture attack.

4. Related work

Wu et al. [32] designed a framework related to CIoT. It provides the fundamental cognitive activities. Next, the main enabling mechanisms involved in the cognitive tasks are discussed. In addition, the design of performance evaluation metrics along with research challenges are also discussed.

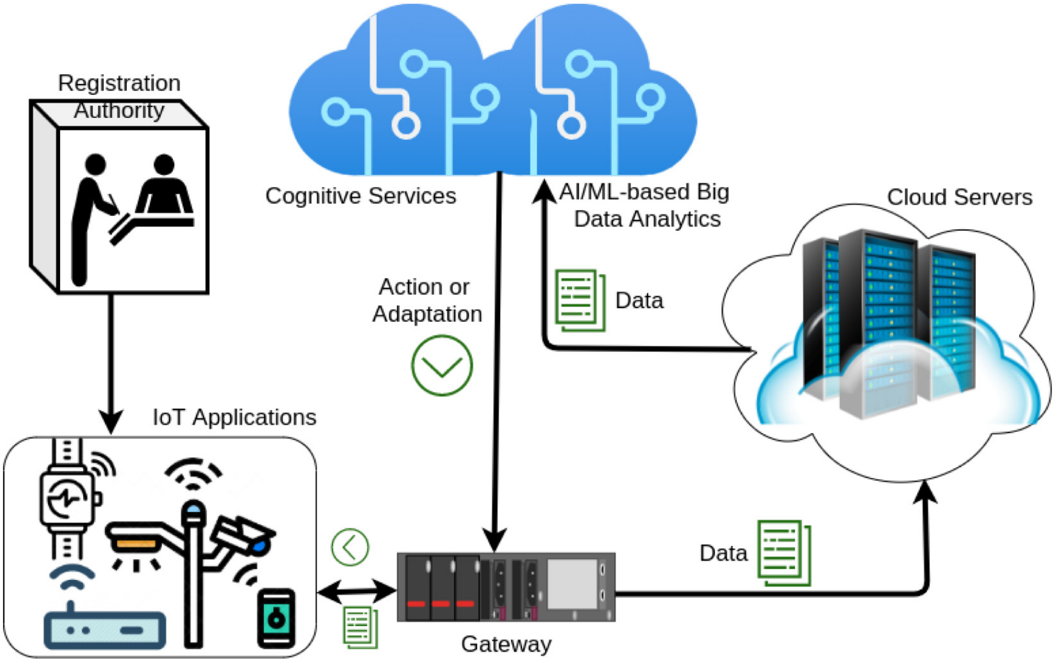


Fig. 4. CIoT network model without blockchain.

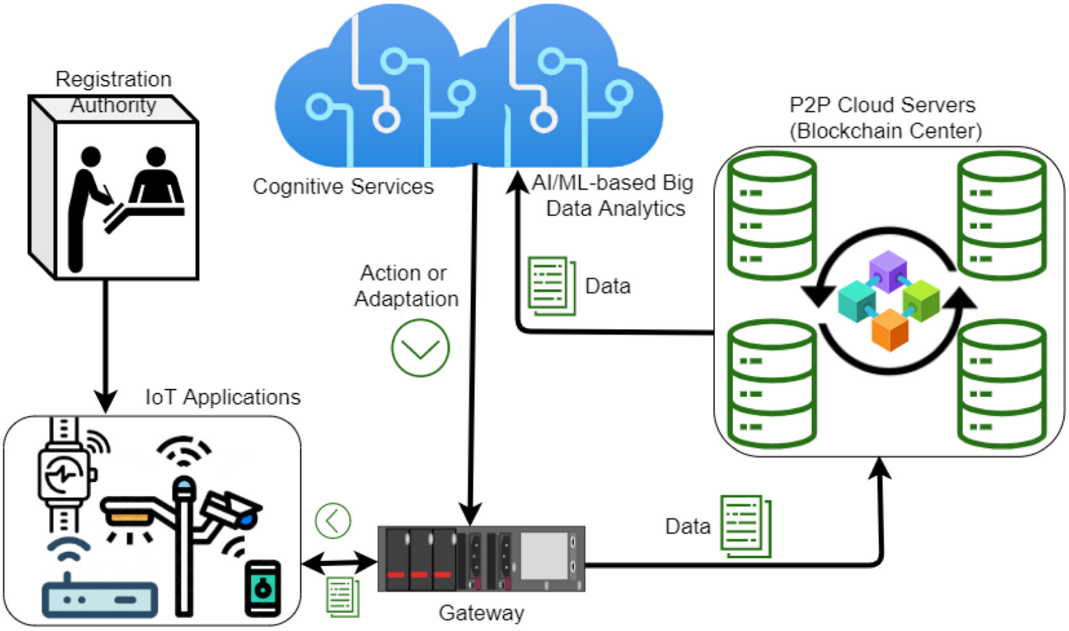


Fig. 5. CIoT network model with blockchain.

Awin et al. [74] presented a detailed survey by exploring the integration of emerging technologies with cognitive radio (CR)-based IoT. In addition, they also highlighted several emerging challenges and open issues in the domain of CIoT.

Li et al. [75] discussed the requirements for IoT that need to be equipped with "cognitive radio functionality". They discussed the spectrum-function associated with the cognitive technology in IoT, like "intelligent spectrum sensing", "dynamic spectrum access" and "efficient spectrum sharing".

Patnaik et al. [64] observed that "Spectrum Sharing Data Falsification (SSDF)" attacks may lead to degrade the of CR-based Internet of Battlefield Things (IoBT) networking environment (CR-IoBT). To mitigate such type of attack, they designed a "Proactive Blockchain based Spectrum Sharing (ProBLeSS)" approach which uses the blockchain technology in order to resist SSDF attacks in CR-IoBT environment.

Rathee et al. [76] argued that for proper management of the manufacturing settings, the cognitive systems play an important role for efficiently adapting their actions that are relied upon sensing data as well as uncertainty management. To enable such facilities, they suggested a decision-making mechanism in industrial informatics during the transmission of information, manufacturing process, and also for storing information via the simple "additive weighting and analytic hierarchy process".

Kasturi et al. [77] suggested a machine learning-based classification mechanism for various types of radio frequency jamming attacks. These jamming attacks can particularly interfere with genuine wireless signals by reducing the signal to-noise ratio. The ML model is also useful for detecting malicious URLs that are contained in emails and web pages. Johnson et al. [78] suggested such a model to combat with such malicious URLs detection using the machine learning techniques.

In recent years, the research on data poisoning attacks on the ML model has gained much popularity for correct predictions on the stored data [79–83]. The authors in [79] demonstrated how the label flipping attacks can be mounted on federated learning systems using data-sets and deep neural networks.

The authors in [80] investigated data poisoning attacks based on support vector machines, where the attacks are farmed on gradient ascent strategy. Sun et al. [81] designed an optimization framework that can calculate "optimal poisoning attacks" on a federated machine learning (FML) model. After that they derived an optimization mechanism by which they solved the optimal attack problem. They also showed that it can address various challenges in systems that are associated with the FML model. In a similar direction, the authors in [82] focused on the data poisoning attacks on the FML model.

The authors in [83] also mentioned and showed the data poisoning attacks based on auto regressive models. Furthermore, the authors in [84] showed the data poisoning attacks against matrix factorization based collaborative filtering.

5. Blockchain-based AI/ML-enabled Big data analytics security framework for CIoT

This section provides a blockchain-based AI/ML-enabled Big data analytics security framework for CIoT that is based on the network model shown in Fig. 5.

In the proposed framework, the registration authority (RA) (called the trusted party) takes the responsibility to register all the smart devices in the CIoT environment, and the gateway nodes attached with the smart devices depending on the applications. After successful registration process, the RA loads the secret credentials into their memories prior to their deployment in the CIoT networks.

