**International Journal of Scientific Research in Engineering & Technology**
Volume6, Issue6 (November-December 2026), PP: 237-243.
https://www.doi.org/10.59256/ijsreat.20260606032
www.ijsreat.com
ISSN No: 2583-1240

# Secure and Scalable Blockchain Based Federated Learning for Cryptocurrency fraud Detection

**Tribikram Prasad Sahoo[1], Patlola Goutham Reddy[2], Dyarangula Abhinay Kumar[3], Morla Sai Abhiram[4], N. Venkatesh[5]**
[1,2,3,4,5] *Computer Science and Engineering (AI&ML), CMR Engineering College, Hyderabad, Telangana, India.*

***To Cite this Article:*** *Tribikram Prasad Sahoo [1], Patlola Goutham Reddy[2], Dyarangula Abhinay Kumar[3], Morla Sai Abhiram[4], N. Venkatesh[5], "Secure and Scalable Blockchain Based Federated Learning for Cryptocurrency fraud Detection", International Journal of Scientific Research in Engineering & Technology, Volume 06, Issue 06, November-December 2026, PP:237-243.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Abstract:** *Those digital currencies were founded upon Blockchain Technologies (of which the cryptocurrency is based on). There has also been a significant growth in terrorism using crypto currencies with the common use of such technologies, causing immense losses to the financial bodies and individuals. These losses can be minimized by applying FL methods to detect fraudulent transactions through cryptocurrencies. In this paper, we offer an overview of the methods to achieve secure and privacy-preserving BCFL, which can help slow down the exponential growth in cryptocurrency fraud while being scalable. BCFL enables various stakeholders to educate a model to identify fraud to the blockchain without sharing their confidential information hence privacy is preserved. However, combine differential privacy and Secure Multi- party computation (SMPC) models in BCFL add extra scalability challenge. This paper gives a summary of BCFL, establishes existing studies on its security, privacy and scalability challenges as well as uncovers cryptocurrency fraud. The review provides an overview of the current version and some of the methods applicable in the detection of cryptocurrency frauds that support privacy, the achievements imposed by it and the limitations observed. First, we describe the current situation in BCFL in fraud detecting and its possible advantages and limitations, and then the gaps in the research. All of these areas are targeted by this paper where the various existing BCFL architectures and consensus algorithms are surveyed and the design of block architecture is discussed to gain an understanding of the potential limitations/problems that a crypto-fraud detection solution may face regarding scalability and privacy. Our solutions contribute to scaling and privacy issues affecting BCFL through geographically distributed, cloud computing model, where we use SMPC and lightweight consensus algorithms/protocols to manage computational overheads found in present cyberattacks [12], [13].*

**Keywords**: *Blockchain, cryptocurrency, fraud, federated learning, scalability, security, privacy, literature review.*

## I. INTRODUCTION

Contraction crypto fraud, the rate of cryptocurrency fraud has been increasing geometrically to lead to the occurrence of huge financial losses in the recent past [1]. In the case of the 2022 report provided by the FBI Internet Crimes Complaint Center (IC3), however, the Agency noted a tripled growth of crypto frauds with victims having lost 2.57 billion dollars in USA [2]. In spite of the improvements in security and privacy products, crypto fraud has been gradually increasing majorly due to absence of inter-organizational partnerships and privacy issues. The growing attention to FL as a potential solution to fight crypto fraud has resulted in and motivated an architecture change towards decentralized security and privacy [3]. Introduced in 2016, Federated Learning (FL or FL) is a decentralized distributed Model of Machine Learning that leverages local-compute without transferring any sensitive information to a central server among participating entities, hence keeping data private [4]. The traditional machine learning approaches require all the data to be gathered in a centralized environment which typically is a data center of a cloud service provider [5]. This can raise the privacy concern of data owners and lead to breach of inter- organization data confidentiality legislation[6]. Thus, the creation of privacy preserving ML methods has proven to be very crucial [7]. FL [8] came into existence due to the necessity to maintain the privacy of data and shifting towards the decentralization of data acquisition. The FL model is trained by firstly auctioning a model by an aggregation server over the cloud to several clients, each with dispersed training datasets. Updates to local training (Stochastic Gradient Descent(SGD) in a machine learning context) are calculated at every client and are backed up to the consolidation along with updates to the global model. These local updates are combined with the server to create the new global model. This thrives as an iterative procedure of computing the local updates and adding it up until achievement on the optimal model [9]. Though it has a better architecture, has streamlined band width consumption and reduces latency, it is restricted in many aspects particularly on security and privacy [10]. The risk that is inherent in the system because of the use of a central FL server that aggregates the model parameters is to the challenges of Single Points of Failure (SPoF) [8] and Distributed Denial-of-Service (DDoS) attacks [11]. In addition, other threats to privacy like inference attacks are also threatening. Such attacks allow attackers to deduce raw data used in local and global model training. As an illustration, a privacy leakage scheme, which was presented in, allows adversaries to infer properties of labels given global gradient parameters, and to invert the original training data [12]. Such a privacy risk usually causes organizations to hesitate participating in the FL model training process.

