Security provisions in smart edge computing devices using blockchain and machine learning algorithms: a novel approach

Kamta Nath Mishra¹ · Vandana Bhattacharjee¹ · Shashwat Saket¹ · Shivam Prakash Mishra¹

Received: 16 July 2022 / Revised: 21 October 2022 / Accepted: 5 November 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
It is difficult to manage massive amounts of data in an overlying environment with a single server. Therefore, it is necessary to comprehend the security provisions for erratic data in a dynamic environment. The authors are concerned about the security risk of vulnerable data in a Mobile Edge based distributive environment. As a result, edge computing appears to be an excellent perspective in which training can be done in an Edge-based environment. The combination of Edge computing and consensus approach of Blockchain in conjunction with machine learning techniques can further improve data security, mitigate the possibility of exposed data, and it reduces the risk of a data breach. As a result, the concept of federated learning provides a path for training the shared data. A dataset was collected that contained several vulnerable, exposed, recovered, and secured data and data security was precepted under the surveillance of two-factor authentication. This paper discusses the evolution of data and security flaws and their corresponding solutions in smart edge computing devices. The proposed model incorporates data security using consensus approach of Blockchain and machine learning techniques that include several classifiers and optimization techniques. Further, the authors applied the proposed algorithms in an edge computing environment by distributing several batches of data to different clients. As a result, the client privacy was maintained by using Blockchain servers. Furthermore, the authors segregated the client data into batches that were trained using the federated learning technique. The results obtained in this paper demonstrate the implementation of a Blockchain-based training model in an edge-based computing environment.

Keywords Blockchain technology · Edge computing · Federated learning systems · Machine learning techniques · Voting classifier

1 Introduction
The fourth major paradigm shift in modern computing is edge computing. It is processing data closer to the point where this is generated. With the digital revolution, it is now possible to reduce the latency of client–server architecture, particularly on a large scale. Edge computing is a distributed information technology architecture that processes client statistics at the network’s edge. It is based on the idea that instead of bringing data closer to the computer, we bring the computer closer to the data. Edge computing relocates some storage and compute resources away from the central data center and closer to the data source. The three pillars of edge computing, namely autonomy, edge device security, and data sovereignty, are among its most significant advantages [1, 2]. While the disadvantages of the edge computing could be summarized in its low redundancy, potential loss as well as data poisoning, longer outrage time. Mobile edge computing (MEC) devices are now widely used to solve artificial intelligence (AI) problems. Traditionally, AI requires a central server with powerful processing power for Machine Learning. This approach has several drawbacks, including
communication overheads (CO) and data privacy issues (DPI) [3].

Federated learning is a machine learning technique that trains algorithms across multiple devices (edge/mobile) without exchanging or accumulating them in a single server [4]. Unlike traditional machine learning, which requires data sets to be brought to a single location, federated learning promotes trust in machine learning (ML) processes by keeping the data at the source and directing the model toward it [5]. Each device has its own set of local data ready to be trained on. In a single epoch, the weights are calculated and returned to the MEC. The MEC fine-tunes the model before sending it out for iteration [6, 7]. This process is repeated until the desired level of accuracy is obtained. In the current Federated Learning system, users must trust the MEC system for model aggregation. What started out as a benefit may turn out to be a disadvantage. Figure 1 depicts the classical federated learning architecture.

The interconnection having Mobile Edge Computing (MEC) servers, aggregation and global data at one end and Internet of Things (IoT) devices, phones at other end are given in Fig. 1. It fosters the model training at local and global levels. While the model is being trained at some of the devices, it is simultaneously being tested at other devices. The model trained here is further improved at the global level with added noise and some improvements. This keeps the check and balance on the model accuracy. Further, it provides us with a faster and more accurate model. But along with the given advantages, it paves a way for high security risk of several attacks such as data poisoning attacks, susceptible data breaching attack, etc.

Blockchain aptly is a decentralized database. The transactions being managed by chain are the backbone of the technology. The blocks of data distributed are secured using the crypto-graphical methods. After a new block is merged into the chain, it is believed that the new block can interact the all the other blocks in the chain. Proof of Work (PoW) is one of the methods used for merging blocks by adding the hash function in the current block and then mines it. PoW provides a simple algorithm with a complete centralization as well as it procures a node free access. But along with the advantages, it is lucid that there is a waste of energy.

Transmission of learning parameters is highly vulnerable to attacks when travelling between nodes. The MEC server can be a single point of failure in an attack, and this model has scalability issues. Hence, a consensus approach of Blockchain based Federated Learning system may have following advantages over others [8, 9]:

\textit{Advantage}_1: It is using decentralized approach. Hence, it has the advantages in terms of security and privacy.

\textit{Advantage}_2: It does not allow a single point of failure attacks.

\textit{Advantage}_3: It is highly scalable. The biggest reason for slowed progress is the unavailability of data.

Differential privacy (DP) is a technique that protects user privacy by adding random noise to the data. It requires the output of a query to be the same when a user is removed from the original dataset [10]. The disadvantage with the differential privacy is that the picking of outliers such as max, sum can alter the meaning of the data and can provide random inferences. Along with this, the addition of the noise must be done at every level of queries leading towards the data breaches.

In local differential privacy (LDP), the noise is added directly to the data by the individual data owners. In the global differential privacy (GDP), the noise is added by the complete dataset owner to the output of a query on the database [11]. Figure 2 defines a schematic representation of local differential privacy and global differential privacy in association to the mobile edge computing servers.

The contributions of this paper are as follows:

i. Data security is established with susceptible data, recoverable data in an edge computing environment using two-level-architecture.

ii. Proper security provisions for several breached data as well as susceptible data using differential privacy using data stored in blockchain based storage.

iii. Addition of noise in the blocks of data securing anonymity of the data in the local differential privacy and further exaggerated in a global differential privacy.

iv. Implementation of machine learning techniques for training the data from individual blocks with ensemble techniques procuring a locally asymptotic stability and further training the model in a federated edge-based environment for obtaining the global asymptotic stability of the data.

v. Mitigate the risk of poisoning attack by segregating the data using consensus approach of blockchains. Further, differential privacy techniques are implemented in order to hinder the malicious user reaching out to the highly susceptible data.

2 Literature review

Throughout the spread of numerous IoT (Internet of Things) devices in the cloud environment, the authors’ encountered challenges in meeting the low latency and high bandwidth requirements, creating a void for decentralized computing, this later evolved into edge computing.
Furthermore, edge computing depicts several security threat itineraries. Another issue arises when the central server goes offline for an extended period. In these circumstances, the best approach is to implement the consensus approach of Blockchain, which can be visualized as an actual chain of blocks that operates on the principle of decentralization. It is clear from this that machine learning models like deep learning, deep reinforcement learning, and reinforcement learning are difficult to adapt to IoT and Edge environments. This opens the door to a plethora of potential risks and is one of the most important areas for research enthusiasts. Edge Computing devices are vulnerable in terms of software and resources, allowing them to integrate with a variety of machine learning techniques.

This section will concentrate on several works that have been completed in the respective domain.

Nowadays, Blockchain has been homogenized into IoT and cloud environments; Sharma et al. [12] proposed Software-Defined Network (SDN) architecture blended with Blockchain, which is divided into three layers. The first layer is the device layer, which collects data from the devices. The raw data was processed by the second layer, the fog layer, with the help of the SDN controller. Furthermore, the third layer contains all of the received processed data. As a result, they collected, classified, and analyzed the streaming data obtained from IoT. They reduced the cost of end-to-end delay by bringing computational resources to the edge of IoT devices using Pan et al. [13] proposed a work demonstrating the EdgeChain
framework for ensuring Blockchain collaboration in smart contracts. Edge devices collected data that IoT devices could access. Edge Chain’s core concept is to combine Blockchains with a coin system, blend them under the IoT framework, and merge the resource pool of edge clouds. They proctored all transactions through Blockchains for security auditing and logging without overburdening the devices. Rahman et al. [14] proposed a Blockchain-based infrastructure that guarantees privacy and security-based spatial–temporal contract services. This infrastructure anchors several fog nodes at the host’s edge and integrates them with computational intelligence methods. Further, they proposed a Mobile Edge Computing-based sharing economy system through the off-chain framework and Blockchain implementation to contain immutable ledgers.

Xu et al. [15] discussed how edge computing could meet low latency, high security, and data privacy requirements. With the channel of the short distance between the edge computing devices and the power terminals, low delay is ensured while collecting the gigantic data. They implemented a deep learning mechanism to enchant the real-time data flow using examples such as load balancing in the intelligent energy system, power prices predictions, and many more. The network architecture used through the itineraries of edge computing following the peculiarities of security proved to be much more effective than the cloud computing frameworks implemented beforehand. Still, the security of each node that arises in the edge computing framework makes it strenuous to adept in general use. Thus, to overcome this, they proposed a security architecture named as the physical layer security that involved only end-to-end users constrained by the limits of resources and energy. This paper effectuated many artificial intelligence prototypes for achieving the required efficiency and security under the chain of intelligent grids associated with edge computing.

It is significant to fortify the working of the critical infrastructural system in a society and its economy. The volume of data generated due to expanding IoT devices is turning out to be a hot topic in terms of scalability and security factors. Wu et al. [16] introduced the IoT/IIoT (Industrial Internet of Things) based critical infrastructure in industry 4.0. Moreover, it entered the Blockchain and edge computing paradigms in this reference. They also
worked on a blend of these paradigms and analyzed how they could enter the scalable and secure environment era. They also surveyed state of the art for scalability and security purposes.

One of the most arduous tasks is to provide security and privacy in the Internet of Things environment. Some of the limitations are solved with the introduction of Blockchain as a decentralized ledger. Pajooh et al. [17] presented the advancements of Hyper-ledger Fabric, which serves as a permission Blockchain for protecting edge computing devices in an environment of the locally employed process used in authentication. The proposed model also procured the traceability of data obtained from the IoT devices addressing scalability challenges. The paper encountered the execution of Hyper-ledger Fabric Blockchain framework desegregated with edge computing and the Internet of Things. Then they performed the tests on the VMware Virtual desktops and physical environmental setups such as Raspberry Pi devices. Apart from this, they secured the framework with multi-layered Blockchains with authentication protocols at IoT nodes of every cluster. The aftermath of this turned out to be a remarkable boost in the throughput of IoT applications.