Next, after deployment of the IoT devices and the gateway nodes, the devices can sense the real time data and send the data to its attached gateway. In order to send the data, the devices first establish a secure session key between them and also with the gateway node for secure communication using some robust access control mechanisms, for example, the scheme suggested in [85]. Note that according to the threat model discussed in Section 3.2, under the CK-adversary model, ephemeral secret leakage (ESL) attack needs to be protected. Thus, after successful mutual authentication among the entities in the network, the data is being sent securely to the gateway. As a result, the information is securely gathered by the gateway, and the in-charge gateway makes a transaction with the data, say Tx. Since the blockchain is considered here as *public*, the gateway creates a digital signature, say $ECDSA.Sig(Tx)$ on the generated transaction Tx using its own private key with the help of "elliptic curve digital signature generation algorithm $ECDSA.Sig(\cdot)$" [86].

Now, the gateway sends (Tx, $ECDSA.Sig(Tx)$) to the P2P cloud severs network. After receiving these signed transactions from the gateway, the blocks are created by a peer node. A typical block structure used in our public blockchain scenario is illustrated in Fig. 6, with the following components:

- **Block Header**: It contains the following fields:
  - *Block Version*: It is a unique serial positive number that identifies a block in the blockchain uniquely.
  - *Previous Block Hash (PBH)*: It represents hash value of the previous block. Thus, if a block is attempted to be modified by an attacker, it will not be successful as all the previous and future blocks hash values need to be updated.

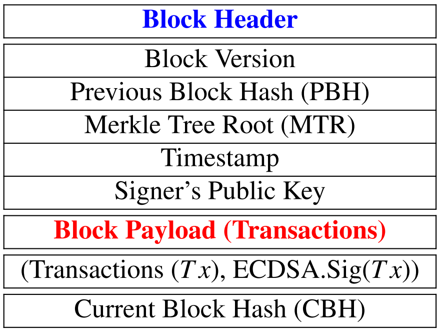


Fig. 6. A block structure in public blockchain.

  - *Merkle Tree Root (MTR)*: In 1979, Merkle [87] introduced a data structure, called the Merkle tree. Merkle trees are important data structures that are utilized to authenticate, with a unique digital signature along with a set of messages. Meanwhile, any verifier can check authenticity of a single message even without revealing other messages at the same time.
  - *Timestamp*: It denotes the time when a block was created by the block owner (here, a leader of the P2P cloud servers network).
  - *Signer's Public Key*: It is elliptic curve cryptography (ECC)-based public key of a signer (a gateway node for a particular CIoT application).
- **Block Payload (Transactions)**: It contains a list of transactions and their respective signatures of the form (Transactions (Tx), ECDSA.Sig(Tx)).
- **Current Block Hash (CBH)**: The hash of current block containing the hash of *Block Header* and *Block Payload (Transactions)*. The hash function is used here as the Secure Hash Algorithm (SHA-256) [88], which produces 256-bits message digest on an arbitrary length input string.

Once the blocks are created, we discuss in the following about the phases related to block creation, block addition into the public blockchain, and AI/ML-enabled Big data analytics.

5.1. Block creation phase

The gateway first sends the transactions with signatures (Transactions (Tx), ECDSA.Sig(Tx)) to the cloud servers network. Once the transactions are broadcasted to each peer node in the cloud servers network, it can be stored into the transactions pool which is replicated by the peer nodes. Whenever the transactions pool reaches to a pre-defined transaction threshold value, a block (as shown in Fig. 6) is created by a newly elected proposed or leader from the network by a round robin fashion.

5.2. Block addition phase in blockchain center

Addition of a block into the public blockchain is performed by executing a consensus mechanism. In this paper, we adopt a voting based consensus algorithm, known as "Practical Byzantine Fault Tolerance (PBFT)" [50]. Under this scenario, a proposer is elected and it then constructs a block, say $Block_i$ having a fixed number of transactions and their signatures, and also broadcasts the block $Block_i$ to the cloud servers network. The follower receives the block $Block_i$ and verify it with their own maintained local transactions pool. If all the transactions are verified successfully, and all other validating conditions, like Merkle tree root verification on the transactions containing in $Block_i$, signatures on transactions (Tx) and current block hash (CBH) are successful,

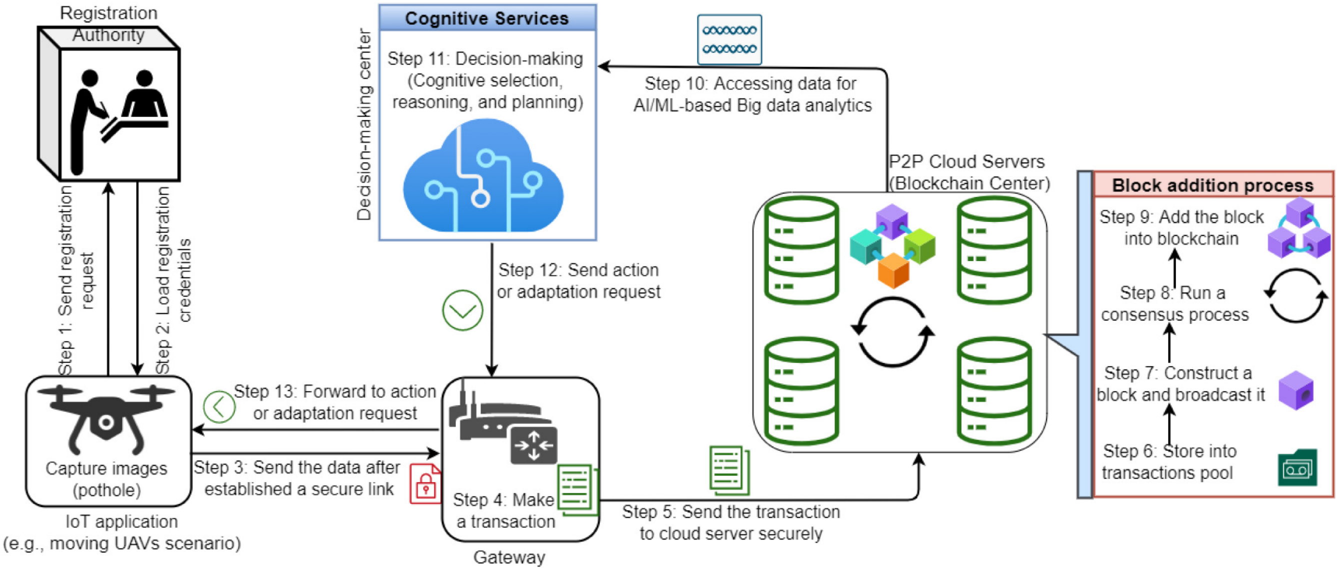

Registration Authority
Step 1: Send registration request
Step 2: Load registration credentials
Capture images (pothole)
IoT application (e.g., moving UAVs scenario)
Decision-making center
Cognitive Services
Step 11: Decision-making (Cognitive selection, reasoning, and planning)
Step 10: Accessing data for AI/ML-based Big data analytics
Step 12: Send action or adaptation request
Step 13: Forward to action or adaptation request
Step 3: Send the data after established a secure link
Step 4: Make a transaction
Gateway
Step 5: Send the transaction to cloud server securely
P2P Cloud Servers (Blockchain Center)
Block addition process
Step 9: Add the block into blockchain
Step 8: Run a consensus process
Step 7: Construct a block and broadcast it
Step 6: Store into transactions pool

Fig. 7. Overall process in blockchain-based AI/ML-enabled Big data analytics.

the block $Block_i$ can be added into the public blockchain. Once the block is added into the chain, the same process continues with the newly generated blocks by a newly selected other leader in the P2P cloud servers network.

5.3. Blockchain-based AI/ML-enabled Big data analytics phase

Once the data is stored into the public blockchain, we can use the AI/ML-based Big data analytics to offer a cognitive service. Therefore, the cognitive service can be provided based on the decision making, cognitive selection, reasoning, and planning. The decisions from the cognitive service center (also called an action or adaptation) is sent back to respective devices for the respective CIoT applications via the gateway node. Finally, the IoT devices can react accordingly to give better performance.