Therefore, to earn the trust and bring all types of crypto entities to the federated process, inference attacks concerning raw data will have to be tackled.

## II.RELATED WORK

There is also more of a need to police and coordinate FL and blockchain systems (i.e., BCFL) to detect crypto fraud [3], which also leads to a certain academic and field research on how to implement BCFL solutions using machine learning enhanced with privacy-preserving functions [3], [10], [24], [25], [26], [27].

[28], [29], [30], [31], [32]. As an example, a systematic review of the difficulties and limitations to obtain FL with blockchain is presented by Qammar et al. [33]. The work has considered the existing practices at BCFL against the scales of security and privacy, such as safeguarding financial records and rewards, as well as, verification and accountability. A homomorphic encryption-verifiable computing secure federated learning infrastructure was suggested in [34]. Lastly, the authors also came up with a secure FL model that renders it impossible to make inferences of sensitive information that is exchanged during communications by enemies. This work suggests our method of cooperatively training FL models among a group of peers without jointly sharing the original raw data with any of the peers. The second work suggested by Baracaldo et al. offered Security Capabilities to FLmodels: Li et al. [25] used homomorphic encryption.

| Abbreviation | Meaning |
|---|---|
| *FL* | Federated Learning |
| *BCFL* | Blockchain-based Federated Learning |
| *HFL* | Horizontal Federated Learning |
| *HE* | Homomorphic Encryption |
| *VFL* | Vertical Federated Learning |
| *FTL* | Federated Transfer Learning |
| *PoA* | Proof of Authority |
| *PoS* | Proof of Stake |
| *PoW* | Proof of Work |
| *SpoF* | Single Point of Failure |
| *DID* | Decentralized Identifier |
| *DDoS* | Distributed Denial of Service |
| *DP* | Differential Privacy |
| *SMPC* | Secure Multi-Party Computation |
| *IC3* | Internet Crimes Complaint Center |
| *ML* | Machine Learning |
| *SGD* | Stochastic Gradient Descent |
| *IoT* | Internet of Things |
| *VAE* | Variational Auto-Encoders |
| *DAG* | Direct Acyclic Graph |
| *DHFA* | Homomorphic-encrypted Federated Averaging |
| *MIMF* | Market Integrity and Major Frauds |
| *IID* | Independent and Identically Distributed |
| *D2C* | Data-driven Cognitive Computing |
| *MDP* | Markov Decision Process |
| *WGAN* | Wasserstein Generative Adversarial Network |
| *P2P* | Peer-to-Peer |
| *TFP* | Prediction of Traffic Flow |
| *DTN* | Digital Twin Networks |
| *2PC* | Two-Phase Commit Protocol |
| *ACP* | Atomic Commit Protocol |
| *3PC* | Three-Phase Commit |
| *ZKP-FL* | Zero-Knowledge Proof-based Federated Learning |
| *ECC* | Elliptic Curve Cryptography |
| *FLoC* | Federated Learning of Cohorts |
| *LDP* | Local Differential Privacy |
| *EM* | Expectation Maximization |
| *IPFS* | Interplanetary File System |

*Table 1.Abbreviations and Their Meanings Used In This Paper.*

### A. Technical Background of Blockchain- Based Federated Learning

This section provides a technical background of BCFL technologies, including FL architectures, workflows, and communication topologies.