The obstacle behind the popularised usage of edge computing was data storage security, which played an indispensable role in the evolution of intelligent computing. The authors in [18] proposed a mechanism in which they combined Blockchain with the regeneration coding to improve the reliability and security of the stored data through edge computing. They also built global layers of Blockchains in a cloud service. They also formulated local Blockchain at IoT terminals for another pool of verification. The residual nodes then selected the terminal and repaired them with regeneration coding resulting in the development of resources under the umbrella of edge computing.

Kuo et al. [19] further proposed ModelChain, which focuses on training medical health prediction framework. This allowed multiple institutions for the training in the environment of Blockchain using intelligent machine learning techniques and developed a framework for the prediction of medical health. Every site involved here fostered the estimation of model parameters. They applied transactional metadata and therefore integrated privacy-preserved models using intelligent computing. Rathore et al. [20] proposed BlockDeepNet, a Blockchain-based Deep Learning model that enlightened the collaborative paradigm of IoT and Deep Learning techniques. They verified the compatibility and feasibility of object detection under the IoT archetype through experimental analysis. However, the proposed model portrayed the need for higher computational resources, and the implementation would be impotent in the low computational machines. Ferran et al. [21] suggested a deep learning framework based upon BlockChain for the implementation under Smart Grid environment named DeepCoin. It achieved a very high throughput by applying a novel energy system associated with Byzantine fault tolerance. They used hash functions and short signatures using Blockchain technology. They prevented attacks on Smart Grids. Singh et al. [22] proposed a Deep Learning-based IoT-oriented framework under the store of a secure, secure smart city in the domain of Cyber-Physical systems and Software-Defined networking. Further, they compared their model with scalability and latency parameters and evaluated pre-existing security and privacy. But, it still suffers from hindrances occurring due to the centralization of the workspace. He et al. [23] studied the implications of edge computing with IoT and the security issues. They proposed a general framework for edge computing based upon a Blockchain environment. Further, they designed a smart contract in the private Blockchain network to fathom the hindrances caused by resource allocation in edge computing, focusing on multiple service subscribers providing good efficiency. They displayed the amalgamation of Artificial Intelligence with the Blockchain. Also, they simulated their model to achieve the accuracy of resource allocation in Edge Computing.

Dai et al. [24] witnessed the implications of Edge empowered intelligent computing for medical prospects against COVID-19. They addressed concerns like the over-centralization of resources on the Internet of Medical Things and the hysteresis of digitalization in medical services, and hindrances in proper security over the medical data. They analyzed that the collaboration of the Internet of Medical things with Blockchain technologies can somehow elixir the severity of COVID-19. However, they had several challenges to overcome, such as the absence of context in this regard, latency, and privacy issues. They presented an architecture based upon the Edge Intelligence working on the sub-domain of Blockchain technologies at the Internet of Medical Things (IoMT) platform. After that, they monitored the initiation of the pandemic and traced the supply chain that must be broken down to hinder the spread of the pandemic virus.

Li et al. [25] integrated the Blockchain and Federated learning system (BCFL) framework. They provided an in-depth survey of the framework and further discussed the insights of this new paradigm. They further studied the design of BCFL framework and discussed challenges and issues of this technology. Lastly, they discussed the applications of the framework. The researchers Nguyen et al. [26] presented the fundamental concept of Federated Learning and worked on the opportunities it opens for Mobile edge computing devices. They explored several topics such as incentive mechanisms, communication cost,
resource distribution, security and privacy issues. They also surveyed the use of Blockchain based Federated Learning in the case of famous applications in an edge-based network environment which includes edge crowd sensing, edge content caching, and edge-based data sharing. Zhang et al. [27] proposed the architecture for sharing the data in autonomous vehicles in an edge-based environment. The architecture worked on the hybrid of permission-based-blockchain and federated learning systems using a local Directed Acyclic Graph (DAG). Moreover, they adopted Deep Reinforcement Learning (DRL) methods for selecting the significant nodes and further improving the effectiveness.

3 Problems with existing federated learning systems

The following are some of the fundamental problems with the existing federated learning system:

Problem 1: Users need to trust the MEC system for model aggregation.
Problem 2: Transmission of learning parameters is highly vulnerable to attacks.
Problem 3: The MEC server can be a single point of failure in an attack.
Problem 4: Organizations may not want to provide data for various reasons.
Problem 5: Malicious users may trace back the original owner of the data.
Problem 6: Scalability issues.

Some of the existing solutions for above problems (Problem 1 to Problem 6) of federated learning system are as follow:

Solution 1: Poisoning attacks can be decreased by increasing the difficulty level of blockchain mining.
Solution 2: Using smart contracts to secure communication and message passing.
Solution 3: Differential Privacy approach to reduce the possibility of individual record identification.

4 Proposed improvement in existing solutions

4.1 Overview of the proposed approach

The current issues for Blockchain-based federated learning system are attacks like model poisoning and data privacy threats that can make the network vulnerable to outside attacks; there can be attempts to train a model using designed false data; and individual record identification; attempt to reconstruct training set from the generated gradient. Therefore, the following steps are needed:

Step 1: Increasing the difficulty level of consensus approach based Blockchain mining to decrease poisoning attacks.
Step 2: Using smart contracts to secure communication and message passing.
Step 3: Differential privacy approach to reduce the possibility of individual record identification.

Figure 3 represents the addition of noise and the corresponding relation between the raw value, noise, secure data, and queries to retrieve the information. This model propels the idea of data security after the addition of noise. To receive further improvements in existing solutions, the following actions may be implemented:

Action 1: Identification of individual records and private raw data, since it is one of the most severe threats that almost defeats the purpose of Federated Learning.
Action 2: Study of current solutions using differential privacy concerning local devices.
Action 3: Performing experiments to compare the effectiveness of the new consensus approach based Blockchain federated learning architecture with differential privacy-dependent machine learning algorithms.

4.2 Proposed mathematical model

The variables used in mathematical model are presented and described in Table 1. This section aims to formulate a mathematical model regarding the data breaching dynamics under an Edge environment. The authors have provided an approach for minimizing the risk of a data breach in an Edge Computing environment.

The proposed mathematical model consists of following hypothesis:

H1: We formulate an emphasis on a security model with a random influx rate ($B > 0$) and a general security breach rate ($\eta_1 > 0$).
H2: Exposed Data has undergone Username and Password-based authentication ($\beta_1 > 0$).
H3: Breaching of Username/Password authenticated data at a constant rate of ($\omega > 0$).
H4: The susceptible Data is breached through already breached data at a constant rate of ($\gamma > 0$).
H5: Susceptible data are sent for the authentication at Username/Password-based authentication at a rate of $\delta_1$.
H6: Transfer rate of sensitive data after Username/Password authentication to secured data after implementing proper security protocols ($\delta_2 > 0$).
H7: Already breached data are sent to OTP-based authentication at a constant rate of ($\omega_1 > 0$).

H8: OTP authenticated data are secured at a constant rate of ($\psi > 0$).

H9: Breached data at level 1, i.e., Username/Password authentication level unable to get recovered at a rate of ($\eta_2 > 0$).

H10: Data Breached at level 2, i.e., OTP authentication was unable to get recovered at a constant rate of ($\eta_3 > 0$).

H11: $B > \eta_2 > \eta_3 > \eta_1 > 0$.

Through the above hypothesis on data breaching at two-tier securities, the authors have observed the following equations to be apt as per Fig. 4:

\[
\frac{dT}{dt} = B - \gamma TJ - \eta_1 T + \delta_2 R_1
\]
Table 1 Nomenclature of variables used in research

| Symbol | Description |
|--------|-------------|
| T      | Susceptible Data |
| F      | Exposed Data |
| R₁     | Susceptible Data after User Id, Password authentication |
| J      | Already Breached Data and has the potential to breach other data |
| R₂     | Vulnerable Data after OTP authentication |
| S      | Recovered data, i.e., retrieved by applying several security measures under this paper |
| B      | Influx rate of data |
| γ₁     | Rate of a general security breach, i.e., other than authentication |
| γ₂     | Rate of a security breach due to breach at Level 1, i.e., User Id/Password authentication |
| γ₃     | Rate of a security breach due to breach at Level 2, i.e., OTP authentication |
| β₁     | Transfer rate of exposed data to susceptible data at Level 1, i.e., User Id/Password authentication |
| β₂     | Transfer rate of exposed data to breached data |
| δ₁     | Transfer rate of suspected data to already breached Data |
| δ₂     | Transfer rate of data from suspected block to another block where suspected data was secured |
| ω₁     | Rate of transfer of breached data to Vulnerable data after positive security risk |
| ω₂     | Transfer rate of breached data to recovered data |
| Ψ      | The recovery rate of breached data after OTP authentication security procedures |
| W      | A matrix containing the Jacobians concerning the rate of other transitions between blocks of w of and breached blocks, ∀i ∈ [1, 5] |
| G      | A matrix containing the Jacobians concerning the rate of appearance of new security attacks in the block of g of and breached blocks ∀i ∈ [1, 5] |
| JRFE   | Jacobian at Risk-Free Equilibrium |
| τESP   | τ at the Equilibrium for Security Provisions in Smart Edge Computing Devices using Block-chain and Machine Learning Algorithms |

\[
\frac{dF}{dt} = \gamma TJ - (\beta_1 + \beta_2 + \eta_1)F \\
\frac{dR_1}{dt} = \beta_1 F - (\delta_1 + \delta_2 + \eta_1)R_1 \\
\frac{dJ}{dt} = \delta_1 R_1 - (\omega_1 + \omega_2 + \eta_1 + \eta_2)J + \beta_2 F \\
\frac{dR_2}{dt} = \omega_1 J - (\eta_1 + \eta_3 + \psi)R_2 \\
\frac{dS}{dt} = \psi R_2 + \omega_2 J - \eta_1 S
\]  

(1)

### 4.2.1 Positivity and boundedness of proposed model

The data and parameters associated with it are assumed to be non-negative. Thus, the rate of alteration in the security factor of the data can be given by:

\[
\frac{dM}{dt} = \frac{dT}{dt} + \frac{dF}{dt} + \frac{dR_1}{dt} + \frac{dJ}{dt} + \frac{dR_2}{dt} + \frac{dS}{dt}
\]  

(2)

\[
\Rightarrow \frac{dM}{dt} = B - \eta_1 M - \eta_2 J - \eta_3 R_2
\]  

(3)

with \( M = T + F + R_1 + J + R_2 + S \)

Now, it can be deduced that whenever there is no security risk, then \( F = J = R_2 = 0 \).