The overall process in blockchain-based AI/ML-enabled Big data analytics security framework for CIoT is shown in Fig. 7.

6. Experimental setup

In this section, we discuss about the experimental setup for showing the experimental results in Section 7.

The traditional data analysis has a huge difference with an IoT data analysis. The traditional data analysis is generally performed on a static dataset which is non-real time data. In this case, at first, the data is gathered and then stored in a database server. Next, we do the data analysis on the collected data. But, in case of IoT data analysis, it is real time data, because in an IoT network, the data is captured in real-time using the deployed IoT devices. So, in this case, the amount of data generated is huge and on the top of it, the data variation is also significant. As a result, we cannot simply store a huge amount of data in a centralized server. In addition, the IoT network usually uses the concept of edge computing, where the immediate server, that is directly connected to IoT devices, runs the data analysis task, which is called as edge computer. An edge computer often uses the federated machine learning model in order to perform the data analysis in a distributed computing environment. This technique involves the training of a machine learning algorithm across multiple different decentralized edge computers holding the training dataset that is local to them. This approach has a fundamental difference with the traditional centralized machine learning techniques where the data is usually stored in one server [89].

6.1. Data set

For our experiments, we select the pothole image data set [90] that are captured from the IoT-enabled drones to detect the pothole on the roads as shown in Fig. 8. The data set contains 681 images of potholes and normal roads. All the images in the data set are the colored images.

The training data for the machine learning purpose is collected from the images of roads and potholes that are available in [90]. This dataset has two directories: a) one is for the potholes and b) another is for the roads. The images are collected from various angles and also from various IoT smart devices, like drones and smart cameras. We have used this dataset for our experiment, because the dataset contains a good variation of angles of pictures and other image properties, like scale, brightness and contrast. This makes our predictions more robust and reliable during the experiments.

We have applied 75% of the data for training purpose and the remaining 25% of the data for the testing purpose. For demonstrating the effect of the data poison attacks, we have injected either noise on some part of the data (i.e., percentage noise injecting attacks) or alter the label of the data (i.e., label alteration attacks).

6.2. CNN architecture

We use the widely-recognized Convolution Neural Network (CNN) algorithms [68] for training our machine learning model. We consider five convolution layers and two dense layers. Every convolution layer is followed by a max-pooling layer. For each convolution layer, we use a rectified linear unit as the activation function. Obviously, in the last dense layer, we use softmax as activation function. This particular architecture of CNN is best suited for our experiments, and we show in Section 7 that it achieves 87.65% test accuracy without poisoning the data.

6.3. Machine learning model training setup

We implement our CNN network using the python language and keras library functions. We use 30 epochs to train the machine learning model and applied a batch size of 43 images. This is the best configuration for our experimental setup as we observe that 30 epoch is more than sufficient for the machine learning algorithm to converge.

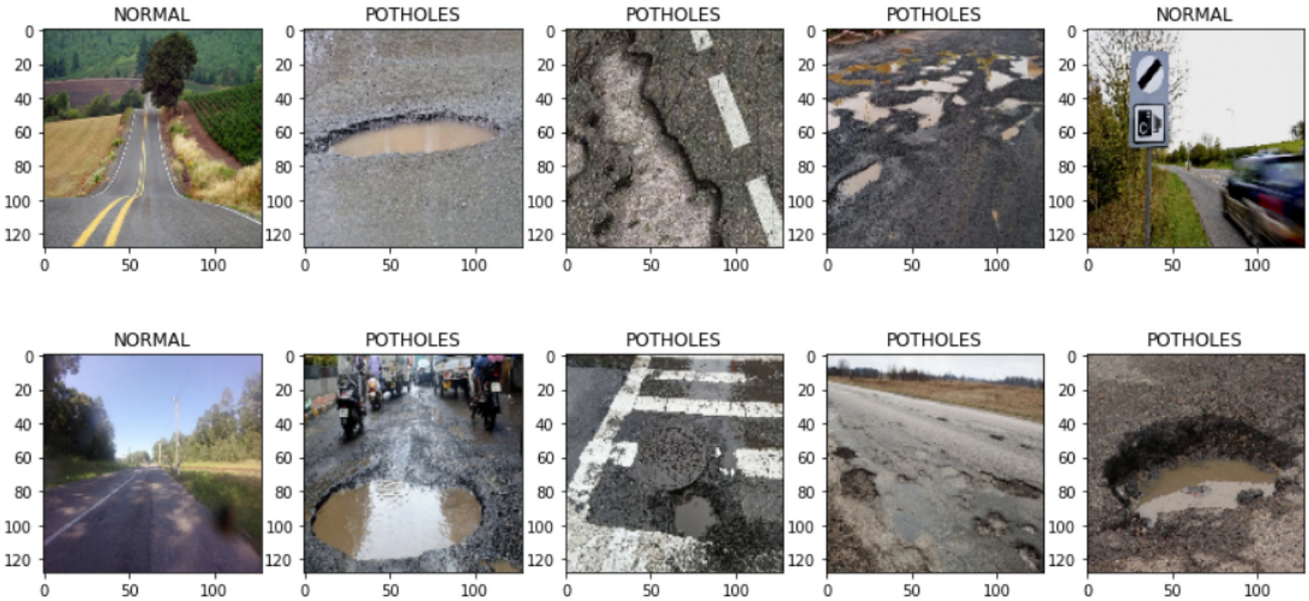


Fig. 8. Pothole data sets.

6.4. Noise injection process

As part of data poisoning attacks, we consider noise injecting attack by an adversary. To show the effect of the data poisoning attacks, we use four popular noise functions: slat noise, Poisson noise, Gaussian noise and salt-and-pepper (s&p) noise. We believe that the effect of other noises will be very similar to the effect of these four standard noises. In the experiment, we introduce a percentage noise on the original data set and then shown how a little noise may affect the performance of the machine learning model.

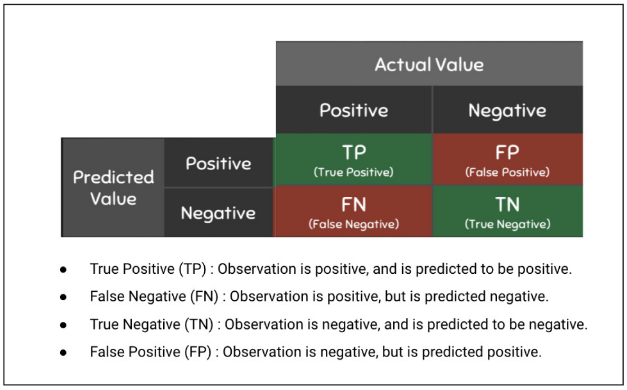
For one round of experiment, we first fix the percentage of the data that is needed to be corrupted (say, y%) and then fetch that percent of the data ($|D| \times y\%$) randomly from the data set, where $|D|$ represents the size of the data set, say D. Next, we modify the original image data to a noisy image data using four standard noise functions, one at a time. Finally, we compare the performance of the machine learning model trained on original data with machine learning model trained with corrupted data under the noise injecting attack.

6.5. Label flipping process

As part of data poisoning attacks, we also consider the label flipping attack by the adversary. To simulate the label flipping attack, we first fix the percentage of the data for one round of experiment that needed to be corrupted (y%) and then fetch that percent of the data ($|D| \times y\%$) randomly from the data set D. After that we modify the label of original images. Finally, we compare the performance of the machine learning model trained on original data with machine learning model trained with corrupted data under the label flipping attack.

