Towards Blockchain-Based Reputation-Aware Federated Learning

Muhammad Habib ur Rehman, Khaled Salah, Ernesto Damiani, Davor Svetinovic
Center for Cyber-Physical Systems
Khalifa University of Science and Technology
Abu Dhabi, United Arab Emirates
{muhammad.rehman, khaled.salah, ernesto.damiani, davor.svetinovic}@ku.ac.ae

***Abstract*—Federated learning (FL) is the collaborative machine learning (ML) technique whereby the devices collectively train and update a shared ML model while preserving their personal datasets. FL systems solve the problems of communication-efficiency, bandwidth-optimization, and privacy-preservation. Despite the potential benefits of FL, one centralized shared ML model across all the devices produce coarse-grained predictions which, in essence, are not required in many application areas involving personalized prediction services. In this paper, we present a novel concept of fine-grained FL to decentralize the shared ML models on the edge servers. We then present a formal extended definition of fine-grained FL process in mobile edge computing systems. In addition, we define the core requirements of fine-grained FL systems including personalization, decentralization, fine-grained FL, incentive mechanisms, trust, activity monitoring, heterogeneity and context-awareness, model synchronization, and communication and bandwidth-efficiency. Moreover, we present the concept of blockchain-based reputation-aware fine-grained FL in order to ensure trustworthy collaborative training in mobile edge computing systems. Finally, we perform the qualitative comparison of proposed approach with state-of-the-art related work and found some promising initial results.**

***Index Terms*—blockchain, machine learning, federated learning, mobile edge computing, reputation, trust.**

I. Introduction

Google introduced federated learning (FL) as a new mechanism to share privacy preserving local machine learning model updates in edge devices for global updates in centralized deep learning models on their cloud environments [1]–[3]. FL works as collaborative learning scheme whereby the edge devices perform onboard execution of local learning models and continuously update in their local execution environments. In the case of significant change detection, edge devices push new information to centralized cloud infrastructure after applying anonymization and security techniques whereby the global deep learning models are trained and the updates are pushed to other interested edge devices. FL benefits in terms of lowering the latency, optimizing the bandwidth and network communication, preserving privacy, and establishing secure data channels. Google uses TensorFlow Federated and TensorFlow Encrypted which are the FL-variants of their famous toolkit for deep learning based applications but a few other famous toolkits for FL include coMind, Horovod, OpenMined, PaddleFL, and Clara Training framework [4]–[8].

Although FL was a novel term and it is being well accepted by academic and industry researchers, however, similar concepts were introduced since the emergence of mobile cloud computing (MCC) back in 2009 [9]. Gradually, MCC transformed from two-tier computing architecture to a three-tier architecture whereby a third layer is embedded to replicate the centralized remote cloud services in the proximity of edge devices in order to minimize the latency, reduce the bandwidth consumption and enable privacy preserving local analytics [10], [11]. A few different variants of these new three-tier architectures were named as mobile edge computing (MEC), multi-access edge computing and fog computing. However, there objective remained same *i.e.*, to provide latency-minimal and communication-efficient application execution environments at the one-hop wireless distances from edge devices. Google's FL framework, caters the needs of MCC applications only whereby the model updates are shared between edge devices for local analytics and their cloud environments for cloud-based analytics [2]. However, in our previous research works we developed three-tier analytic-rich architectures, namely UniMiner [12] and RedEdge [13], with primary objectives of communication efficiency and three-tier analytics capabilities for local analytics (*i.e.*, on-device), collaborative analytics (*i.e.*, between edge devices via same edge server), and cloud analytics (*i.e.*, to ensure global analytic and knowledge discovery via cloud servers). This fine-grained FL, as depicted in Fig. 1, brings more flexibility and fine-grained knowledge availability for MEC application users.