Hence, \( \frac{dM}{dt} = B - \eta_1 M \) which defined the data size \( M \) having the storage of \( \frac{M}{\eta_1} \) as \( t \to \infty \).

It further defines the solution of (1) to lie in the range of \( \tau = \{(T, F, R_1, J, R_2, S) \in R^6_+: T(t) \geq 0, F(t) \geq 0, R_1(t) \geq 0, R_2(t) \geq 0, S(t) \geq 0, T + F + R_1 + J + R_2 + S \leq \frac{M}{\eta_1}\} \). Due to the positive invariant boundness of the solutions in the region defined by \( \tau \), the problem is well-formed.

### 4.2.2 Security breach number (S₀)

We will implement the next-generation matrix method to define the security breach number at the Level 2 OTP-based authentication. The security breach number is designated as the average number of security breaches at Level 2 authentication, i.e., One Time Password authentication, when data from one breached block enters another block with totally susceptible data. We denote this number as \( S_0 \). We can calculate this using the spectral radius of the GW⁻¹ matrix, which is the most significant absolute value of their...
Eigen values. This is lucidly formed through the linearization of the Eq. (1).

As per the principle of the next-generation matrix, we can deduce the security breach number as the spectral radius of the next generation matrix formed as $GW^{-1}$ of the system of Eq. (1).

\[
\begin{align*}
    g_i &= \begin{pmatrix}
        \gamma TJ \\
        0 \\
        0 \\
        0
    \end{pmatrix}, \quad
    w_i = \begin{pmatrix}
        (\beta_1 + \beta_2 + \eta_1)F \\
        (\delta_1 + \delta_2 + \eta_1)R_1 - \beta_1 F \\
        (\omega_2 + \omega_1 + \eta_2)J - (\delta_1 R_1 + \beta_2 F) \\
        (\eta_1 + \eta_3 + \psi)R_2 - \omega_3 J \\
        \eta_1 S - (\psi R_2 + \omega_2 J)
    \end{pmatrix}
\end{align*}
\]

Here, in this equation, $g_i$ is the rate of appearance of new security attacks in the block and $w_i$ is the rate of other transitions between blocks of (i) of and breached blocks, $\forall i \in [1, 5]$.

The matrix for $G$ and $W$ can be formed as follows:

\[
\begin{pmatrix}
    \gamma TJ \\
    0 \\
    0 \\
    0
\end{pmatrix}
\]

Accordingly,
\[ G = \begin{pmatrix} g_1 & g_1 & g_1 & g_1 & g_1 \\ \frac{\partial F}{\partial R_1} & \frac{\partial F}{\partial R_2} & \frac{\partial F}{\partial S} & \frac{\partial F}{\partial g_1} & \frac{\partial F}{\partial g_2} \\ \frac{\partial F}{\partial g_1} & \frac{\partial F}{\partial g_2} & g_1 & g_1 & g_1 \\ \frac{\partial F}{\partial g_1} & \frac{\partial F}{\partial g_2} & \frac{\partial F}{\partial g_3} & \frac{\partial F}{\partial g_4} & \frac{\partial F}{\partial g_5} \\ \frac{\partial F}{\partial g_1} & \frac{\partial F}{\partial g_2} & \frac{\partial F}{\partial g_3} & \frac{\partial F}{\partial g_4} & \frac{\partial F}{\partial g_5} \end{pmatrix} \\
= \begin{pmatrix} 0 & 0 & \gamma T_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \]

where \( T_0 = \frac{B}{\eta_1} \), and,

\[ W = \begin{pmatrix} \partial w_1 / \partial w_1 & \partial w_1 / \partial w_2 & \partial w_1 / \partial w_2 \\\n\partial w_2 / \partial w_1 & \partial w_2 / \partial w_2 & \partial w_2 / \partial w_2 \\\n\partial w_3 / \partial w_1 & \partial w_3 / \partial w_2 & \partial w_3 / \partial w_2 \\\n\partial w_4 / \partial w_1 & \partial w_4 / \partial w_2 & \partial w_4 / \partial w_2 \\\n\partial w_5 / \partial w_1 & \partial w_5 / \partial w_2 & \partial w_5 / \partial w_2 \\\n\frac{\partial F}{\partial R_1} & \frac{\partial F}{\partial R_2} & \frac{\partial F}{\partial S} \end{pmatrix} \]

Hence, \( W^{-1} = \frac{\text{Adj}(W)}{|W|} \)

Now, we know that \( W^{-1} = \frac{\text{Adj}(W)}{|W|} \)

\[ S_0 = \kappa(GW^{-1}) \]

\[ = \frac{\gamma B (\beta_1 \delta_1 + \beta_2 (\delta_1 + \delta_2 + \eta_1))}{\eta_1 (\beta_1 + \beta_2 + \eta_1) (\delta_1 + \delta_2 + \eta_1) (\omega_1 + \omega_2 + \eta_1 + \eta_2)} \]

(11)

Also, we can say that for the steady-state condition, our system of Eq. (9) becomes:

B - \gamma T J - \eta_1 T + \delta_2 R_1 = 0

\[ \gamma T J - (\beta_1 + \beta_2 + \eta_1) F = 0 \]

\[ \beta_1 F - (\delta_1 + \delta_2 + \eta_1) R_1 = 0 \]

\[ \delta_1 R_1 - (\omega_1 + \omega_2 + \eta_1 + \eta_2) J + \beta_2 F = 0 \]

\[ \omega_2 J - (\eta_1 + \eta_3 + \psi) R_2 = 0 \]

\[ \psi R_2 + \omega_2 J - \eta_1 S = 0 \]

(12)

Here, we have discussed the local stability of a risk-free equilibrium and the equilibrium of Security Provisions in Smart Edge Computing Devices with system given in Eq. 12 by scrutinizing the following characteristics equations.

**Theorem 1** If \( S_0 < 1 \), the risk-free equilibrium is locally asymptotically stable in the region \( \tau_0 \) for Security Provisions in Smart Edge Computing Devices using consensus approach of Blockchain and Machine Learning Algorithms, otherwise it is unstable.

**Proof** The Jacobian Matrix (12) formulates as \( \tau_0(T = 1, F = R_1 = J = R_2 = 0) \), in a risk-free equilibrium condition.

\[ J_{DFE}(\tau_0) = \begin{pmatrix} -\eta_1 & -\beta_1 & -\delta_1 & -\gamma & 0 \\ 0 & -\beta_1 & -\delta_1 & -\gamma & 0 \\ 0 & 0 & -\beta_1 & -\delta_1 & -\gamma \\ 0 & 0 & 0 & -\beta_1 & -\delta_1 \\ 0 & 0 & 0 & 0 & -\beta_1 \end{pmatrix} \]

(13)

This corresponding Eigen value of \( J_{DFE}(\tau_0) \) are as follows:

\[ \begin{align*}
S_0 &= \kappa(GW^{-1}) \\
&= \frac{\gamma T_0 (\beta_1 \delta_1 + \beta_2 (\delta_1 + \delta_2 + \eta_1))}{(\beta_1 + \beta_2 + \eta_1) (\delta_1 + \delta_2 + \eta_1) (\omega_1 + \omega_2 + \eta_1 + \eta_2)}
\end{align*} \]

(10)
\[ \lambda_1 = -\eta_1 \]
\[ \lambda_2 = -(\psi + \eta_1 + \eta_3) \quad (14) \]

Also, we can get the other three Eigen values through the equation of the cubic polynomial we have obtained.

\[ \lambda^3 + b_1 \lambda^2 + b_2 \lambda + b_3 = 0, \]

where
\[ b_1 = (\beta_1 + \beta_2 + 3\eta_1 + \delta_1 + \delta_2 + \omega_1 + \omega_2 + \eta_2), \]
\[ b_2 = (\beta_1 + \beta_2 + \eta_1)(\delta_1 + \delta_2 + \eta_1) + (\beta_1 + \beta_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\delta_1 + \delta_2 + \eta_1), \]
\[ b_3 = (\beta_1 + \beta_2 + \eta_1)(\delta_1 + \delta_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(1 - S_0). \]

If \( S_0 < 1 \), then we can deduce that
\[ b_1 > 0; b_2 > 0; b_3 > 0; \text{and} \quad b_1 \cdot b_2 > b_3, \quad [25, 26]. \]

Thus, we can say that
\[ b_1 > 0; b_2 > 0; b_3 > 0; \text{and} \quad b_1 \cdot b_2 > b_3. \]

Therefore, by implementing the Routh-Hurwitz criterion, we can deduce that in the region if \( S_0 < 1, J_{DFE}(\lambda_0) \), then it is locally asymptotically stable.

### 4.2.3 Equilibrium for security provisions in smart edge computing devices using block-chain and ML algorithms

We can obtain the Equilibrium for Security Provisions in Smart Edge Computing Devices using Block-chain and Machine Learning Algorithms with \( \tau^*(T^*, F^*, R_1^*, J^*, R_2^*, S^*) \), after solving Eq. (12) simultaneously. Thus, we get the following equations:

\[ S^* = \frac{BS_0 \delta_1 - (\gamma \delta_1 + S_0(\omega_1 + \omega_2 + \eta_1 + \eta_2)) J^*}{S_0 \delta_1 \eta_1 \delta_1} \]
\[ F^* = \frac{\gamma J^*}{S_0(\beta_1 + \beta_2 + \eta_1)} \]
\[ R_1^* = \frac{(\eta_1 + \eta_2 + \omega_1 + \omega_2) J^*}{\delta_1} \]
\[ J^* = \frac{(\omega_1 + \omega_2 + \eta_1 + \eta_2)}{\delta_1 R_1^*} \]
\[ R_2^* = \frac{(\psi + \eta_1 + \eta_3)}{(\psi + \eta_1 + \eta_3) J^*} \]

Local stability of Equilibrium for Security Provisions in Smart Edge Computing Devices using consensus approach of Block-chain and Machine Learning Algorithms:

**Theorem 2** The Equilibrium for security breaches in a Smart Edge Computing Devices using consensus approach of Block-chain and Machine Learning Algorithms with \( \tau^*(T^*, F^*, R_1^*, J^*, R_2^*, S^*) \), is locally asymptotically stable when \( S_0 > 1 \); otherwise, it is unstable.