6.6. Metrics used in experiment

For describing the performance of the machine learning model (with or without data poisoning attacks), we use the following four different metrics that are briefly described below. It is worth noticing that evaluation of the performance of a classification (ML) model relies on the counts of test records that are accurately and inaccurately projected by the model. The confusion matrix is a kind of matrix that offers a more intuitive observation about the performance of a predictive model [91]. In addition, the confusion matrix also provides which classes are predicted accurately and inaccurately, and what kind of errors are made. The confusion matrix can be pictorially represented as shown in Fig. 9, where TP, TN, FP and FN represent the number of "true positives", the number of "true negatives", the number of "false positives", and the number of "false negatives", respectively.



		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Fig. 9. Structure of a confusion matrix [91].

- *Accuracy*: Accuracy is one of the performance metrics, which depicts the accuracy of the model. It is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- *Recall*: Recall is another metric that depicts an actual positive rate. It gives a comparison among the number of positives that the model claims and the actual number of positives that are present in the data. It is defined as follows

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

- *Precision*: Precision is a very useful metric in order to show the performance of the machine learning model, which gives the positive predictive value. It provides a comparison between the number of true positives that the model claims and the total number of positives that the model claims. It is defined by the following relation:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

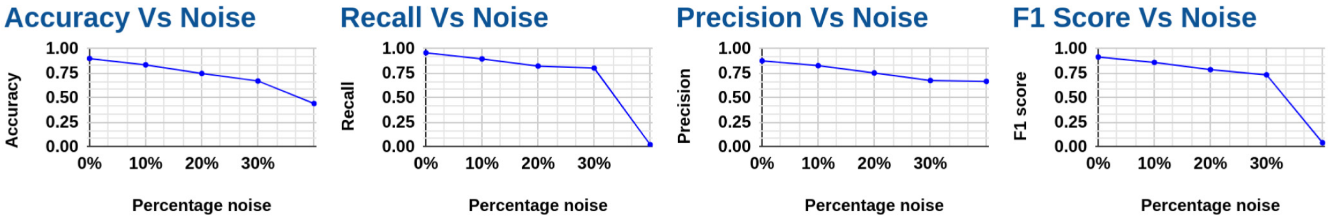


Fig. 10. Experimental results under salt noise insertion attacks.

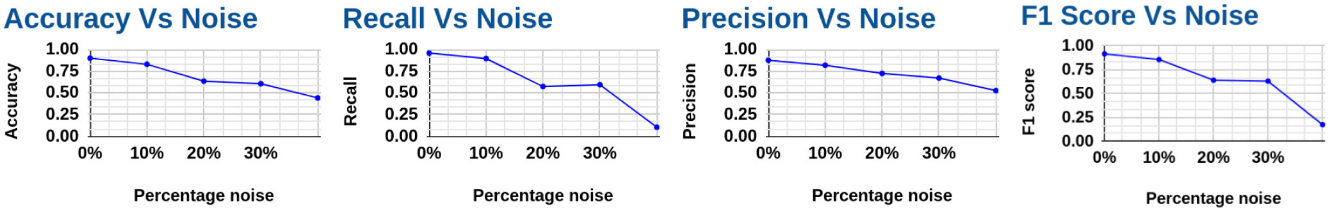


Fig. 11. Experimental results under Gaussian noise insertion attacks.

- *F1 score*: F1 score is another important metric which is considered as the weighted average of precision and recall. It is then defined by the following relation:

$$\text{F1 score} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

7. Experimental results

In this section, we provide a comprehensive experimental results under two circumstances: (1) performance of ML model under data poisoning attacks under the network model as shown in Fig. 4 and (2) performance of ML model without data poisoning attacks under the network model as shown in Fig. 5. In the first case, we show how the data poison attacks can effect the ML model when the data is on some cloud storage (i.e. not into the blockchains), whereas in the second case we show the effect when the data is in the blockchains (i.e., without data poisoning attacks).

Under a deep metric learning, if we are given two images, say y_i and y_j, the learned features are defined as $af_i = mlf(\theta; y_i)$, known as the anchor feature, and $cf_j = mlf(\theta; y_j)$, known as the compared feature [92]. Here, mlf and θ represent the "metric learning function" and the "learned parameters", respectively. Now, if the af_i is taken as signal, and the cf_j is considered as noisy signal, the noise n_{ij} in terms of af_i and cf_j can be defined as $n_{ij} = cf_j - af_i$. In traditional statistical theory, the ratio of signal variance to noise variance (SNR) is defined between af_i and cf_j as follows:

$$\begin{aligned} SNR_{ij} &= \frac{var(af_i)}{var(cf_j - af_i)} \\ &= \frac{var(af_i)}{var(n_{ij})} \end{aligned} \tag{5}$$

where $var(x) = \frac{\sum_{i=1}^{n}(x_i - m)^2}{n}$ is the variance of x, m is the mean of x and n is the number of data points in x. The significance of the signal variance is to measure the useful information. On the other side, the noise variance is applied to measure the useless information.

In our experiments, we have used the percentage noise, not SNR. This is because SNR is mostly applicable to the continuous data and the SNR ratio acts like a constant value. This implies that if we increase the signal strength, the noise will be also increased in the same proportion. In case of percentage noise, the ratio of noise to signal (or data) is not constant. For this reason, we can set (or increase) the amount of noise according to our requirements.

7.1. Performance of ML model under data poisoning attacks

In this section, we consider a case where the data is stored in some cloud severs, that is, the data is not stored into the blockchain. It allows an adversary can easily launch data poisoning attacks by either injecting some noise on the data stored on the cloud server(s) or changing the label of the data. In case of noise injecting attacks, the adversary can use some noise function on the original data in order to corrupt it.

7.1.1. Salt noise insertion attack

Salt noise is a very common type noise seen in the images. It is also known as impulsive noise. In this part of our experiment, we consider that if an adversary tries to inject the salt noise on the data, it effects the machine learning model significantly as demonstrated in Fig. 10. It is worth noting that 0% noise indicates that no attack on the data as the data will be stored into the blockchain in that case. It is evident from Fig. 10 when the percentage of noise is increased on the data, it leads to degrade the performance in terms of the measurement parameters (accuracy, recall, precision and F1 score) of the ML model significantly.

7.1.2. Gaussian noise insertion attack

Gaussian noise is another standard noise that has used in our experiments. It is referred as a statistical noise having a probability density function equal to that of the Gaussian distribution. We have shown in Fig. 11 that how Gaussian noise insertion attack can degrade the overall performance of the model. The experimental results demonstrate the effect of accuracy, recall, precision and F1 score under the ML model and it is seen that the performance of these parameters degrade when the percentage of noise is increased on the data.

7.1.3. Poisson noise insertion attack

Poisson noise is a standard noise that is used in the experimental results as demonstrated in Fig. 12 to check the effect of accuracy, recall, precision and F1 score under the ML model. Note that Poisson noise is a type of statistical noise which can be modeled by a group of Poisson processes. Similar to the observations in cases of salt noise insertion attack and Gaussian noise insertion attack, the accuracy, recall, precision and F1 score parameters under the ML model also degrade when the percentage of noise is inserted more on the data.

7.1.4. Label flipping attack

In a label filliping attack, Fig. 13 shows that if the adversary tries to alter the labels of the original data, how it can effect on the

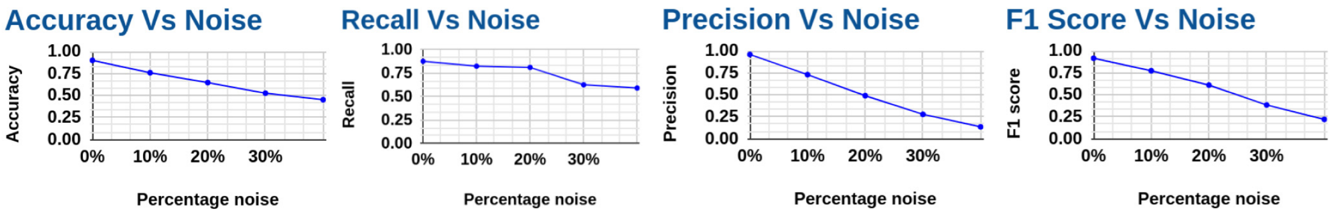


Fig. 12. Experimental results under Poisson noise insertion attacks.