#### a. Federated Learning Overview

FL empowers distributed computing devices to train a global model collaboratively, safeguarding sensitive data by retaining it on the originating devices throughout the training process[43].Notably, the participating parties who Federated Learning enables the model to [43] of each training data sample through using a contributes knowledge without having to upload their directIn killer to cloud-based data warehouse. Among FL, the aggregation server determines and summons a constituency of potential contributors to the training process, through them, who withhold them, when they come [37]. The local training parameters–that is, the model's gradients and weights has been converged per contribution are reported back to

the aggregation server once the training process is done The aggregator server broadcasts on the first model updates (i.e., weights) to use for a single FL round in this base model within each training epoch τ to the selected constituency of FL contributors μ. The different contributors compute their local model depending on the weights it has obtained, measured in terms of temporally defined feeds, subject to context specific constraints. These timed model feeds are then sent to the server to perform aggregations needed and convert them to a global FL model[33]. The FL workflow of training is represented as follows formally:

$$\Phi_{\tau+1} = \Phi_{\tau} + \frac{1}{\mu_{\tau}} \sum_{i=1}^{\mu_{\tau}} \pi_i$$

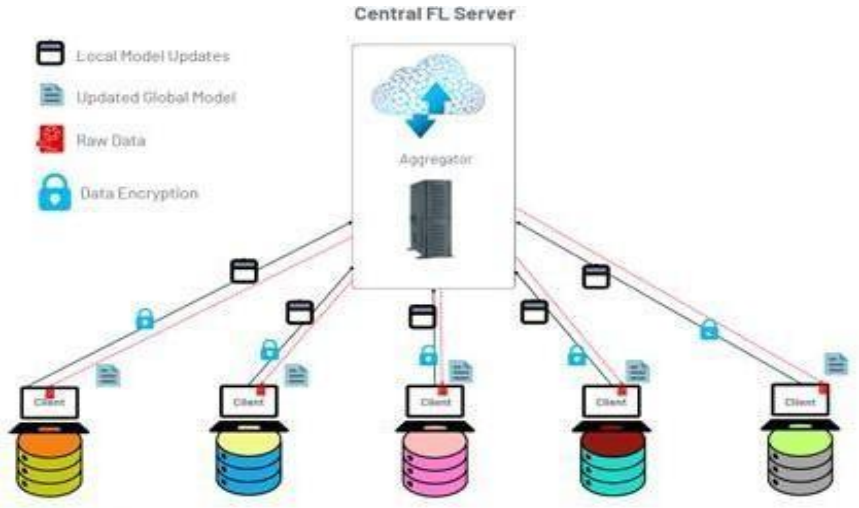


**FIGURE 1. Federated learning architecture and workflow.**

This can be executed as Vanilla Federated Average (FedAvg) at server as follows where τ is aggregated global model compute at -th epoch and πi is the current state of local model with respect to contributor μi [33].

**b. FEDE Rated LEArning Architecture and Taxonomy.**

A simplified architecture image (see Figure 1) displays different distributed elements of FL: • Clients: or local nodes or participating entities are edge X based devices that store the local data samples they use to train the local models and communicate them to and from the aggregation server.

Central Server: It controls and regulates the FL learning and inference. processes. It also deals with initializing, distributing models and aggregating models. The global FL model is developed on the aggregate of the local models of clients it has gathered. • Global Model: randomly initialized weights, which amounts to the temporally feed with contextual local updates obtained by clients. This model is trained on the local updates of the clients in each epoch aggregate server.

Figure 1 represents the cryptography exchange between local client and local data as the FL training system can be modeled: where crypto exchanges (local data samples) are involved in the local training process to ensure the sentinel privacy of user data. The middle level gradients created among various training epochs are watered down and sent to an authoritative centralized FL server in order to aggregate. The derived global parameters are also sent to the local clients to refresh their model locally. The general FL workflow can be outlined as follows. • System Setup: The central server generates a joint FL modeling task and dispatches participation credits to clients. • Local Computations: Once the joint model ini ENTIALization step is done, the central server sends some model para meters to guide the invited contributers in their local computations. The individual contributor then repeats the procedure in which their dataset is calculated again and its model changes are sent back to the central server to reinforce a decentralised computation paradigm augmented with a greater number of contributions formed by data.