**Proof** At the Equilibrium for Security Provisions in Smart Edge Computing Devices using Block-chain and Machine Learning Algorithms with \( \tau^*(T^*, F^*, R_1^*, J^*, R_2^*, S^*) \), the variation matrix equation (12) becomes.

\[
\begin{pmatrix}
\gamma J^* & -\beta_1 \beta_2 - \eta_1 & 0 & 0 & 0 & 0 \\
0 & -\beta_1 \beta_2 - \eta_1 & -\beta_1 \beta_2 - \eta_1 & 0 & 0 & 0 \\
0 & 0 & -\beta_1 \beta_2 - \eta_1 & -\beta_1 \beta_2 - \eta_1 & 0 & 0 \\
0 & 0 & 0 & -\beta_1 \beta_2 - \eta_1 & -\beta_1 \beta_2 - \eta_1 & 0 \\
0 & 0 & 0 & 0 & -\beta_1 \beta_2 - \eta_1 & 0 \\
0 & 0 & 0 & 0 & 0 & -\beta_1 \beta_2 - \eta_1 \\
\end{pmatrix}
\]

Thus, through Routh–Hurwitz criterion, if \( S_0 > 1 \), then the equation for Security Provisions in Smart Edge Computing Devices using Block-chain and Machine Learning Algorithms is locally asymptotically stable.
Theorem 1 asserts the risk-free equilibrium if \( S_0 < 1 \) and Theorem 2 gives an assertion for the equilibrium in the cases of security breaches if \( S_0 > 1 \) in accordance with the Routh Hurwitz criterion. The locally asymptotically stable conditions determine the stability of equilibrium formed in the local block of data stored for mobile edge-based device. The risk-free equilibrium demonstrates that the security is established under the conditions that \( S_0 < 1 \) and the data must be secured in local edge-based device if the security breach number \( (S_0) \) is greater than 1. The globally asymptotic conditions determine the optimization of security criterion for several blocks forming a consensus approach based blockchain of secured data. The global asymptotic stability signifies that the device is secured under the data connection in a mobile edge based dynamic environment.

### 4.2.4 Algorithmic representation of proposed steps

In this paper, the algorithmic representation consists of two parts. The First algorithm will perform the classification using several classifiers and model them using metrics such as f1-score, accuracy. On the other hand, the second algorithm uses the federated learning techniques for implementing this model in an edge computing-based smart environment.

Classification is a procedure for labelling the data into distinguished classes. There are several classification models in machine learning, such as Ridge Classifier, XGB Classifier, Logistic Regression, KNN classifier, Decision Tree Classifier, Random Forest Classifier, etc. [28, 29]. The authors implemented several standard classification models and found that the K-Neighbour Classification model is apt in terms of metrics such as accuracy and f1-score [30–32]. The K-Neighbour algorithm works by storing all the instances of the dataset that has been used in training using dimensional spaces. And using a simple majority approach from the neighbor nodes establishes the classification. Further, the authors considered several optimization algorithms such as S.G.D., Momentum S.G.D., Adam, and RMSprop for generating an optimized loss vs. epoch graph [33, 34]. Furthermore, the researchers used a voting classifier to ensemble multiple classifiers for the dataset of several blocks [35, 36].

\[
\begin{align*}
F_p R &= \frac{F_p}{P} \\
F_N R &= \frac{F_N}{P} \\
T_p R &= \frac{T_p}{P}
\end{align*}
\] (19) (20) (21)

\[
T_N R = \frac{T_N}{P}
\] (22)

\[
\text{Accuracy} = \frac{T_N + T_P}{T_N + T_P + F_N + F_P}
\] (23)

\[
\text{Specificity} = \frac{T_N}{T_N + F_P}
\] (24)

\[
\text{Precision error} = \frac{T_P}{F_P + T_P}
\] (25)

\[
\text{Sensitivity} = \frac{T_P}{T_P + F_N}
\] (26)

In Eqs. (19) to (26), ‘\( F_p R \)’ represents the false-positive ratio, ‘\( F_N R \)’ represents the false-negative ratio, ‘\( T_p R \)’ represents the true-positive ratio, and ‘\( T_N R \)’ represents the true-negative Ratio whereas, ‘\( F_P \)’ represents the number of false-positive cases, ‘\( F_N \)’ represents the number of false-negative cases, ‘\( T_P \)’ represents the number of true-positive cases, and ‘\( T_N \)’ represents the number of true-negative cases [37, 38]. The accuracy, specificity, precision error, and sensitivity values for providing security provisions in smart edge computing devices using consensus approach of blockchain and machine learning algorithms can be obtained using Eqs. (23), (24), (25), and (26) respectively. The algorithmic representation for classification of data blocks using machine learning techniques for providing the security provisions in smart edge computing devices is given in Fig. 5.

Then they share the data with the clients and form the data for the runtime testing under a federated learning technique [38–40]. A federated learning technique implements the training of the algorithms over several decentralized edge computing enabled devices containing the blocks of data connected over the Blockchain [41, 42]. For the same, the authors implemented the steps given in Fig. 6 to obtain the aftermath in an edge computing-based environment for the security provisions.

The Algorithm 1 as shown in Fig. 5 implements several machine learning techniques on a single block of data and further approaches to determine a risk-free equilibrium for a locally asymptotically stabilized condition for a region \( \tau_0 \) defined by the block of data in an edge computing based mobile devices. The authors in this paper applied the simple training using naïve bayes classification and then improved the optimization using several models as per apt representation for the paper. The Algorithm 2 as shown in Fig. 6 henceforth processes the trained model using the methods proposed in Algorithm 1 for each block of data and hence performed a training using Multi-Layer Perceptron based model and obtained a risk-free equilibrium state in the total region defining the consensus approach based blockchain of data for globally asymptotically stabilized model.
4.3 Experiments and result analysis

In a dynamically changing non-linearity, the optimization of risk-free equilibrium can be achieved by solving the differential equations using machine learning in a supervised learning setup. In this paper, the authors considered the dataset having several features such as data vulnerabilities, data inconsistency for people of different ages, sex, location, etc. from various media portals namely press, print media, electronic media, television reports, radio coverage, and other social media such as Facebook, Twitter, LinkedIn, and Instagram. The dataset contains several vulnerable, exposed, recovered, and secured data. The labels were determined using the classification where the data displayed a risk-free equilibrium for locally asymptotic non-linear optimization when the security breach number $S_0$ is less than 1 i.e., $S_0 < 1$ the data portraying the risk of security breaches when security breach number $S_0$ is greater than 1 i.e., $S_0 > 1$ as per Theorems 1 and 2. Henceforth, minimization of $S_0$ results in mitigating the security risk in an edge based locally asymptotic regions. Thus, the optimization goal is to minimize $S_0$ with respect to increase in time.

In this part the authors procured the dataset and classified them using the Naïve Bayes classification method to determine the class probability based upon the observed attributes [43, 44]. Further, the acquired data was normalized into the blocks of data with several chunks containing various severities. The re-sampling and approximation algorithms were applied on the blocks while

Algorithm_1 Classification_of_data_blocks_using_MLT( )
{
Step 1: Gather the dataset consisting of data having the risk of security attacks from several media sources such as social media, print, cloud, and edge.
Step 2: Generate the heatmap that represents the data correlation in a block of data using a classifier, Naïve Bayes classifier in this case, and further calculate its accuracy and F1 score by the data using Eq. 19-26.
Step 3: Feed the data from several blocks and resample them with several machine learning classification models.
Step 4: Select the model with the best score from the classification models used in step 3 using metrics such as accuracy, F1 score. Further, draw the ROC (Receiver Operating Characteristics) curve for the aftermaths obtained so far. Concerning this paper, we found that the K Neighbour classification model procured a score of 92%, which marks it better than all the models.
Step 5: Generate a report for precision, recall, F1-score, and support for several optimizers such as SGD (Stochastic Gradient Descent), Momentum SGD (Stochastic Gradient Descent), Adam, RMSprop (Root Mean Square propagation).
Step 6: Plot the loss vs. epoch graph and validation loss vs. epoch graph where loss denotes the loss of data because of a security breach in training data, validation loss represents the loss of data in a security risk for validation data, and epoch represents the years in this case.
Step 7: Use voting classifier and ensemble numerous classifier models to predict output for a model class and plot the same AUC curve securing the locally asymptotic stability.
Step 8: End of the algorithm.
}

Fig. 5 Algorithmic representation for classification of data blocks using machine learning techniques

Algorithm_2 Data_Security_enhancement_Algorithm( )
{
Step 1: Create the clients that return the clients' names mapped with the data shared amongst them.
Step 2: Generate the Batch Data, which takes the clients' data and creates objects with shared data and its labels.
Step 3: Process and separate the training data into batches for each client—other process and batch the data for the test set.
Step 4: Train the data using a simple Multi-Layer Perceptron and return the model obtained through it.
Step 5: Obtain the global accuracy and global loss across the edge devices from the model.
Step 6: End of the algorithm.
}

Fig. 6 Algorithmic representation for performing the data security in an edge-based environment
keeping a check on the target class [45, 46]. Furthermore, the authors generated a heatmap using the dataset and observed the multiple features of the correlation matrix such as age, sex, cp, etc. Using the correlation matrix, the authors concluded that the Naïve Bayes classifier gave good results for the datasets. They observed aptly correct predictions for true labels tending towards 1.00, as presented in Fig. 7. The evaluation of a classification model is based upon the performance of metrics such as accuracy, precision, recall, F1 Score [47–49]. Thus, the authors find the accuracy and the F1 Score for the given classification. The accuracy score gives the fraction of correct predictions in the classification model [50, 51]. The F1 Score shows the harmonic mean of precision and recall. In our model, we obtained an accuracy of 86% and an F1 score of 87%, as per Appendix 1. Through these scores, the authors concluded that the model is classified correctly [52–54].