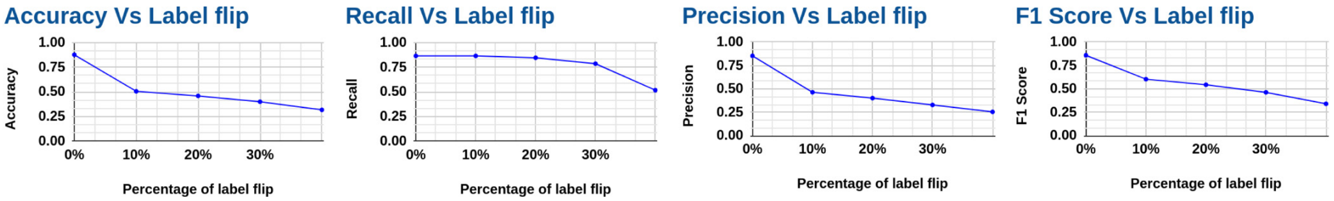


Fig. 13. Experimental results under label flipping attacks.

Table 1
Experimental results without any data poisoning attacks.

Accuracy	Recall	Precision	F1 Score
0.8764	0.8648	0.8533	0.859

overall performance of the ML model. From this figure, it is imperative that likewise other cases, the accuracy, recall, precision and F1 score parameters under the ML model also degrade once the percentage of noise is increased on the data.

7.2. Performance of ML model without data poisoning attacks

In this section, we see the effect of no data poising attacks under the ML model. 0% noise insertion means that no attacks on the data. In our proposed security framework (see Fig. 7), the data in terms of transactions along with their signatures is stored into the blockchain. A block is verified by three levels process: a) checking the Merkle tree root (MTR) stored in the block with the re-computed MTR on the transactions present on the block, b) checking the individual transaction's signature using the ECDSA signature verification algorithm, $ECDSA.ver(\cdot)$ [86], with the help of public key of the signer of the transactions (in this case, a gateway node), and c) checking the current block hash (CBH) on the entire block. Thus, if the data (transaction) is kept into the blockchain, the adversary cannot modify or alter the data due to immutability property of the blockchain. We now summarize the effect on the accuracy, recall, precision and F1 score parameters when no data poisoning attacks are performed, and the results are provided in Table 1. As compared to experimental results discussed in Sections 7.1.1 and 7.1.4, it is clear that when there are no data poisoning attacks on the data, the ML model has better performance in terms of accuracy, recall, precision and F1 score.

7.3. Important observations

We have provided a comparative study between storing and running the data analysis in the could servers (edge servers) without the blockchain (see Fig. 4) *versus* storing the data in the cloud servers into the blockchain (see Fig. 5), and then performed the data analysis on these for analyzing the impact on data poisoning attacks by a privileged insider attacker inside the cloud servers. It is worth noticing that we have used the concept of federated machine learning without the blockchain technology while doing the data analysis on the could servers (edge servers). However, in our case, it is performed on the blockchain data because the data residing in the blocks into the blockchain is tamper-proof.

8. Blockchain implementation

This section describes the implementation of the blockchain system in the proposed security framework that is discussed in Section 5. The sole purpose of the blockchain simulation is to measure the computational time needed for a varied number of mined blocks into the blockchain and also for a varied number of transactions present in each block. This gives the performance measurement of the blockchain system in the proposed security framework through the blockchain simulation study.

We created a blockchain system with the help of node.js script. We also created the virtual distributed blockchain network with twelve P2P (cloud) servers. All the servers work on the localhost, but they use different ports for communication among them. The servers run the consensus based on their roles by means of message passing mechanism. When the size of the transactions pool reaches to a pre-defined transactions threshold, one of the P2P servers (i.e., proposer) fetches the transactions from the pool and forms a block. After that, the proposer proposes the block to other P2P servers. The servers then run the voting-based PBFT consensus algorithmic steps to reach a consensus result, and finally the proposer adds the block to the blockchain. In the node.js script, we have written the code to record the block addition time in a text file, and then plotted the recorded time to show the performance of the blockchain under various different conditions.

To simulate the blockchain system, we have used a host computer having the configuration as: "OS: Ubuntu 18.04 LTS, Processor: Intel i5-8400 (2.80 GHz), Memory: 7.6 GiB, OS Type: 64 bit, Disk Type: HDD, Disk Size: 152.6 GB". We have considered the following different cases:

- **Case 1.** In this case, we fixed the number of transactions to a fixed value as 34 (i.e., the size of each block will be the same) and recorded the time of the blocks addition when we varied the number of blocks in the blockchain. The simulation results in Fig. 14 show that when the size of each block is fixed, the total computation time linearly varies with the number of blocks mined in the blockchain. Here, the total computation time (in seconds) refers to the time needed to mine a number of blocks into the chain including blocks verification time and mining time with the voting-based PBFT consensus execution time for a fixed number of transactions per block.
- **Case 2.** In this case, we fixed the total number of blocks in the blockchain to a fixed value as 36, and recorded the time of the blocks addition when we varied the size of the blocks (i.e., we varied the number of transactions in a block). The simulation results in Fig. 15 illustrate that when the number of blocks into the blockchain is fixed, the total computation time varies linearly

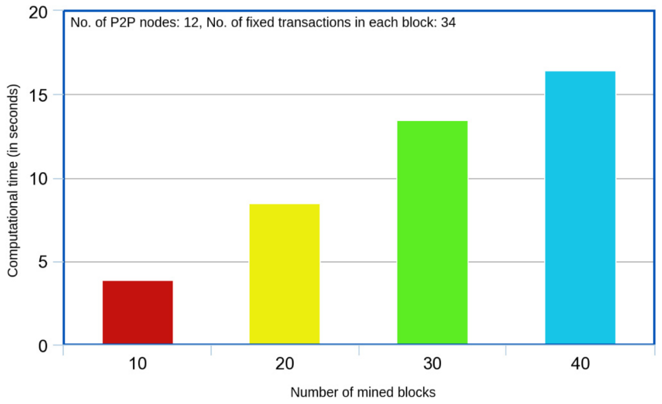


Fig. 14. Blockchain simulation outcomes in Case 1.

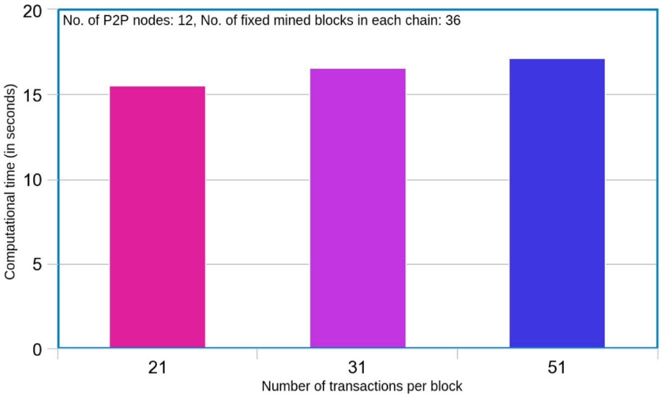


Fig. 15. Blockchain simulation outcomes in Case 2.

with the number of transactions in each block in the blockchain. Here, the total computation time (in seconds) refers to the time required to mine a number of blocks into the chain including blocks verification time and mining time with the voting-based PBFT consensus execution time for a fixed number of blocks per chain.