Despite fast acceptance of FL, both classical and fine-grained FL need to address a few pressing issues to realize the collaborative learning applications. Edge devices normally operate in heterogeneous environments whereby heterogeneity arises at all levels of MEC in terms of battery power, sensors and their data collection settings, types of data sources, processing capabilities, communication interfaces, data being generated, types and granularity of learning models, learning rates of applications, sparsity in datasets, frequency of incoming data streams, missing and noisy data, and application of statistical inferencing techniques [14]. In addition, the mobility and limited battery power bring the issue of asynchronization in FL model training whereby devices abruptly leave the model training processes and either completely fail or delay in reporting the local updates. This asynchronization issue results in centralized model training over an obsolete data streams which may not be relevant in certain scenarios

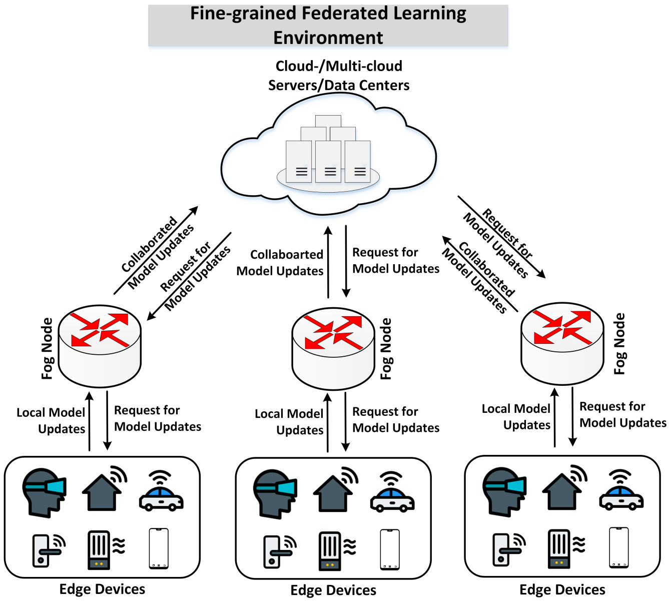
Fine-grained Federated Learning Environment
Cloud-/Multi-cloud Servers/Data Centers
Collaborated Model Updates
Request for Model Updates
Collaboarted Model Updates
Request for Model Updates
Request for Model Updates
Collaborated Model Updates
Fog Node
Fog Node
Fog Node
Local Model Updates
Request for Model Updates
Local Model Updates
Request for Model Updates
Local Model Updates
Request for Model Updates
Edge Devices
Edge Devices
Edge Devices

Fig. 1. Federated Learning vs. Fine-grained Federated Learning.

and applications requiring real-time or near-real-time model updates. The establishment of secure communication channels is another issue because of knowledge sharing among devices. Finally, the incentive mechanisms are required to attract and engage the edge devices to ensure a fully-participating fine-grained FL in MEC systems.

Researchers employed different privacy preservation techniques on local updates before transmitting them to centralized cloud environments [15]. These privacy preservation techniques use differential privacy mechanisms to anonymize the data in compute-efficient way or they use multiparty computation (MPC) techniques to ensure better data reporting because MPC enables multiple participants (*i.e.*, edge devices in the case of FL and edge devices and Fog nodes in the case of fine-grained FL) to collectively report and validate their local updates [15], [16]. However, classical FL models are centralized and prone to centrality attacks which result in compromises over privacy, security, and performance. In addition, there arise the issues of bias and fairness whereby the training strategies are executed by centralized entities who can authoritatively configure FL learning processes to select the specific subsets of samples, populations, instances, communities, and devices who do not, in essence, cover the whole populations under considerations.

Considering privacy preservation requirements and the issues of centralization, fairness, and bias, researchers proposed blockchain-based decentralized FL techniques. Blockchain technologies ensure transparency, decentralization, immutability, and traceability of reported data from multiple edge devices [17]–[19]. In addition, they enable the trust among all participants due to consensus mechanisms and the implicit property of non-repudiation [20]. Existing blockchain-based FL systems, reported in Section II, were deployed in different application domains such as IoT networks [21], handwriting recognition [22], news-feed [23], human activity recognition [24], across entire data pipelines in artificial intelligence applications [16], and distributed learning in 5G networks [25].