The authors appended several models given in Appendix 2, such as K-Neighbours Classifier, Decision Tree Classifier, Random Forest Classifier, MLP Classifier, etc. [55–57]. Further, they feed the data from several susceptible and non-susceptible blocks and resample them following the appended machine learning classification models [58–60]. The best Score for each model was calculated and compared. According to the dataset, K Neighbours Classifier procured the best results [61, 62]. The best Score for K Neighbours was found out to be 93.1% as mentioned in Fig. 8.

Further, the authors generated a receiver operating characteristics curve (ROC curve) plotting the true positive rate vs. the false positive rate for the classification models [63–65]. The comparison of the classifiers can perform concerning area under the curve (AUC) [66, 67]. The performance of a classifier is directly proportional to the area under the curve in a ROC curve [68, 69]. In Fig. 9, the AUC for several classifiers has been depicted and it can be observed that the AUC values for K-Neighbours Classifier is highest, i.e., 0.873. According to the ROC AUC, the second-best classifier for the dataset is the MLP Classifier.

The classification report provides an insight into the behavior of the classifier around the global accuracy masking the multi-class datasets [70–72]. The authors analyzed and compared the classification report for several optimization algorithms used in training, such as Stochastic Gradient Descend Optimizer, Momentum Optimizer, Adam Optimizer, and RMSprop Optimizer [73–75].

The Stochastic Gradient Descent (SGD) optimizer generates the gradient of the cost function concerning the training dataset [76–78]. The authors developed the

Fig. 7 Generating Heat-map using Naïve Bayes Classification

Fig. 8 Result displaying the best Score achieved after using several models on our dataset
The classification report for SGD optimizer in this paper is presented in Fig. 10. The authors observed that class ‘0’ has a precision of 0.82 and recall of 0.76, further giving an F1 Score of 0.78 with a support of 41 observations. Class ‘1’ has a precision of 0.81 and a recall of 0.86, procuring an F1 Score of 0.83 with a support of 50 observations.

The momentum optimizer boosts the acceleration of the Stochastic Gradient descend in a particular direction and further stifle the oscillation of the gradient [79–81]. The classification report for Momentum Optimizer is presented in Fig. 11 [82, 83]. This optimizer suggests that class 0 has a precision of 0.76 and recall of 0.71, further giving an F1 Score of 0.73 with a support of 41 observations. Class 1 has a precision of 0.77 and a recall of 0.82, procuring an F1 Score of 0.80 with a support of 50 observations.

The Adaptive Moment Estimation implements the adaptive learning rates for every variable. It works as an amalgamation of AdaGrad and Momentum Optimizers [84–86]. The classification report for Adam Optimizer for data used in this paper is presented in Fig. 12 [87, 88]. This optimizer presents that class 0 has a precision of 0.74 and recall of 0.71, further giving an F1 Score of 0.72 with a support of 41 observations. Class 1 has a precision of 0.77 and a recall of 0.80, procuring an F1 Score of 0.78 with a support of 50 observations.
The Root Mean Squared Propagation (RMSprop) portrays itself to use the decaying average of partial gradients implemented with the adaptation of step size measured for every parameter [89, 90]. The classification report for RMSprop Optimizer for data used in this paper is presented in Fig. 13 [91, 92]. This optimizer presents that class 0 has a precision of 0.74 and recall of 0.71, further giving an F1 Score of 0.72 with a support of 41 observations. Class 1 has a precision of 0.77 and a recall of 0.80, procuring the F1 Score of 0.78 with a support of 50 observations.

The authors in this paper plot graph for loss which signifies the security breach, vs. Epoch. As per Theorem 1, if the security of the locally asymptotic region decreases significantly with increase in time, then the security of that specific region $\tau_0$ defined by the local block of data stored in an edge-based system increases. This stabilizes the
equilibrium conditions of the risk-free state as mentioned in Theorem 1. This phenomenon illustrates time (in years) in x-axis and the minimization loss for the optimization function $S_0$ in y-axis for multiple optimizers as implemented in this paper [93, 94]. Figure 14 bestows a graph that gives the inference that SGD optimizer performs poorly in this model [95]. In the case of momentum of SGD, RMSprop performs significantly well in minimizing the data loss [96, 97]. The Adam Optimizer seems to be the best fit optimizer in this paper as we see that the time, the security breach is being minimized.

Then, the authors plot a graph in Fig. 15 between validation losses vs. Epoch, representing the possibility of security breach during the validation in terms of age (in years). The SGD optimizer gives us an insight that the model is under-fitting and needs some alteration in its parameters [98, 99]. The Adam Optimizer and RMSprop hint at over-fitting as there is a significant difference between the training and validation losses [100, 101]. The Momentum Optimizer gives better results for validation loss as the loss decreases concerning the time [102]. The voting classifier implements the aggregation of different classifiers such as Random Forest Classifier, SVC (support vector classifier), Logistic Regression and predicts the class probability for the highest votes [103]. Thus, the authors implemented the ensemble classifier technique and aggregated the aftermaths procured from this model, as shown in Figure 16 after the training of the voting classifier and found that the true positive rate is increasing better than individual classifiers [107]. This defined assertion for Theorem 1 which under the risk-free equilibrium of locally asymptotic stability confirms the secured atmosphere for batches of data. As per the optimization methods mentioned in Algorithm 1 depicted in Fig. 5, the ensemble classifier obtains a risk-free equilibrium condition in a locally asymptotic region. The aftermath for ensemble classifier is obtained with an area under curve (AUC) of 92%, which is better than K-Nearest Neighbour and all other classifiers implemented in this paper [108]. This result fostered the implementation of an Ensemble classifier in this model and minimized the security risk with several blocks of data [109].

This paper determines the goal of securing the data over several blocks of the edge-based atmosphere each having locally stable regions $t_0$ and henceforth generating the combined stability of the data in a fully-fledged global server as per Algorithm 2 mentioned in Fig. 6. This serves the functionality of risk-free equilibrium in a globally asymptotic stabilized region. The results procured in this paper assert the security provisions for edge-based model using intelligent machine learning techniques. The authors
Fig. 16 Generation of area under curve for the model

Table 2 Performance comparison of various security provision approaches for edge computing

| Algorithm name                                      | Classifier taken/technologies used                                      | Global accuracy | Global loss | F1-score | AUC         |
|-----------------------------------------------------|------------------------------------------------------------------------|-----------------|-------------|----------|-------------|
| 1 SDN Based Blockchain [12]                        | S.D.N., Blockchain, Fog Nodes                                          | 87%             | 3%          | Not available | Not available |
| 2 Edge Chain [13]                                  | Edge Computing, Blockchain                                            | Not available   | Not available | Not available | Not available |
| 3 Smart Grid [15]                                  | Smart Grids, Edge Computing, K-Nearest Neighbours                     | 91%             | 8%          | Not available | Not available |
| 4 Hyper ledger Fabric Blockchain [17]              | Edge Computing, Hyper ledger Fabric, IoT                              | Not available   | Not available | Not available | Not available |
| 5 Hybrid Storage architecture [18]                 | Blockchain, smart computing, Edge Computing                           | Not available   | Not available | Not available | Not available |
| 6 Model Chain [19]                                 | Blockchain, Machine Learning                                          | 78%             | 7%          | Not available | Not available |
| 7 Block Deep Net [20]                               | Blockchain, Deep Learning                                             | 72%             | 5.4%        | Not available | Not available |
| 8 Deep Coin [21]                                   | Deep Learning, Blockchain, Smart Grid                                 | 92%             | 3.98        | 0.761      | Not available |
| 9 Deep Block Scheme [22]                            | Blockchain, Deep Learning                                             | 89%             | 2.65        | 0.697      | Not available |
| 10 Asynchronous advantage actor-critic algorithm [23]| Blockchain, Edge Computing, Deep Reinforcement Learning              | 76%             | 1.91%       | Not available | Not available |
| 11 Proposed approach                               | Edge Computing, consensus approach of Blockchain, Voting Classifier based on K Neighbours, SVM, Random Forest, Logistics Regression | 96%             | 1.5%        | 0.78       | 92%         |
Table 2 describes the performance analysis of different edge-based algorithms synced with Blockchain and other technologies. Many algorithms worked for security in edge computing and Blockchain-based environment. These algorithms are compared on the metrics such as global accuracy, global loss, f1-score, and AUC curve. The proposed model using Edge Computing, consensus approach of Blockchain, and a voting classifier based on K-Neighbours, SVM, Random Forest, Logistic Regression lucidly performs better than the pre-existing algorithmic model. The authors shared the data blocks with several clients mapped with data shared amongst them. The data contained some vulnerable data, susceptible data as shown in Appendices 4 and 5. The influx of data with the clients is taken in the edge-based environment and henceforth trained using Multi-Layered Perceptron (Appendix 6). The global accuracy obtained using the following technique was observed as 96% with a worldwide loss of 1.51 as mentioned in Fig. 17, which gives a deduction that the model can efficiently minimize the risk of data leakage in an edge-based environment when the Data is shared amongst the clients in the form of blocks [110]. The consensus approach of Blockchain bolstered the security of the data in the edge-based environment.

Fig. 17 Global accuracy and global loss as obtained after the training

5 Applications of proposed work

Edge Computing is a computing paradigm shift that paves the way for many modern computing applications. The theory associated with the proposed method can lead to several paths in the security of IoT, or widely used smart devices. As a result, we can say that the paper’s applications are diverse and are given below:

i. Banking is the first application that aptly defines the behaviour of mobile edge computing, because transactional data can be secured using consensus approach of blockchain and it allows for effective secured lines.

ii. Medical wearables can be improved, and data poisoning can be reduced, allowing patient data to be secure and model training to improve in both local and global differential privacy.

iii. The approaches used in this paper to build global and local model training frameworks can be obtained by an automated vehicle driving system. As a result, it will improve the driving experience in areas that have been trained for spatial models.

iv. Gaming experiences based on cloud computing can be enhanced using the techniques provided, resulting in a secure and risk-free gaming environment.
v. The advertisement sector can also benefit from having the noise parameters handy with the dataset in order to improve marketing strategies for specific products and secure their value in a global market.

vi. The open networks provide several paths for security breaches. So, the anomaly detection can be performed while the network is sharing an application of the proposed model.