9. Conclusion

This paper provides an insightful observation on the impact on a blockchain-based AI/ML-enabled Big data analytics for CIoT environment. We first discussed the proposed security framework based on the defined network and threat models. We discussed how the blocks are created and then the blocks are added into the blockchain via voting-based PBFT consensus algorithm. Next, we provided an extensive experimental study on the effect of various data poising attacks, namely salt noise insertion attack, Gaussian noise insertion attack, Poisson noise insertion attack and label flipping attack. We then observed that when there are no data poisoning attacks on the data put into the blockchain, the ML model has significantly better performance in terms of accuracy, recall, precision and F1 score as compared to that for the case when there are possibilities of data poisoning attacks on the data put in non-blockchain storage platform. Finally, the blockchain simulation results that when the number of blocks into the blockchain is fixed, the total computation time varies linearly with the size of blocks mined in the blockchain as well as the number of transactions in each block in the blockchain. In future, we would like to conduct a detailed study to see the impact on blockchain-based AI/ML-enabled Big data analytics for CIoT where the external attacks can also influence ML parameters and drop accuracy.

CRediT authorship contribution statement

Ankush Mitra: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Basudeb Bera:** Conceptualization, Methodology, Software, Writing – original draft. **Ashok Kumar Das:** Conceptualization, Methodology, Writing – review & editing, Visualization, Supervision, Project administration. **Sajjad Shaukat Jamal:** Visualization, Project administration. **Ilsun You:** Conceptualization, Writing – review & editing, Visualization, Supervision, Funding acquisition.

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.

Acknowledgments

The authors thank the anonymous reviewers and the associate editor for their valuable feedback on the paper, which helped us to improve its quality and presentation. This work is supported by the Unified Blockchain Framework (UBF) project funded by the Ministry of Electronics & Information Technology, New Delhi, Government of India, under the Grant No. 4(4)/2021-ITEA. The authors extend their gratitude to the Deanship of Scientific Research at King Khalid University, Saudi Arabia for funding this work through research groups program under grant number R.G.P. 2/86/43.

References

[1] A.K. Das, S. Zeadally, D. He, Taxonomy and analysis of security protocols for Internet of Things, Future Gener. Comput. Syst. 89 (2018) 110–125.
[2] G. Glissa, A. Meddeb, 6LowPSec: An end-to-end security protocol for 6LoWPAN, Ad Hoc Netw. 82 (2019) 100–112.
[3] M. Wazid, A.K. Das, S. Shetty, P. Gope, J.J.P.C. Rodrigues, Security in 5G-Enabled Internet of Things Communication: Issues, Challenges, and Future Research Roadmap, IEEE Access 9 (2021) 4466–4489.
[4] M. Wazid, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, Secure Three-Factor User Authentication Scheme for Renewable-Energy-Based Smart Grid Environment, IEEE Trans. Ind. Inform. 13 (6) (2017) 3144–3153.
[5] M. Wazid, A.K. Das, M.K. Khan, A.A. Al-Ghaiheb, N. Kumar, A.V. Vasilakos, Secure Authentication Scheme for Medicine Anti-Counterfeiting System in IoT Environment, IEEE Internet Things J. 4 (5) (2017) 1634–1646.
[6] M. Wazid, A.K. Das, N. Kumar, V. Odelu, A. Goutham Reddy, K. Park, Y. Park, Design of Lightweight Authentication and Key Agreement Protocol for Vehicular Ad Hoc Networks, IEEE Access 5 (2017) 14966–14980.
[7] A.K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, X. Huang, Provably secure user authentication and key agreement scheme for wireless sensor networks, Secur. Commun. Netw. 9 (16) (2016) 3670–3687.
[8] A.K. Das, A.K. Sutrala, S. Kumari, V. Odelu, M. Wazid, X. Li, An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks, Secur. Commun. Netw. 9 (13) (2016) 2070–2092.
[9] S. Challa, A.K. Das, P. Gope, N. Kumar, F. Wu, A.V. Vasilakos, Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems, Future Gener. Comput. Syst. 108 (2020) 1267–1286.
[10] M. Wazid, A.K. Das, V. Bhat K, A.V. Vasilakos, LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment, J. Netw. Comput. Appl. 150 (2020) 102496.
[11] M. Wazid, P. Bagga, A.K. Das, S. Shetty, J.J.P.C. Rodrigues, Y. Park, AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment, IEEE Internet Things J. 6 (5) (2019) 8804–8817.
[12] S. Ruj, B. Roy, Key Predistribution Using Combinatorial Designs for Grid-group Deployment Scheme in Wireless Sensor Networks, ACM Trans. Sensor Netw. 6 (1) (2010) 4:1–4:28.
[13] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly-Secure Key Distribution for Dynamic Conferences, in: 12th Annual International Cryptology Conference - Advances in Cryptology (CRYPTO'92), in: Lecture Notes in Computer Science, vol. 740, Santa Barbara, California, 1993, pp. 471–486.
[14] D. Liu, P. Ning, R. Li, Establishing Pairwise Keys in Distributed Sensor Networks, ACM Trans. Inf. Syst. Secur. 8 (1) (2005) 41–77.
[15] D. Liu, P. Ning, Improving key pre-distribution with deployment knowledge in static sensor networks, ACM Trans. Sensor Netw. 1 (2) (2005) 204–239.

[16] A.K. Das, ECPKS: An Improved Location-Aware Key Management Scheme in Static Sensor Networks, Int. J. Netw. Secur. 7 (3) (2008) 358–369.

[17] A.K. Das, A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks, Int. J. Inf. Secur. 11 (3) (2012) 189–211.

[18] S. Chatterjee, A.K. Das, J.K. Sing, An Enhanced Access Control Scheme in Wireless Sensor Networks, Ad Hoc Sensor Wirel. Netw. 21 (1–2) (2014) 121–149.

[19] S. Chatterjee, A.K. Das, An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks, Secur. Commun. Netw. 8 (9) (2015) 1752–1771.

[20] D. Mishra, A.K. Das, S. Mukhopadhyay, A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card, Peer-to-Peer Netw. Appl. 9 (1) (2016) 171–192.

[21] S. Malani, J. Srinivas, A.K. Das, K. Srinathan, M. Jo, Certificate-Based Anonymous Device Access Control Scheme for IoT Environment, IEEE Internet Things J. 6 (6) (2019) 9762–9773.

[22] A.K. Das, M. Wazid, A.R. Yannam, J.J.P.C. Rodrigues, Y. Park, Provably Secure ECC-Based Device Access Control and key Agreement Protocol for IoT Environment, IEEE Access 7 (2019) 55382–55397.

[23] S.A. Chaudhry, K. Yahya, F. Al-Turjman, M.H. Yang, A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems, IEEE Access 8 (2020) 139244–139254.

[24] Z. Ali, A. Ghani, I. Khan, S.A. Chaudhry, S.H. Islam, D. Giri, A robust authentication and access control protocol for securing wireless healthcare sensor networks, J. Inf. Secur. Appl. 52 (2020) 102502.

[25] S.A. Chaudhry, H. Alhakami, A. Baz, F. Al-Turjman, Securing Demand Response Management: A Certificate-Based Access Control in Smart Grid Edge Computing Infrastructure, IEEE Access 8 (2020) 101235–101243.

[26] M. Luo, Y. Luo, Y. Wan, Z. Wang, Secure and Efficient Access Control Scheme for Wireless Sensor Networks in the Cross-Domain Context of the IoT, Secur. Commun. Netw. (2018) 6140978, http://dx.doi.org/10.1155/2018/6140978.

[27] S. Lawa, R. Krishnan, Policy Review in Attribute Based Access Control: A Policy Machine Case Study, J. Internet Serv. Inf. Secur. 10 (2) (2020) 67–81.

[28] Y. Fu, Z. Yan, J. Cao, O. Koné, X. Cao, An Automata Based Intrusion Detection Method for Internet of Things, Mob. Inf. Syst. 2017 (2017) 1750637.

[29] S. Pundir, M. Wazid, D.P. Singh, A.K. Das, J.J.P.C. Rodrigues, Y. Park, Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges, IEEE Access 8 (2020) 3343–3363.

[30] A.J. Meera, M.V.V.P. Kantipudi, R. Aluvalu, Intrusion Detection System for the IoT: A Comprehensive Review, in: A. Abraham, M.A. Jabbar, S. Tiwari, I.M.S. Jesus (Eds.), 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR'19), Springer International Publishing, Cham, 2021, pp. 235–243.