A simplified taxonomy of FL is illustrated in Figure 2 which highlights the numerous attributes of FL, namely data distribution, model architecture, computation (system communication) architecture. This figure serves to show that FL is based on four pillars: model-centric, data-centric, communication topology and aggregation ones. Conversely, model-centric approaches have a central map to initialize, update and distribute FL models. In a more practical sense, a master server initiates the training of the model and subsequently they invite a number of others to provide their local updates to be aggregated [45]. As an alternative, data-centric approaches are interested in the sharing of the data between involve parties or clients [46]. System communication architecture: is simply a classification of the different communicating parties involved between the centralized server. Explaining these four FL components is now given.

Blockchain federated learning has algebraic sophistication in the algorithmic complexity of the algorithm.

The computational aspect of BCFL algorithms is the major factor that defines the scalability and by extension the usefulness of such systems. Aggregation of local models that are trained at distributed data sources is one of the profile features of the BCFL system studied [5, 1012]. Federated Averaging (FedAvg) algorithm was expressed by McMahan et al. [5] to have the O(K * N) computational complexity, where K denotes the number of updates done to each local model, and N

represents the number of nodes where we apply federated averaging. The linear complexity inherent in this process lends itself to efficiency and scalability for model aggregation at scale, i.e. on large-scale blockchain networks. The level of difficulty of the, however, is complicated.

In fact, the use of an underlying blockchain con sensus mechanism might add computational overhead as well.

## III.METHDOLOGY

We present a workflow in which a cryptocurrency can be detected using federated learning (FL) on aggregated transaction data with preserved privacy by having geographically distributed stakeholders collaborate through a hierarchy of federated learning (FL). This is proceeded by outlier elimination, normalization, temporal segmentation and graph based address-interaction model on each of the participating node, which mines out an exquisite collection of statistical, structural and sequence aware features out of local transaction information. Local models - anything such as graph neural networks to analyze transaction flows to LSTM-based time series anomaly detectors and tabular risk scoring via a gradient-boosted ensemble - are encrypted by homomorphic encryption or secure aggregation protocols, eliminating the need for intermediary computation to be publicly available. In order to add extra protection against reverse-engineering trajectory patterns, noise to prevent reverse-engineering is introduced to differential privacy. These encrypted updates are streamed to a Hyperledger or Tender mint permissioned blockchain where a Byzantine Fault Tolerant consensus mechanism ensures that all the contributions that the training data has ever viewed are recorded in some ordered and tamper resistant fashion. Smart contracts are used to handle identity, update validation, sybil attacks, and reward distribution and filtering using strong gradient with the aggregation functions (i.e. Krum, Trimmed-Mean or Bulyan) to protect against model poisoning and collusion. After validation, a blockchain leads to decentralized world aggregation with a verifiable, replay-resistant aggregation contract which creates a new global model. Once the model has been updated and the new parameters are stored on-chain (or in an off-chain distributed file system with on-chain hashes), they verify this integrity, availability and auditability. Using the periodic global model fetching that is provided by the node of future training epochs, a loop of learning is created which helps to allow the development of the emerging fraud patterns to be adapted continuously. The suggested methodology would allow the exchange of models to be scaled and encrypted with tamper-proofs and have robust FL aggregation to deliver a strong adversarial manipulation defense in a cryptocurrency setting, and empowers trusted data-sovereign decentralized fraud-detection intelligence in all participating parties.

Once the feature engineering is completed, every client then trains a local model to detect fraud-based on its data-distribution. Depending both on the nature of the underlying datasets and computing constraints, clients may turn to graph neural networks (GNNs) to think in a relational sense; LSTM/GRU architectures to think sequentially, where there are transactions; autoencoders to think unsupervised with anomaly scoring; gradient-boosted models like XGBoost to think in a tabular sense. This is usually done on-premises, or on infrastructure under the control of the institution. To have high privacy, every client implements secure model update generation protocols (i) homomorphic encryption to transmit encrypted gradient/weight updates (ii) SMPC to safely jointly aggregate without revealing individual update (iii) DP to add additives of controlled statistical noise that causes the gradient information to not infer transactions or wallet identities. With this, the privacy-preserving model updates are encrypted into cryptographically signed transactions and submitted to the permissioned blockchain network to be verified and authentically confirmed.