6 Conclusions

The authors used consensus approach of Blockchain and Machine Learning techniques to improve the security provisions in smart edge computing devices. The authors obtained the dataset, which contained susceptible data, exposed data, and recovered data etc. Further, the authors used several classification techniques, including Naïve Bayes, Random Forest, K-Neighbors, and SVM, and optimized with various optimizers, including SGD, Momentum SGD, RMSprop, and Adam. Furthermore, the authors combined these classifiers using ensemble classification techniques on several blocks of data and they obtained the best results using the ensemble classifier through these implementations. The AUC for the best classifier used in the model, the K-Neighbours Classifier, was 87% whereas the Ensemble classifier increased the accuracy to 92%. After training, the authors initiated the two-factor authentication process and assessed their vulnerability. Furthermore, the authors distributed the data in batches to several clients and carried out the federated learning in an intelligent Edge computing-based environment using a multi-layered perceptron model. Finally, the proposed model was trained with a global accuracy of 96% and a global loss of 1.51. The authors used consensus approach of Blockchain to maintain a data flow channel and provided a comparative analysis of machine learning and differential privacy approaches for obtaining secure data while including noise and other datasets in the actual information. The results obtained in this paper have room for improvement by performing the experiments from both the client and server sides using distributed and global differential privacy. Further research could be conducted by comparing state-of-the-art machine learning models with dynamic programming dataset-based algorithms. Experimenting with distributed and global differential privacy can be done on both the client and server sides.

Appendix 1: Calculation of Accuracy score and F1 score using Naïve bayes approach

```python
print('Accuracy Score:')
print(metrics.accuracy_score(y_test,y_pred))

Accuracy Score:
0.8688524590163934

from sklearn.metrics import f1_score
f1_score = f1_score(y_test, y_pred)
print("F1 Score:")
print(f1_score)

F1 Score:
0.870967741935484
```
Appendix 2: Appending models for the required classification of the dataset

```python
models = []
models.append(['RidgeClassifier', RidgeClassifier()])
models.append(['XGBClassifier', XGBClassifier(use_label_encoder=False, objective='binary:logistic', random_state=0, eval_metric='logloss')])
models.append(['KNeighborsClassifier', KNeighborsClassifier(n_neighbors=3)])
models.append(['GaussianNB', GaussianNB()])
models.append(['BernoulliNB', BernoulliNB()])
models.append(['DecisionTreeClassifier', DecisionTreeClassifier(random_state=0)])
models.append(['RandomForestClassifier', RandomForestClassifier(random_state=0)])
models.append(['AdaBoostClassifier', AdaBoostClassifier()])
models.append(['MLPClassifier', MLPClassifier(random_state=42, max_iter=1000)])
models.append(['ExtraTreesClassifier', ExtraTreesClassifier()])
models.append(['CatBoostClassifier', CatBoostClassifier()])
models.append(['GradientBoostingClassifier', GradientBoostingClassifier()])
models.append(['SGDClassifier', SGDClassifier()])
```

Appendix 3: Implementation of Ensemble classifier

```python
In [27]:  
clf.fit(X_train, y_train)  
# train the ensemble classifier

Out[27]:  
VotingClassifier(estimators=[('rf', RandomForestClassifier(n_estimators=500, random_state=0)), ('svm', SVC(probability=True, random_state=0)), ('log', LogisticRegression(random_state=0))], voting='soft')
```

Appendix 4: Creation of clients for sharing the data using federated learning

```python
In [6]:
def create_clients(image_list, label_list, num_clients=10, initial='clients'):
    """
    return: a dictionary with keys clients' names and value as data shards - tuple of images and label lists.
    
    args:
    image_list: a list of numpy arrays of training images
    label_list: a list of labeled images for each image
    num_clients: number of federated members (clients)
    initial: the clients name prefix, e.g. clients_1
    ...
    """
    
    # create a list of client names
    client_names = ['{0}_{1}'.format(initial, i+1) for i in range(num_clients)]
    
    # randomize the data
    data = list(zip(image_list, label_list))
    random.shuffle(data)
    
    # shard data and place at each client
    size = len(data)//num_clients
    shards = [data[i:i + size] for i in range(0, size*num_clients, size)]
    # number of clients must equal number of shards
    assert(len(shards) == len(client_names))
    
    return {client_names[i]: shards[i] for i in range(len(client_names))}
```
Appendix 5: Assigning data into the batches and then processing the train and test data for each client

```python
In [8]: def batch_data(data_shard, bs=32):
    
    ```
    'Takes in a clients data shard and create a tfds object off it
    args:
    shard: a data, label constituting a client's data shard
    bs: batch size
    return:
    tfds object'
    
    #seperate shard into data and labels lists
    data, label = zip(*data_shard)
    dataset = tf.data.Dataset.from_tensor_slices((list(data), list(label)))
    return dataset.shuffle(len(label)).batch(bs)

In [9]:

#process and batch the training data for each client
clients_batched = dict()
for (client_name, data) in clients.items():
    clients_batched[client_name] = batch_data(data)

#process and batch the test set
```

```python
In [10]: class SimpleMLP:
    
    ```
    @staticmethod
    def build(shape, classes):
        model = Sequential()
        model.add(Dense(200, input_shape=(shape,)))
        model.add(Activation("relu"))
        model.add(Dense(200))
        model.add(Activation("relu"))
        model.add(Dense(classes))
        model.add(Activation("softmax"))
        return model

Appendix 6: Simple Multilayer perceptron model for the federated learning procedure

```python
Author contributions The third and fourth authors wrote the and conducted experiments in the supervision of first and second authors.

Funding The fund for completing this work was given by Birla Institute of Technology, Mesra, Ranchi, India (Grant No. 00071).

Data availability All data generated or analyzed during this study are included in this article (and its supplementary information files).

Code availability The required software application or custom code is given in Appendices.

Declarations

Conflict of interest Being the corresponding author, I declare that there is no conflict of interest between the authors or with any organization. The information about relevant data (if any) has been provided by the authors in the manuscript.