[31] K. Sai Kiran, R.K. Devisetty, N.P. Kalyan, K. Mukundini, R. Karthi, Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques, Procedia Comput. Sci. 171 (2020) 2372–2379.

[32] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, K. Long, Cognitive Internet of Things: A New Paradigm Beyond Connection, IEEE Internet Things J. 1 (2) (2014) 129–143.

[33] X. Liu, X. Zhang, NOMA-Based Resource Allocation for Cluster-Based Cognitive Industrial Internet of Things, IEEE Trans. Ind. Inform. 16 (8) (2020) 5379–5388.

[34] X. Liu, X. Zhang, Rate and Energy Efficiency Improvements for 5G-Based IoT With Simultaneous Transfer, IEEE Internet Things J. 6 (4) (2019) 5971–5980.

[35] M.K. Gurucharan, Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network, 2020, https://www.upgrad.com/blog/basic-cnn-architecture/. Accessed on March 2021.

[36] T. Mitchell, Machine Learning, McGraw Hill International, 1997.

[37] K. Ren, T. Zheng, Z. Qin, X. Liu, Adversarial Attacks and Defenses in Deep Learning, Engineering 6 (3) (2020) 346–360.

[38] H. Huang, J. Mu, N.Z. Gong, Q. Li, B. Liu, M. Xu, Data Poisoning Attacks to Deep Learning Based Recommender Systems, 2021, CoRR abs/2101.02644. URL: https://arxiv.org/abs/2101.02644.

[39] X. Gong, Q. Wang, Y. Chen, W. Yang, X. Jiang, Model Extraction Attacks and Defenses on Cloud-Based Machine Learning Models, IEEE Commun. Mag. 58 (12) (2020) 83–89.

[40] M. Juuti, S. Szyller, S. Marchal, N. Asokan, PRADA: Protecting Against DNN Model Stealing Attacks, in: IEEE European Symposium on Security and Privacy (Euro S&P), Stockholm, Sweden, 2019, pp. 512–527.

[41] S. Nakamoto, Bitcoin open source implementation of P2P currency, P2P Found. 18 (2009).

[42] H. Dai, Z. Zheng, Y. Zhang, Blockchain for Internet of Things: A Survey, IEEE Internet Things J. 6 (5) (2019) 8076–8094.

[43] S. King, S. Nadal, PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake, 2012, Accessed on March 2021. https://decred.org/research/king2012.pdf.

[44] X. Han, Y. Yuan, F. Wang, A Fair Blockchain Based on Proof of Credit, IEEE Trans. Comput. Soc. Syst. 6 (5) (2019) 922–931.

[45] S. Sharkey, H. Tewari, Alt-PoW: An Alternative Proof-of-Work Mechanism, in: IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), Newark, CA, USA, 2019, pp. 11–18.

[46] D. Puthal, S.P. Mohanty, Proof of Authentication: IoT-Friendly Blockchains, IEEE Potentials 38 (1) (2019) 26–29.

[47] M.A. Kumar, V. Radhesyam, B. SrinivasaRao, Front-End IoT Application for the Bitcoin based on Proof of Elapsed Time (PoET), in: Third International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2019, pp. 646–649.

[48] S. Park, A. Kwon, G. Fuchsbauer, P. Gaži, J. Alwen, K. Pietrzak, SpaceMint: A Cryptocurrency Based on Proofs of Space, in: S. Meiklejohn, K. Sako (Eds.), Financial Cryptography and Data Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, pp. 480–499.

[49] G.S. Veronese, M. Correia, A.N. Bessani, L.C. Lung, P. Verissimo, Efficient Byzantine Fault-Tolerance, IEEE Trans. Comput. 62 (1) (2013) 16–30.

[50] M. Castro, B. Liskov, Practical Byzantine Fault Tolerance and Proactive Recovery, ACM Trans. Comput. Syst. 20 (4) (2002) 398–461.

[51] L. Zhang, Q. Li, Research on Consensus Efficiency Based on Practical Byzantine Fault Tolerance, in: 10th International Conference on Modelling, Identification and Control (ICMIC), Guiyang, China, 2018, pp. 1–6.

[52] M. Alizadeh, K. Andersson, O. Schelen, A Survey of Secure Internet of Things in Relation to Blockchain, J. Internet Serv. Inf. Secur. 10 (3) (2020) 47–75.

[53] B. Bera, S. Saha, A.K. Das, A.V. Vasilakos, Designing Blockchain-Based Access Control Protocol in IoT-Enabled Smart-Grid System, IEEE Internet Things J. 8 (7) (2021) 5744–5761.

[54] S. Jangirala, A.K. Das, A.V. Vasilakos, Designing Secure Lightweight Blockchain-Enabled RFID-Based Authentication Protocol for Supply Chains in 5G Mobile Edge Computing Environment, IEEE Trans. Ind. Inform. 16 (11) (2020) 7081–7093.

[55] A. Rejeb, J.G. Keogh, S. Zailani, H. Treiblmaier, K. Rejeb, Blockchain Technology in the Food Industry: A Review of Potentials, Challenges and Future Research Directions, Logistics 4 (4) (2020) 1–26.

[56] B. Bera, D. Chattaraj, A.K. Das, Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment, Comput. Commun. 153 (2020) 229–249.

[57] Y.A. Shichkina, G.V. Kataeva, Y.A. Irishina, E.S. Stanevich, The use of mobile phones to monitor the status of patients with Parkinson's disease, J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 11 (2) (2020) 55–73.

[58] S. Saha, A.K. Sutrala, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, On the Design of Blockchain-Based Access Control Protocol for IoT-Enabled Healthcare Applications, in: IEEE International Conference on Communications (ICC'20), Dublin, Ireland, 2020, pp. 1–6.

[59] M. Wazid, B. Bera, A. Mitra, A.K. Das, R. Ali, Private Blockchain-Envisioned Security Framework for AI-Enabled IoT-Based Drone-Aided Healthcare Services, in: Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and beyond (DroneCom'20), London, United Kingdom, 2020, pp. 37–42.

[60] P. Bagga, A.K. Sutrala, A.K. Das, P. Vijayakumar, Blockchain-based batch authentication protocol for Internet of Vehicles, J. Syst. Archit. 113 (2021) 101877.

[61] A. Vangala, A.K. Das, N. Kumar, M. Alazab, Smart Secure Sensing for IoT-Based Agriculture: Blockchain Perspective, IEEE Sens. J. 21 (16) (2021) 17591–17607.

[62] P. Angin, M.H. Anisi, F. Goksel, C. Gursoy, A. Buyukgulcu, AgriLoRa: A digital twin framework for smart agriculture, J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 11 (4) (2020) 77–96.

[63] A. Vangala, A.K. Sutrala, A.K. Das, M. Jo, Smart Contract-Based Blockchain-Envisioned Authentication Scheme for Smart Farming, IEEE Internet Things J. 8 (13) (2021) 10792–10806.

[64] M. Patnaik, G. Prabhu, C. Rebeiro, V. Matyas, K. Veezhinathan, ProBLeSS: A Proactive Blockchain Based Spectrum Sharing Protocol Against SSDF Attacks in Cognitive Radio IoBT Networks, IEEE Netw. Lett. 2 (2) (2020) 67–70.

[65] K. Kotobi, S.G. Bilen, Blockchain-enabled spectrum access in cognitive radio networks, in: Wireless Telecommunications Symposium (WTS'17), Chicago, IL, USA, 2017, pp. 1–6.

[66] X. Xie, Z. Hu, M. Chen, Y. Zhao, Y. Bai, An Active and Passive Reputation Method for Secure Wideband Spectrum Sensing Based on Blockchain, Electronics 10 (11) (2021) 1–19.