The complexities and the involvement of multiple participants in MEC-based fine-grained FL systems cause the heterogeneity in multiple forms (*i.e.*, raw data, pre-processed data, trained models, or deployed models) and at multiple levels (such as users, sensors, devices, data sources, edge devices, fog nodes, blockchain networks, cloud service providers, and application users). This massive heterogeneity creates a pressing demand to design a fully collaborative, trustworthy, and reliable fine-grained FL system. Using an integrated blockchain-based decentralized reputation system could help in ensuring authenticity, traceability, provenance, incentivization, and penalization of all stakeholders in the fine-grained FL environments. To the best of our knowledge, there is no study addressing the issue of fine-grained FL and its integration with blockchain-based reputation systems. Hence, the main contributions of this paper are:

- We formally introduce, depict, and elaborate the concept of fine-grained FL in MEC networks.
- We discuss the performance objectives, highlight the limitations, and define the core requirements of fine-grained FL systems.
- We put forward the concept of blockchain-based reputation-aware FL to design a trustworthy collaborative ML in MEC systems.
- We perform the qualitative evaluation of proposed technique and compare it with state-of-the-art research work.

The paper structure is: Section II discusses related work and section III presents discussion on fine-grained FL. Section IV elaborates the concept of blockchain-based reputation-aware FL in MEC systems and section V concludes the article.

II. Related Work

A few early blockchain-based FL implementations and proposals were presented by researchers recently. BlockDeepNet integrates blockchain and collaborative learning algorithms for IoT applications whereby each IoT device defines its local parameters and train its own deep learning model [21]. The IoT devices in BlockDeepNet share their stochastic gradient descent (SGD) updates for global aggregation in cloud environments via edge servers. BlockDeepNet uses Go-ethereum blockchain on the edge servers for secure and reliable exchange of model updates in the IoT network. DeepChain, uses Corda blockchain smart contracts to incentivize the model sharing participants and ensure security and privacy of shared model updates via blockchain network [22]. DeepChain prototype was implemented and tested using MNIST dataset, which is a large database of handwritten digits, in terms of training accuracy of FL models and encryption strength of shared model updates. However, a thorough investigation with real-time live dataset is necessary to generalize the DeepChain.

An early implementation discusses blockchain technologies and their integration with FL to preserve privacy of shared

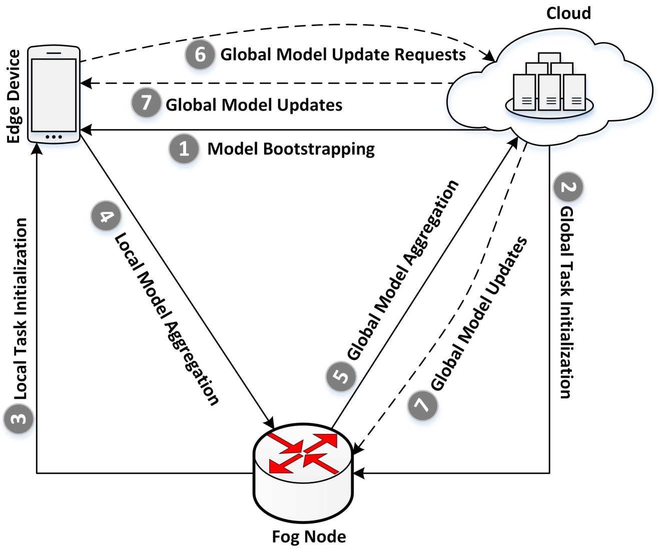
Cloud
Edge Device
6 Global Model Update Requests
7 Global Model Updates
1 Model Bootstrapping
4 Local Model Aggregation
5 Global Model Aggregation
7 Global Model Updates
2 Global Task Initialization
3 Local Task Initialization
Fog Node

Fig. 2. Fine-grained Federated Learning Process.

data. Researchers proposed new consensus mechanism to provide proof of quality of trained models [23]. However, the proposed work is tested on news-feed dataset and there is still a need to test the proposed implementation in industrial environments. Another early implementation used blockchain for FL to preserve the shared data against the privacy breeches of personal data and security attacks by Byzantine devices [24]. Researchers tested their prototype using an activity recognition dataset, however, the implementation and testing in the real-time environment is still missing.