References

1. Nguyen, D.C., Ding, M., Pham, Q.-V., Long, L., Seneviratne, A., Pathirana, P., Li, J., Niyato, D., Poor, H.V.: Federated
learning meets blockchain in edge computing: opportunities and challenges. IEEE Internet Things J. 8, 12806–12825 (2021)
2. Li, Z., et al.: RR-LADP: a privacy-enhanced federated learning scheme for internet of everything. IEEE Consumer Electron. Magazine 10(5), 93–101 (2021)
3. Nguyen, D.C., Cheng, P., Ding, M., Lopez-Perez, D., Pathirana, P.N., Li, J., Seneviratne, A., Li, Y., Poor, H.V.: Enabling AI in future wireless networks: a data life cycle perspective. IEEE Commun. Surv. Tutor. 23(1), 553–595 (2020)
4. Wang, S., Tuor, T., Salonidis, T., Leung, K.K., Makaya, C., He, T., Chan, K.: Adaptive federated learning in resource-constrained edge computing systems. IEEE J. Sel. Areas Commun. 37(6), 1205–1221 (2019)
5. Ma, C., Li, J., Ding, M., Yang, H.H., Shu, F., Quek, T.Q.S., Poor, H.V.: On safeguarding privacy and security in the framework of federated learning. IEEE Netw. 34(4), 242–248 (2020)
6. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: BEdgeHealth: a decentralized architecture for edge-based IoMT networks using blockchain. IEEE Internet Things J. 8(12), 11743–11757 (2021)
7. Majeed, U., Hong, C.S.: FLchain: federated learning via MEController enabled blockchain network. In: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 1–4 (2019)
8. Kim, Y.J., Hong, C.S.: Blockchain-based node-aware dynamic weighting methods for improving federated learning performance. In: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 1–4 (2019)
9. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. 10(2), 1–19 (2019)
10. Park, J., Samarakoon, S., Elgabli, A., Kim, J., Bennis, M., Kim, S.-L., Debbah, M.: Communication-efficient and distributed learning over wireless networks: principles and applications. Proc. IEEE 109(5), 796–819 (2021)
11. Niknam, S., Dhillon, H.S., Reed, J.H.: Federated learning for wireless communications: motivation, opportunities, and challenges. IEEE Commun. Mag. 58(6), 46–51 (2020)
12. Sharma, K., Chen, M.Y., Park, J.H.: A software-defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access 6, 115–124 (2018)
13. Han, J., Wang, J., Hester, A., Alqerem, I., Liu, Y., Zhao, Y.: Edgechain: an edge-IoT blockchain architecture and prototype based on blockchain and smart contracts. IEEE Internet Things J. 6(3), 4719–4732 (2019)
14. Rahman, M.A., Rashid, M.M., Hossain, M.S., Hassanain, E., Alhamid, M.F., Guizani, M.: Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city. IEEE Access 7, 18611–18621 (2019)
15. Xu, A. et al.: Efficiency and security for edge computing assisted smart grids. In: 2019 IEEE Globecom Workshops (GC Wkshps), pp. 1–5 (2019)
16. Wu, Y., Dai, H.-N., Wang, H.: Convergence of blockchain and edge computing for secure and scalable IoIoT critical infrastructures in Industry 4.0. IEEE Internet Things J. 8(4), 2300–2317 (2021)
17. Pajooh, H.H. et al.: Hyperledger fabric blockchain for securing the edge internet of things. Sensors (Basel, Switzerland) 21(2), Article 359 (2021), https://doi.org/10.3390/s21020359
18. Ren, Y., Leng, Y., Cheng, Y., Wanf, J.: Secure data storage based on blockchain and coding in edge computing. J. Math. Biosci. Eng. 16(4), 1874–1892 (2019)
19. Kuo, T.T., Ohno-Machado, L.: Model chain: decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks. arXiv preprint (2018), arXiv:1802.01746
20. Rathore, S., Park, J.H.: DeepBlockIoTNet: a secure deep learning approach with blockchain for the IoIoT network. Trans. Ind. Inform. 11(14), Article 3974 (2019)
21. Ferrag, M.A., Maglaras, L.: DeepCoin: a novel deep learning and blockchain-based energy exchange framework for smart grids. IEEE Trans. Eng. Manage. 67(4), 1285–1297 (2019)
22. Singh, S., Jeong, Y., Park, J.: A deep learning-based IoT oriented infrastructure for secure smart City. Sustain. Cities Soc. 60, 102252 (2020)
23. He, Y., Wang, Y., Qiu, C., Lin, Q., Li, J., Ming, Z.: Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach. IEEE Internet Things J. 8(4), 2226–2327 (2021)
24. Dai, H.-N., Wu, Y., Wang, H., Imran, M., Haider, N.: Blockchain-empowered edge intelligence for Internet of Medical Things against COVID-19. IEEE Internet Things Mag 4(2), 34–39 (2021)
25. Li, D., et al.: Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Comput. 26, 4423–4440 (2022)
26. Nguyen, D.C., et al.: Federated learning meets blockchain in edge computing: opportunities and challenges. IEEE Internet Things J. 8(16), 12806–12825 (2021)
27. Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y.: Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Trans. Veh. Technol. 69(4), 4298–4311 (2020)
28. Brauer, F., Castillo-Chavez, C., Feng, Z.: Mathematical Models in Epidemiology, vol. 32. Springer, New York (2019)
29. Giordano, G., et al.: Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy. Nat. Med. 26(6), 855–860 (2020)
30. Lim, W.Y.B., et al.: Federated learning in mobile edge networks: a comprehensive survey. IEEE Commun. Surv. Tutor. 22(3), 2031–2063 (2020)
31. Zhao, Z., Feng, C., Yang, H.H., Luo, X.: Federated-learning enabled intelligent fog radio access networks: fundamental theory, key techniques, and future trends. IEEE Wirel. Commun. 27(2), 22–28 (2020)
32. Briggs, C., Fan, Z., Andras, P.: A review of privacy-preserving federated learning for private IoT analytics. arXiv Preprint (2020), arXiv:2004.11794
33. Yang, R., Yu, F.R., Si, P., Yang, Z., Zhang, Y.: Integrated blockchain and edge computing systems: a survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. 21(2), 1508–1532 (2019)
34. Querlatta, J.P., Westerlund, T.: blockchain for mobile edge computing: consensus mechanisms and scalability. arXiv Preprint (2020), arXiv:2006.07578
35. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning meets blockchain in edge computing: consensus mechanisms and scalability. arXiv Preprint (2020), arXiv:2006.07578
39. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: Blockchain for 5G and beyond networks: a state of the art survey. J. Netw. Comput. Appl. 166, 102693 (2020)
40. Fang, M., Cao, X., Jia, J., Gong, N.: Local Model poisoning attacks to byzantine-robust federated learning. In: 29th [USENIX] Security Symposium ([USENIX] Security 20), pp. 1605–1622 (2020)
41. Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Delghantahan, A., Srivastava, G.: A survey on security and privacy of federated learning. Futur. Gener. Comput. Syst. 115, 619–640 (2021)
42. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q.S., Poor, H.V.: Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans. Inf. Forensics Secur. 15, 3454–3469 (2020)
43. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: Blockchain and edge computing for decentralized E.M.R.s sharing in federated healthcare. In: 2020 IEEE Global Communications Conference, Taipei, Taiwan, pp. 1–6 (2020)
44. Kim, H., Park, J., Bennis, M., Kim, S.-L.: Blockchain on-device federated learning. IEEE Commun. Lett. 24(6), 1279–1283 (2020)
45. Pokhrel, S.R., Choi, J.: Federated learning with blockchain for autonomous vehicles: analysis and design challenges. IEEE Trans. Commun. 68(8), 4734–4746 (2020)
46. Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y.: Low-latency federated learning and blockchain for edge computing in digital twin empowered 6G networks. IEEE Trans. Ind. Inf. 17(7), 5098–5107 (2020)
47. Lu, Y., Huang, X., Zhang, K., Maharjan, S., et al.: Blockchain empowered asynchronous federated learning for secure data sharing in the Internet of vehicles. IEEE Trans. Veh. Technol. 69(4), 4298–4311 (2020)
48. Sattler, F., Wiedemann, S., Muller, K.-R., Samek, W.: Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Trans. Neural Netw. Learn. Syst. 31(9), 3400–3413 (2019)
49. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: Integration of blockchain and cloud of things: architecture, applications, and challenges. IEEE Commun. Surv. Tutor. 22(4), 2521–2549 (2020)
50. Kim, H., Kim, S.-H., Hwang, J.Y., Seo, C.: Efficient privacy preserving federated learning for blockchain network. IEEE Access 7, 136481–136495 (2019)
51. Liaskos, S., Wang, B., Alimohammadi, N.: Blockchain networks as adaptive systems. In: IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Montreal, QC, Canada, May 2019, pp. 139–145 (2019)
52. Wang, L., Wang, W., Li, B.: CMFL: mitigating communication overhead for federated learning. In: Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 954–964 (2019)
53. Mills, J., Hu, J., Min, G.: Communication-efficient federated learning for wireless edge intelligence in IoT. IEEE Internet Things J. 7(7), 5986–5994 (2020)
54. Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N.N., Zhou, M.: Delegated proof of stake with downgrade: a secure and efficient blockchain consensus algorithm with downgrade mechanism. IEEE Access 7, 118541–118555 (2019)
55. Liu, Y., Wang, K., Lin, Y., Xu, W.: LightChain: a lightweight blockchain system for industrial Internet of Things. IEEE Trans. Ind. Inf. 15(6), 3571–3581 (2019)
56. Hieu, N.Q., Anh, T.T., Luong, N.C., Niyato, D., Kim, D.I., Elmroth, E.: Resource management for blockchain-enabled federated learning: a deep reinforcement learning approach. pp. 1–5. arXiv preprint (2020). arXiv:2004.04104
57. Nguyen, H.T., Hoang, D.T., Luong, N.C., Niyato, D., Kim, D.I.: A hierarchical game model for OFDM integrated radar and communication systems. IEEE Trans. Veh. Technol. 70(5), 5077–5082 (2021)
58. Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y.: Communication-efficient federated learning for digital twin edge networks in industrial IoT. IEEE Trans. Ind. Inf. (2020). https://doi.org/10.1109/TII.2020.3010798
59. Liao, Z., Couillet, R.: A large dimensional analysis of least squares support vector machines. IEEE Trans. Signal Process. 67(4), 1065–1074 (2019)
60. Liu, M., Yu, F.R., Teng, Y., Leung, V.C.M., Song, M.: Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. IEEE Trans. Wirel. Commun. 18(1), 695–708 (2019)
61. Zhan, Y., Li, P., Qu, Z., Zeng, D., Guo, S.: A learning-based incentive mechanism for federated learning. IEEE Internet Things J. 7(7), 6360–6368 (2020)
62. Toyoda, K., Zhang, A.N.: Mechanism design for an incentive-aware blockchain-enabled federated learning platform. In: Proceedings of the IEEE International Conference on Big Data (Big Data), pp. 395–403 (2019)
63. Ur Rehman, M.H., Salah, K., Damiani, E., Svetinovic, D.: Towards blockchain-based reputation-aware federated learning. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 183–188 (2020)
64. BaranwalSomy, N., Kannan, K., Arya, V., Hans, S., Singh, A., Lohia, P., Mehta, S.: Ownership preserving AI market places using blockchain. In: Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), pp. 156–165 (2019)
65. Bao, X., Su, C., Xiong, Y., Huang, W., Hu, Y.: FLchain: a blockchain for auditable federated learning with trust and incentive. In: Proceedings of the 5th International Conference on Big Data Computing and Communications (BIGCOM), pp. 151–159 (2019)
66. Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., Luo, W.: DeepChain: auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Trans. Depend. Secure Comput. 18(5), 2438–2455 (2021)
67. Martinez, I., Francis, S., Hafid, A.S.: Record and reward federated learning contributions with blockchain. In: Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 50–57 (2019)
68. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: Blockchain for secure E.H.R.s sharing of mobile cloud-based E-health systems. IEEE Access 7, 66792–66806 (2019)
69. Jiao, Y., Wang, P., Niyato, D., Suankaewmanee, K.: Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. IEEE Trans. Parallel Distrib. Syst. 30(9), 1975–1989 (2019)
70. Schmid, R., Pfitzner, B., Beilharz, J., Arnrich, B., Polze, A.: Towards blockchain for auditable federated learning with trust and incentive mechanism for federated learning. IEEE Internet Things J. 7(7), 6360–6368 (2020)
71. Schmid, R., Pfitzner, B., Beilharz, J., Arnrich, B., Polze, A.: Towards blockchain-based reputation-aware federated learning. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 183–188 (2020)
72. Martínez, I., Francis, S., Hafid, A.S.: Record and reward federated learning contributions with blockchain. In: Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 50–57 (2019)
heterogeneous models. IEEE Internet Things J. 7(10), 9600–9610 (2020)
73. Arachchige, P.C.M., Bertok, P., Khalil, I., Liu, D., Camtepe, S., Atiquzzaman, M.: A trustworthy privacy-preserving framework for machine learning in industrial IoT systems. IEEE Trans. Ind. Inf. 16(9), 6092–6102 (2020)
74. Zhao, Y., Zhao, J., Yang, M., Wang, T., Wang, N., Lyu, L., Niyato, D., Lam, K.-Y.: Local differential privacy based federated learning for Internet of Things. IEEE Internet Things J. 8(11), 8836–8853 (2020)
75. Ma, S., Cao, Y., Xiong, L.: Transparent contribution evaluation for secure federated learning on blockchain, pp. 1–4. arXiv preprint (2021) arXiv:2101.10572
76. Lugan, S., Desbordes, P., Brion, E., Ramos-Tormo, L.X., Legay, A., Maq, B.: Secure architectures implementing trusted coalitions for blockchain-based distributed learning (TCLearn). IEEE Access 7, 181789–181799 (2019)
77. Li, X., Wang, Y., Song, J., Liu, Y., Zhang, X., Zhou, K., Li, C.: A low cost and un-canceled Laplace noise-based differential privacy algorithm for spatial decompositions. World Wide Web 23(1), 549–572 (2020)
78. Zhang, S., Lee, J.-H.: Double-spending with a Sybil attack in the bitcoin decentralized network. IEEE Trans. Ind. Inf. 15(10), 5715–5722 (2019)
79. Chen, Y., Qin, X., Wang, J., Yu, C., Gao, W.: FedHealth: a federated transfer learning framework for wearable healthcare. IEEE Intel. Syst. 35(4), 83–90 (2020)
80. Wang, Y., Su, Z., Zhang, N., Benslimane, A.: Learning in the air: secure federated learning for UAV-assisted crowd sensing. IEEE Trans. Netw. Sci. Eng. 8(2), 1055–1069 (2020)
81. Pham, Q.-V., Fang, F., Ha, V.N., Piran, M.J., Le, M., Le, L.B., Hwang, W.-J., Ding, Z.: A survey of multi-access edge computing in 5g and beyond: fundamentals, technology integration, and state-of-the-art. IEEE Access 8, 116974–117017 (2020)
82. Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y.: Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Trans. Ind. Inf. 16(6), 4177–4186 (2020)
83. Chai, H., Leng, S., Chen, Y., Zhang, K.: A hierarchical block-chain enabled federated learning algorithm for knowledge sharing in the internet of vehicles. IEEE Trans. Intell. Transp. Syst. 22(7), 3975–3986 (2020)
84. Yin, B., Yin, H., Wu, Y., Jiang, Z.: F.D.C.: A secure federated deep learning mechanism for data collaborations in the internet of things. IEEE Internet Things J. 7(7), 6348–6359 (2020)
85. Cui, L., Su, X., Ming, Z., Chen, Z., Yang, S., Zhou, Y., Xiao, W.: Creat: Blockchain-assisted compression algorithm of federated learning for content caching in edge computing. IEEE Internet Things J. 9(16), 14151–14161 (2020)
86. Garg, N., Sellathurai, M., Bhatia, V., Bharath, B.N., Ratnarajah, T.: Online content popularity prediction and learning in wireless edge caching. IEEE Trans. Commun. 68(2), 1087–1100 (2020)
87. Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., Chen, M.: In-Edge AI: intelligentizing mobile edge computing, caching, and communication by federated learning. IEEE Netw. 33(5), 156–165 (2019)
88. Kulkarni, V., Kulkarni, M., Pant, A.: Survey of personalization techniques for federated learning. In: 4th World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp. 1–4 (2020)
89. Pandey, S.R., Tran, N.H., Bennis, M., Tun, Y.K., Manzoor, A., Hong, C.S.: A crowdsourcing framework for on-device federated learning. IEEE Trans. Wirel. Commun. 19(5), 3241–3256 (2020)
90. Zhao, Y., Zhao, J., Jiang, L., Tan, R., Niyato, D., Li, Z., Lyu, L., Liu, Y.: Privacy-preserving blockchain-based federated learning for IoT devices. IEEE Internet Things J. 8(3), 1817–1829 (2020)
91. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Futur. Gener. Comput. Syst. 107(C), 841–853 (2020)
92. Wang, S., Wang, C., Hu, Q.: Corking by forking: vulnerability analysis of blockchain. In: Proceedings of the IEEE Conference on Computer Communications, Paris, France, April 2019, pp. 829–837 (2019)
93. Zhang, S., Lee, J.-H.: Mitigations on Sybil-based double-spend attacks in bitcoin. IEEE Consum. Electron. Mag. 10(5), 23–28 (2021)
94. Zhang, J., Zhang, J., Chen, J., Yu, S.: GAN enhanced membership inference: a passive local attack in federated learning. In: Proceedings of the 2020 IEEE International Conference on Communications (I.C.C.), Dublin, Ireland, pp. 1–6 (2020)
95. Silva, P.: Impact of geo-distribution and mining pools on blockchains: a study of ethereum—practical experience report and ongoing Ph.D. work. In: Proceedings of the 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), pp. 73–74 (2020)
96. Shlezinger, N., Chen, M., Eldar, Y.C., Poor, H.V., Cui, S.: UvEqFed: universal vector quantization for federated learning. IEEE Trans. Signal Process. 69, 500–514 (2021)
97. Jhunjhunwala, D., Gadhikar, A., Joshi, G., Eldar, Y.C.: Adaptive quantization of model updates for communication-efficient federated learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–10 (2021)
98. Chen, Z., Tian, P., Liao, W., Yu, W.: Zero-knowledge clustering based adversarial mitigation in heterogeneous federated learning. IEEE Trans. Netw. Sci. Eng. 8(2), 1070–1083 (2021)
99. Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., Arora, R.: FetchSGD: communication-efficient federated learning with sketching. In: International Conference on Machine Learning, November 20, pp. 8253–8265 (2020)
100. Ghosh, A., Hong, J., Yin, D., Ramchandran, K.: Robust federated learning in a heterogeneous environment. arXiv Preprint (2019). arXiv:1906.06629
101. Wang, Y., Su, Z., Zhang, N.: BSIS: blockchain-based secure incentive scheme for energy delivery in vehicular energy network. IEEE Trans. Ind. Inf. 15(6), 3620–3631 (2019)
102. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: Proceedings of the IEEE International Conference on Communications (I.C.C.), May 19, pp. 1–7 (2019)
103. Tang, C., Wu, L., Wen, G., Zheng, Z.: Incentivizing honest mining in blockchain networks: a reputation approach. IEEE Trans. Circuits Syst. II Express Briefs 67(1), 117–121 (2020)
104. Wang, E.K., Liang, Z., Chen, C.-M., Kumari, S., Khan, M.K.: PoRX: a reputation incentive scheme for blockchain consensus of IoT. Futur. Gener. Comput. Syst. 102, 140–151 (2020)
105. Chuan, M., Jun, L., Ming, D., Long, S., Taoatoa, W., Zhu, H., Vincent, P.H.: When federated learning meets blockchain: a new distributed learning paradigm. IEEE Comput. Intell. Mag. 17(3), 26–33 (2022)
106. Dito, A., Reda, H., Chilamkurti, N., Mahmood, A., Zaman, N., Nam, Y.: Lightweight authenticated-encryption scheme for Internet of Things based on publish-subscribe communication. IEEE Access 8, 60539–60551 (2020)
107. Shi, W., Zhou, S., Niu, Z.: Device scheduling with fast convergence for wireless federated learning. In: Proceedings of the IEEE International Conference on Communications (I.C.C.), pp. 1–6 (2020)
108. Biswas, S., Sharif, K., Li, F., Maharjan, S., Mohanty, S.P., Wang, Y.: PoBT: a lightweight consensus algorithm for scalable
IoT business blockchain. IEEE Internet Things J. 7(3), 2343–2355 (2020)
109. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: Privacy preserved task offloading in mobile blockchain with deep reinforcement learning. IEEE Trans. Netw. Serv. Manage. 17(4), 2536–2549 (2020)
110. Yang, H., Xiong, Z., Zhao, J., Niyato, D., Yuen, C., Deng, R.: Deep reinforcement learning based massive access management for ultra-reliable low-latency communications. IEEE Trans. Wirel. Commun. 20(5), 2977–2990 (2021)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Kamta Nath Mishra was born on August 15, 1973, in Chandravat Village of Kushinagar district of Uttar Pradesh, India. He received his Bachelor of Science (B.Sc., 3 years) degree from the University of Gorakhpur, India, in 1992, and Master of Computer Application (M.C.A., 3 years) degree from Madan Mohan Malviya Engineering College (Currently MMMUT), Gorakhpur, U. P., India in 1996. Dr. Mishra completed his M.Tech. (Software Systems) degree from Birla Institute of Technology and Science (BITS) Pilani, India in 2003 and Ph.D. (Engg.) from the C.S.E. department of B.I.T. Mesra—India, in May 2015. Dr. Mishra has more than twenty-one years of teaching and research experience. Currently, he is working as an Assistant Professor (Senior Grade) at Dept. of CS&E, B.I.T. Mesra, Ranchi, India since August 2009; He has worked as a faculty member in the Department of Computer Science, University of Sebha (A public University), Libya, from October 2006 to July 2009. He was a senior lecturer at B.I.T. Mesra, (Noida Campus) from July 2004 to September 2006. Dr. Mishra has worked as a senior project engineer from September 2003 to June 2004 and project engineer from September 2000 to August 2003, in the Centre for Development of Advanced Computing (Ministry of Communication & I.T., Govt. of India) Noida, Uttar Pradesh. Before joining CDAC, Dr. Mishra worked as a lecturer in the CS&E department at Krishna Institute of Engineering & Technology (KIET), Ghaziabad, India, from July 1998 to August 2000. Dr. Mishra has published two books, fifteen book chapters, and more than forty research papers in journals and conferences of international repute. His research interest includes Biometric Systems, Image Processing, Analysis of Algorithms, and Distributed Cloud Computing. Dr. Mishra is a professional member of IEEE Biometric Society U.S.A., and A.C.M., U.S.A.