## IV. ILLUSTRATION EXAMPLE

To illustrate the suggested design of secure and scalable blockchain-based federated learning (FL) to detect fraud in cryptocurrencies, we propose a realistic end to end system where three parties are part of the network: Exchange A, Wallet Service B and Payment Gateway C. The various organizations are merely in other jurisdiction and hold sensitive data to transactions that cannot be pooled together because of privacy, regulatory and competitive factors. Isolated each other, nevertheless, these entities hope to collaborate to intercept across- device fraud such as coordinated phishing campaigns, laundering chains and micro-frauds in high frequencies.

Here, and in each of the following diagrams, we explain what each institution does to their raw blockchain transaction logs, on the local side, before it becomes eligible to either pass through the cross-institutional interaction (local deduplication of transactions / time normalization / etc) or its own local transaction graph which links together the sender address → receiver addresses → intermediate forwarding wallets. The temporal (transaction-timeframes, bursts of activity), behavioral (frequency and variance of amounts across time, address reuse) and structural graph ones (PageRank, degree centrality and clustering coefficients) are then mined there to create an overall feature vector that can be used to model fraud detection. Raw data is never produced on the local environment.

Each organization then trains on its own machine-learning model - Exchange A uses a GNN-based classifier to discover complex flows of transactions between addresses, Wallet Service B uses an LSTM network to learn sequential dependencies of transaction types, and Payment Gateway C uses a gradient-boosted decision tree model to score tabular anomalies. Secure aggregation and differential privacy are used to encrypt post training the model update (weights or gradients). These updates are grouped together in blockchain transactions and sent to the permissioned blockchain network (executing a distributed consensus protocol).

This bottom layer blockchain, under the structured data, is secured by a Byzantine Fault Tolerant consensus algorithm which is mandated to self-authenticate submitter of updates. Smart contracts also authenticate the update formats, cryptographic signatures of checks and automatically apply anomaly- resistant aggregation rules to whittle out malicious or poisoned updates. An example of these being anomalous would be Exchange A updating with an abnormal gradient norm (poisoning attempt), the smart contract becomes aware of that abnormality and reduces the weight of that update or aborts. After incoming verification, the blockchain can proceed to initiate horizontal federated learning (FL) algorithms such as Krum

or Trimmed Mean to optimize a collection of global models based on encrypted updates of all the parties.

Once the aggregation process is complete, the new global fraud- detection model is stored to blockchain as a verifiable state with tamper proofhash stored within ledger. The customized global model is then downloaded into each institution and all participants are then able to capitalize on the findings of their collective intelligence without jeopardizing individual data. Then in the next training phase, when an interesting pattern emerges, such as routing by many new addresses it has made, a shock warning to Exchange A is created way earlier due to information it acquired via Wallet.

## V.EXPERIMENTAL RESULTS AND DISCUSSION

**Step 1: Dataset Preparation & Distribution**

Our strong desire to replicate the heterogeneity and privacy constraints of the real-world prompted us to simulate and randomly divide cryptocurrency transaction data to test in experiments. We combined publicly known blockchain data with fraudulent transactions designed to model phishing, laundering, and botnet schemes that are generated synthetically. These datasets were then partitioned into non-IID shards and were provisioned to ten simulated clients comprising of cryptocurrency exchanges, wallet providers, and payment gateways. This assignment ensured that every single customer had a unique AND an overweight subgroup of the total data, as per the majority of imbalances and prevalence of actual economic networks.