Researchers at IBM considered the heterogeneity across entire data pipeline from selecting the data sources to deploying the learning models [16]. They used blockchain to track the provenance and history of data, learning models, metadata about all relevant activities, and operations and interaction among different participants, however, the study lacks in providing any quantitative evaluation of the proposed methodology [16]. Moreover, PIRATE is another early research proposal for blockchain-based secure distributed learning in 5G networks [25]. A few early proposals for blockchain-based classical FL systems is presented to preserve the data privacy [26] and quality [27]. Finally, the issue of data-poisoning attacks and low-quality data reporting are handled by using reputation systems in the classical FL systems [28]. The reputation system in classical FL systems helped in the selection of reliable data sources and the proposed scheme was implemented using consortium blockchain network. Although the integration of Blockchain, FL, and MEC could potentially become very useful, however, none of the existing research works provide concrete results to address the fine-grained FL related issues.

III. Fine-Grained Federated Learning

We provide the formal definition of fine-grained FL and discuss its associated objectives, limitations, and requirements.

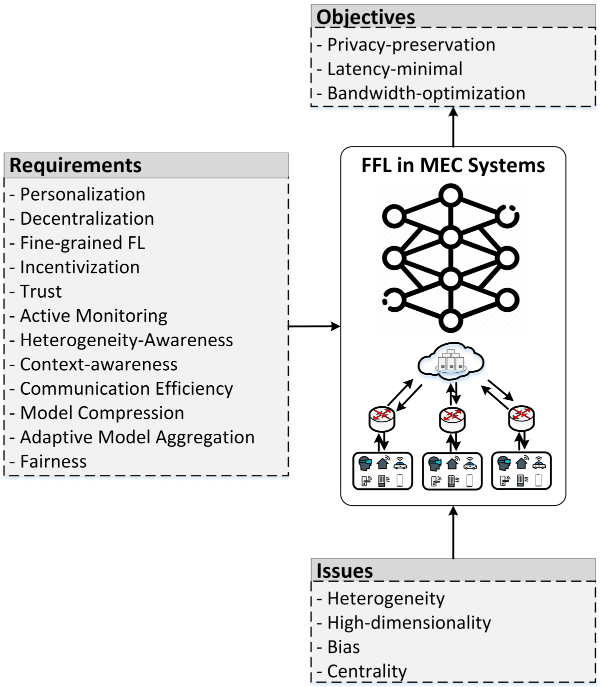

Fig. 3. Limitations, Requirements and Objectives of Fine-grained FL Applications in MEC.

A. Problem Statement

The classical FL ensures privacy-preserving collaborative learning by executing device-first approach whereby data-owners train their local learning models using onboard edge device resources. Later they apply the privacy-preservation techniques and transmit the local model updates (*e.g.*, SGD techniques to update the weights in deep neural networks (DNN)) to centralized cloud servers which execute federated aggregation schemes and update the global learning models. The global model updates are transmitted back to edge devices to update the local learning models. Despite a straightforward execution process and benefits in terms of latency, bandwidth-efficiency and privacy-preservation, the classical FL schemes need to address a few limitations considering, *1)* heterogeneity of devices and servers, *2)* high-dimensionality of data and learning model updates, *3)* bias in terms of data, algorithms, data sources, data preprocessing, and model training, *4)* centrality in model training, and *5)* centralized points-of-failures and compromises. In addition, classical FL schemes introduce centralized global model updates which can facilitate the applications at coarse-grained level, *e.g.*, enabling a general activity detection for whole population of application users on a collaborative social health platform. However, in essence, each mobile user have different activities pattern, *e.g.*, length of footstep, walking speed, sitting postures, and running patterns, which needs to be personalized, at least at the level of a subset of a population.