Vandana Bhattacharjee is working as a Professor in the Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi. She completed her B.E. (C.S.E.) in 1989 from B.I.T. Mesra and her M. Tech and Ph.D in Computer Science from Jawaharlal Nehru University New Delhi in 1991 and 1995 respectively. She has several National and International publications in Journal and Conference Proceedings. She has authored a book on Data Analysis. She is a Life Member of the Computer Society of India. Her research areas include Machine Learning and its applications. Currently, she is working on Deep Learning techniques applied to the domains of Software Fault prediction, Classification of Images, Disease prediction, analysis of Remote sensing images, and sentiment analysis.

Shashwat Saket hails from Ranchi, Jharkhand. He has completed his secondary education from D.A.V. Public School, Pundag, Ranchi, and his higher secondary education from Delhi Public School, Sail Township, Ranchi. Currently, he is pursuing his Bachelor of Technology from Birla Institute of Technology & Science, Jharkhand, India, in Computer Science & Engineering with a specialization in Computational Intelligence. He is intrigued with Python, R language and has a further ambition to follow up his research in the fields such as Deep Learning, Machine Learning, Data Science, AI in Medical, Generative Adversarial Networks, Cryptanalysis, and Network Security.

Shivam Prakash Mishra was born in Ghaziabad, Uttar Pradesh, and completed his major part of schooling, including secondary and higher secondary education from Bethany Convent School Naini, Allahabad. Currently, he is pursuing his Bachelor of Technology from Birla Institute of Technology, Jharkhand, India, in Computer Science & Engineering. He aspires to follow his further research in Deep Learning, Machine Learning, Cryptography, and Cryptanalysis.