[67] S. Althunibat, B.J. Denise, F. Granelli, A Punishment Policy for Spectrum Sensing Data Falsification Attackers in Cognitive Radio Networks, in: 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall), Vancouver, BC, Canada, 2014, pp. 1–5.

[68] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.

[69] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.

[70] R. Canetti, H. Krawczyk, Universally Composable Notions of Key Exchange and Secure Channels, in: International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02), Amsterdam, The Netherlands, 2002, pp. 337–351.

[71] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.

[72] E. Bertino, N. Shang, S.S. Wagstaff Jr., An Efficient Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting, IEEE Trans. Dependable Secure Comput. 5 (2) (2008) 65–70.

[73] M. Wazid, A.K. Das, V. Odelu, N. Kumar, W. Susilo, Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment, IEEE Trans. Dependable Secure Comput. 17 (2) (2020) 391–406.

[74] F.A. Awin, Y.M. Alginahi, E. Abdel-Raheem, K. Tepe, Technical Issues on Cognitive Radio-Based Internet of Things Systems: A Survey, IEEE Access 7 (2019) 97887–97908.

[75] F. Li, K. Lam, X. Li, Z. Sheng, J. Hua, L. Wang, Advances and Emerging Challenges in Cognitive Internet-of-Things, IEEE Trans. Ind. Inform. 16 (8) (2020) 5489–5496.

[76] G. Rathee, F. Ahmad, R. Iqbal, M. Mukherjee, Cognitive Automation for Smart Decision-Making in Industrial Internet of Things, IEEE Trans. Ind. Inform. 17 (3) (2021) 2152–2159.

[77] G. Kasturi, A. Jain, J. Singh, Detection and Classification of Radio Frequency Jamming Attacks using Machine learning, J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (JoWUA) 11 (4) (2020) 49–62.

[78] C. Johnson, B. Khadka, R.B. Basnet, T. Doleck, Towards Detecting and Classifying Malicious URLs Using Deep Learning, J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (JoWUA) 11 (4) (2020) 31–48.

[79] V. Tolpegin, S. Truex, M.E. Gursoy, L. Liu, Data Poisoning Attacks Against Federated Learning Systems, in: L. Chen, N. Li, K. Liang, S.A. Schneider (Eds.), 25th European Symposium on Research in Computer Security (ESORICS'20), in: Lecture Notes in Computer Science, vol. 12308, Springer, 2020, pp. 480–501.

[80] B. Biggio, B. Nelson, P. Laskov, Poisoning Attacks against Support Vector Machines, in: Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML'12), Edinburgh, Scotland, 2012, pp. 1467–1474.

[81] G. Sun, Y. Cong, J. Dong, Q. Wang, J. Liu, Data Poisoning Attacks on Federated Machine Learning, 2020, https://arxiv.org/abs/2004.10020.

[82] A.N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Model poisoning attacks in federated learning, in: Proceedings of 32nd Conference on Neural Information Processing Systems (NeurIPS 2018): Workshop on Security in Machine Learning (SecML), Montreal, Canada, 2018, pp. 1–23.

[83] S. Alfeld, X. Zhu, P. Barford, Data Poisoning Attacks against Autoregressive Models, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, 2016, pp. 1452–1458.

[84] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.R. Muller, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognit. 65 (2017) 211–222.

[85] S. Saha, D. Chattaraj, B. Bera, A.K. Das, Consortium blockchain-enabled access control mechanism in edge computing based generic Internet of Things environment, Trans. Emerg. Telecommun. Technol. 32 (6) (2021) 1–34.

[86] D. Johnson, A. Menezes, S. Vanstone, The Elliptic Curve Digital Signature Algorithm (ECDSA), Int. J. Inf. Secur. 1 (1) (2001) 36–63.

[87] R.C. Merkle, Secrecy, Authentication, and Public Key Systems (Ph.D. thesis), Stanford University, USA, 1979.

[88] W.E. May, Secure Hash Standard, 2015, URL: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995. Accessed on August 2019.

[89] G. Sun, Y. Cong, J. Dong, Q. Wang, J. Liu, Data Poisoning Attacks on Federated Machine Learning, 2020, CoRR abs/2004.10020. https://arxiv.org/abs/2004.10020.

[90] A. Kumar, Pothole Detection Dataset, 2020, Accessed on March 2021. https://www.kaggle.com/atulyakumar98/pothole-detection-dataset.

[91] C. Liu, More Performance Evaluation Metrics for Classification Problems You Should Know, 2020, Accessed on March 2021. https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html.

[92] T. Yuan, W. Deng, J. Tang, Y. Tang, B. Chen, Signal-To-Noise Ratio: A Robust Distance Metric for Deep Metric Learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19), Long Beach, CA, USA, 2019, pp. 4815–4824.

Ankush Mitra received his B.E. degree in Information Technology from Burdwan University, West Bengal, India. He also received his M.Tech. degree in computer science and information security from the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India. His research interests include network security, Internet of Things (IoT), AI/ML and blockchain technology.

Basudeb Bera received his M.Sc. degree in mathematics and computing in 2014 from IIT (ISM) Dhanbad, India, and M.Tech. degree in computer science and data processing in 2017 from IIT Kharagpur, India. He is currently pursuing his Ph.D. degree in computer science and engineering from the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India. His research interests are cryptography, network security and blockchain technology. He has published more than 30 papers in international journals and conferences in his research areas.

Ashok Kumar Das received a Ph.D. degree in computer science and engineering, an M.Tech. degree in computer science and data processing, and an M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, network and system security, security in vehicular ad hoc networks, smart grids, smart homes, Internet of Things (IoT), Internet of Drones, Internet of Vehicles, Cyber–Physical Systems (CPS) and cloud computing, intrusion detection, blockchain and AI/ML security. He has authored over 320 papers in international journals and conferences in the above areas, including over 275 reputed journal papers. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He was/is on the editorial board of IEEE Systems Journal, Journal of Network and Computer Applications (Elsevier), Computer Communications (Elsevier), Journal of Cloud Computing (Springer), Cyber Security and Applications (Elsevier), IET Communications, KSII Transactions on Internet and Information Systems, and International Journal of Internet Technology and Secured Transactions (Inderscience), and has served as a Program Committee Member in many international conferences. He also severed as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, International Conference on Applied Soft Computing and Communication Networks (ACN'20), October 2020, Chennai, India, and second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020. His Google Scholar h-index is 72 and i10-index is 204 with over 14,200 citations. He is a senior member of the IEEE.

Sajjad Shaukat Jamal received the Ph.D. degree in mathematics from Quaid-i-Azam University, Islamabad, Pakistan. He is currently working as an Assistant Professor with the Department of Mathematics, King Khalid University, Abha, Saudi Arabia. His research interests include number theory, cryptography, digital watermarking, steganography, information security, multimedia security, and blockchain technology. He has several journal papers in his research areas.

Ilsun You received the MS and Ph.D. degrees in computer science from Dankook University, Seoul, Korea, in 1997 and 2002, respectively. He received the second Ph.D. degree from Kyushu University, Japan, in 2012. Now, he is a full professor at Department of Financial Information Security as well as Department of Information Security, Cryptology, and Mathematics, Kookmin University. He has served or is currently serving as a Steering Chair, General Chair or a Program Chair of international conferences and symposiums such as MobiSec'16-21, WISA'19-20, ProvSec'18, ACM MIST'15-17 and so forth. Dr. YOU is the EiC of Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) and Journal of Internet Services and Information Security (JISIS) while serving as an associate EiC of Intelligent Automation & Soft Computing (IASC). He is in the Editorial Board for Information Sciences, International Journal of Intelligent Systems, IEEE Access, International Journal of Ad Hoc and Ubiquitous Computing, Computing and Informatics, and Journal of High Speed Networks. Especially, he has focused on 5/6G security, security for wireless networks & mobile internet, IoT/CPS security and so forth while publishing more than 180 papers in these areas. He is a Fellow of the IET and a Senior member of the IEEE.