B. Definition

Early definitions of FL, as given in [1] and [29], enable two-tier collaborative learning process between edge devices and cloud servers. However, with the emergence of MEC, the data requirements and application models are being transformed into three-tier architectures. Considering this opportunity, we re-define the classical FL as fine-grained FL whereby privacy-preserving collaborative learning processes are executed at all three-tiers (*i.e.*, edge, fog, and cloud) of MEC networks. We envision three main entities involving in the fine-grained FL process namely, *1)* data-owners (edge-devices), *2)* data-arbitrators (fog nodes), and *3)* model-owners (cloud servers). Fig. 2 elaborates the fine-grained FL process execution in MEC network. Please note that the solid lines denote the compulsory steps to initiate and execute fine-grained FL process. In general, fine-grained FL training process is based on following seven steps. Given a set of N devices $D = \{1...N\}$, a set of possible proximal n Fog nodes $F = \{1...n\}$, and a set of learning models $L = \{1...n\}$, any arbitrary device D_i must be connected with a F_i in the MEC network and it should be able to execute a given L_i.

- ***Step_1 (Model Bootstrapping)***: All D_i periodically install the updates from global L_i. It is assumed that a D_i has the sufficient onboard resources to execute the given learning tasks using L_i at any instance of time t.
- ***Step_2 (Global Task Initialization)***: Cloud server periodically pushes the updated learning model to F_i. In addition, cloud server delegates the learning tasks (such as hyper-parameters, learning rates, desired accuracy level, optimized/semi-optimized SGDs) to all connected F_i.
- ***Step_3 (Local Task Initialization)***: F_i periodically push the updated local model parameters to connected D_i. In addition, F_i matches the learning tasks with their local L_i. In the case of asynchronized required model updates, the L_i executes Step_5, otherwise, it selects the candidate D_i and delegates the learning tasks to all connected D_i.
- ***Step_4 (Local Model Aggregation)***: Cloud servers request multiple F_i for updated L_i parameters, therefore, latency in MEC networks and later bootstrapping can cause the asynchronized model updates in the fine-grained FL process. Therefore, D_i match the model parameters of all three models before executing their local L_i. In the case of asynchronized model updates, D_i collect the data from onboard sensors and applications and execute the given learning tasks using onboard L_i. The D_i update their local L_i model, apply the privacy-preservation techniques on model updates, and send it to their connected F_i for local aggregation. An F_i applies the local model aggregation algorithms and updates its local L_i parameters accordingly. The F_i appends the privacy information and sends the updated model parameters to cloud server.
- ***Step_5 (Global Model Aggregation)***: Cloud server runs the global model aggregation schemes and update the global L_i. It propagates the L_i updates to all connected F_i in the underlying MEC network.
- ***Step_6 (Global Model Update Requests)***: The asynchronization and multiparty model training requires the D_i to execute updated global model. Therefore, D_i periodically generate the requests for updated global model to update their local L_i.
- ***Step_7 (Global Model Updates)***: The cloud server periodically pushes the updated model parameters to all D_i and F_i in the underlying MEC network.

C. Requirements

Based on the objectives and issues of fine-grained FL in MEC systems and the shortcomings of the related research work, we define, as shown in Fig. 3, some essential requirements in this subsection.

- ***Personalization***: Data-owners share local L_i updates based upon their personal experiences to augment the collaborative learning models in fog nodes and cloud servers. However, local L_i are required to be personalized and resilient to data and model poisoning attacks.
- ***Decentralization***: The centrality in terms of data, data-owners, F_i, and cloud servers leads towards Non-IIDdata and bias FL models. Therefore, decentralization is required among all participants involved in fine-grained FL process.
- ***Fine-grained FL***: The classical FL models provide coarse-grained predictions whereby a centralized global L_i is updated across all the D_i. However, in essence, datasets could be vertically partitioned to get better insights. The MEC enables multi-level data management hence it can facilitate vertical data partitioning at D_i, F_i, and cloud levels. The multi-level partitioned datasets yield in more fine-grained FL training.
- ***Incentivization***: FL application models primarily cater the needs of crowd-sensing applications, whereby decentralized personal datasets are kept on D_i. Therefore, new incentive mechanisms are required to recruit the D_i with high quality data sources.
- ***Trust***: D_i primarily share model updates and metadata which could become more critical in certain scenarios such as image processing in social media applications or bio-markers in healthcare applications. Therefore, the involvement of multiple data-owners, data-arbitrators, and model-owners require a trustworthy, privacy-preserving, and secure environment for all participants.
- ***Active Monitoring***: The participation of D_i in MEC environments is volatile due to limited battery powers and mobility constraints. Therefore, fine-grained FL systems are required to continuously monitor the dropped participants to ensure high quality data collection.
- ***Heterogeneity and Context-Awareness***: The fine-grained FL systems need to handle the heterogeneity at all levels in the MEC systems which may arise due to D_i, F_i, data types, data-sources, and L_i. Moreover the D_i and F_i must be able to infer different situations and establish the right contexts to execute the L_i in their local environments.