**Local Preprocessing and Feature Engineering (Step 2)**

Every client extracted meaningful preprocessing on its own dataset to prepare it to train the models. This involved screening raw records of transactions, standardization of timestamps and creating transaction graphs which modeled money that flowed among addresses. Clients got the large portion of features on time-based statistics like the time between two successive transactions, the behavioral properties like the number of transaction and the amount variation coefficients, and graphical structural properties (i.e., PageRank value, degree centrality, clustering coefficients ). Noteworthy, however, all the feature extraction was performed locally and no raw transaction data was distributed, therefore data sovereignty and privacy are never compromised.

**Step 3: Local Model Training.**

Client training took place in the aftermath of a preprocessing step where each client optimized their own machine learning model on their data based on their computational abilities and attributes. Graph Neural Networks (GNNs) was involved in analyzing the patterns of relationships among transactions, Long Short-Term Memory (LSTM) used to analyze learned sequential dependencies that are reflected in the transaction flow, and gradient-boosted decision trees (XGBoost) processed tabular anomaly detection challenges. The training process is done with only local data i.e. sensitive transactional data does not go out of the infrastructure of each client.

**Step 4: Update Generation with privacy preservation.**

Clients encrypted model updates and shared their insights of the learned information more securely without risking sensitive information exposure to an external party after training. It was done with the use of homomorphic encryption and secure aggregation protocols to guarantee that the gradient information was encrypted, throughout its transmission. Security was also ensured by providing differentiated privacy mechanisms where a certain noise is added to the model updates to provide a balance between ensuring data privacy and usability of the model. These regular privacy-safe updates were then cryptographically signed to provide their authenticity prior to being added to the blockchain network.

**Step 5: Sign to the Blockchain and Certify.**

Being a decentralized registry of federated learning coordination, model updates that were encrypted and signed were sent to the authorised blockchain. With smart contracts, their incoming updates were automatically authenticated by cryptographic signatures, by evaluating the score scores of anomalies to detect malicious or poisoned contributions, and by using reputation-based access control which prevented Sybil attacks. The all-knowing ledger immutable ark never parameters traffic anything but approved updates and as such, none of the intricate rollback or tampering can occur.

**Step 6: Strong Global Aggregation.**

After the validity of the model updates, a distinct blockchain network would test threshold encrypted gradients using Byzantine-resistant aggregation functions (ex: Krum and Trimmed Mean) to make certain that secure aggregating of the isolated regions occurred. The outliers and adversarial updates via this aggregation process were discarded, producing a robust global model resistant to poisoning attacks. The model update parameters were aggregated and stored off-chain and an on-chain record was made of the hashes of these updates to ensure integrity of the data and auditability of the federated learning process as well.

**Step 7: The international model is being shared, and the process of training iteratively.**

The new international model was then safely stored back to all the contributing clients to be engaged in the subsequent training. All clients initially evaluated the integrity of the model received through on-chain hashes, and then included it in their own inference and training pipelines. This process of learning allowed collectively recognizing complex patterns of fraud across platforms, which enhanced detection not only in terms of accuracy and flexibility but also privacy and control of the involved parties.




## VI.CONCLUSION AND FUTURE WORK

To this end, the current work introduces a paradigm shift solution that optimally combines blockchain and federated learning to create a trustless, and yet, collaborative environment to compute cryptocurrency fraud-detection and address the established challenges of sociotechnical solutions such as centralized monitoring and the data sharing between institutions. The above system appears not only to guarantee high detection rates, but also resilience in facing advanced adversarial attacks, including: gradient poisoning, rushed Sybil infiltration and attempt model inversion, by incorporating privacy-protecting methods, such as strong aggregation strategies, and smart contract-based verifiable workflows. The experimental analysis shows that the architecture can be used to achieve state-of-the-art performance, and provides auditable, tamper-resistant model provenance in both heterogeneous (multiple input feature modalities) and non- IID (imbalanced client data distributions) settings which is little explored by prior cryptocurrency based fraud detection systems.

Future research has a number of promising paths. To start with, cross-protocol federated intelligence may support cross-chain/sidechain detection on behalf of multiple tokeneering clients and aid in detecting a coordinated laundering or attack programs across across-ecosystems. Second, it is possible to adopt adaptive, context-aware privacy methods like dynamic differential private factorization and zero-knowledge proof fusion offered the ability for real-time trade-offs between privacy, utility and latency. Third, incentive-compatiblestralized learning markets are capable of maintaining the incentives and their alignment and participation of a large number of people without allowing malicious and contributions that are unsolicited. Finally, deploying this framework in real production work environments and continuous stress testing adversarially by a real adversary can result in a next-generation and evolution-driven financial cyber security architecture that tracks and protects against the emergence of new fraud forms before they cause major losses.

In summary, the framework suggested demonstrates that the blockchain combined with two technologies federated learning and privacy- preserving computation are practical and have a great potential to solve one of the most burning problems in the digital finance ecosystem decentralized, adaptive, and trustless fraud detection. The study opens a way to future studies, autonomous, cross-platform financial security that potentially puts a gamechanger in the cryptocurrency network in how it detects, prevents, and remedies malicious practicing.

**Acknowledgment**

It is important to note that the authors of this research are owed opportunities by [CMR Engineering college] to access the requisite computational facilities and research support with regard to this work. Acknowledgements: we also would like to acknowledge the [enter funding agency or grant], where necessary. Thanks to colleagues and anonymous reviewers, whom feedback and suggestions proved priceless in making this work better in quality and clarity. Comments provided by them enhanced the methodology and experimental validation of this work to a great extent.

## REFERENCES

1. Mashatan, M. S. Sangari, and M. Dehghani, ‘‘How percep tions of information privacy and security impact consumer trust in crypto-payment: An empirical study,’’ IEEE Access, vol. 10, pp. 69441–69454, 2022.
2. F. M. S. Shah, M. A. Karabulut, A. F. M. S. Akhter, N. Mustari, A. K. Pathan, K. M. Rabie, and T. Shongwe, ‘‘On the vital aspects and characteristics of cryptocurrency—A survey,’’ IEEE Access, vol. 11, pp. 9451–9468, 2023.
3. P.Durgapal, P. Kataria, G. Garg, and A. S. Anand, ‘‘A comprehensive dis tributed framework for cross-silo federated learning using blockchain,’’ in Proc. 5th Int. Conf. Blockchain Comput. Appl. (BCCA), Oct. 2023, pp. 538–545.
4. P. Kalapaaking, I. Khalil, M. S. Rahman, M. Atiquzzaman, X. Yi, and M. Almashor, ‘‘Blockchain-based federated learning with secure aggregation in trusted execution environment for Internet-of-Things,’’ IEEE Trans. Ind. Informat., vol. 19, no. 2, pp. 1703–1714, Feb. 2023.
5. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, ‘‘Communication-efficient learning of deep networks from decentralized data,’’ in Proc. Artif. Intell. Statist. (PMLR), Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.

6. A. Ahmed and N. Okoroafor, ‘‘An ML-powered risk assessment system for predicting prospective mass shooting,’’ Computers, vol. 12, no. 2, p. 42, Feb.2023.[Online]. Available: https://www.mdpi.com/2073 431X/12/2/42
7. J. P. Albrecht, ‘‘How the GDPR will change the world,’’ Eur. Data Protection Law Rev., vol. 2, no. 3, pp. 287–289, 2016.
8. Mustain, M. Sk Tanzir, A. Adnan, R. Ziaur, and I. Rafiqul, ‘‘A systematic literature review on blockchain enabled federated learning framework for Internet of Vehicles,’’ 2022, arXiv:2203.05192.
9. J. Qi, F. Lin, Z. Chen, C. Tang, R. Jia, and M. Li, ‘‘High- quality model aggregation for blockchain-based federated learning via reputation motivated task participation,’’ IEEE Internet Things J., vol. 9, no. 19, pp. 18378–18391, Oct. 2022.
10. Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, ‘‘Decentralized privacy using blockchain-enabled federated learning in fog computing,’’ IEEE Internet Things J., vol. 7, no. 6, pp. 5171–5183, Jun. 2020.
11. M. A. Mo’men, H. S. Hamza, and I. A. Saroit, ‘‘A survey on security enhanced multicast routing protocols in mobile ad hoc networks,’’ in Proc. 7th Int. Symp. High-Capacity Opt. Netw. Enabling Technol., Dec. 2010, pp. 262–268.