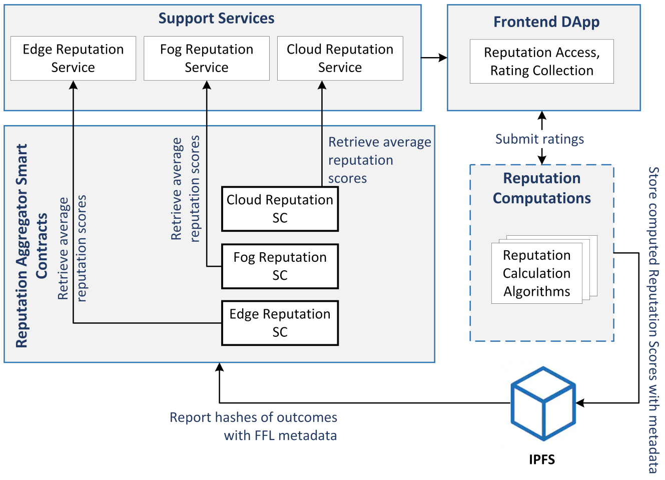
Support Services
Edge Reputation Service
Fog Reputation Service
Cloud Reputation Service
Frontend DApp
Reputation Access, Rating Collection
Submit ratings
Reputation Aggregator Smart Contracts
Retrieve average reputation scores
Retrieve average reputation scores
Retrieve average reputation scores
Cloud Reputation SC
Fog Reputation SC
Edge Reputation SC
Reputation Computations
Reputation Calculation Algorithms
Store computed Reputation Scores with metadata
Report hashes of outcomes with FFL metadata
IPFS

Fig. 4. Blockchain-based Reputation Calculation for Fine-grained FL.

TABLE I
COMPARISON WITH STATE-OF-THE-ART

Requirements	**BlockDeepNet**	**DeepChain**	**This Work**
Personalization	Coarse-grained	Coarse-grained	Fine-grained
Decentralization	D_i & F_i	D_i & F_i	D_i & F_i
Fine-grained FL	No	No	Yes
Incentivization	No	Yes	Yes
Trust Model	Decentralized	Decentralized	Decentralized
Active Monitoring	No	No	Yes
Heterogeneity Awareness	No	No	Yes
Context Awareness	No	No	Yes
Communication-efficiency	Yes	Yes	Yes
Bandwidth Optimization	No	No	Yes
Adaptive aggregation	No	No	Yes
Fairness	No	Yes	Yes
Blockchain Network	Private	Private	Public
Reputation-awareness	No	No	Yes

- ***Communication and Bandwidth-Efficiency***: Proximal D_i in the same environment with same learning task result in generating same model updates. Therefore, sophisticated data reduction, model compression and adaptive model aggregation techniques are required. In addition, the F_i should ensure minimal transient delays to improve the communication-efficiency.
- ***Fine-grained Model Synchronization***: The variations in onboard resources in D_i and F_i lead towards dropped participants and varying execution times which results in asynchronous fine-grained FL environments. Therefore, fine-grained FL systems are required to minimize the delay at all the communication paths to ensure maximum synchronization at all three levels in MEC systems.

IV. BLOCKCHAIN-BASED REPUTATION-AWARE FL

The presence of malicious, faulty, and ghost D_i could become a major bottleneck in achieving fine-grained FL requirements, however, apriori reputation information about D_i can overcomes this bottleneck. Fig. 4 presents a snapshot of blockchain-based reputation system for fine-grained FL.

The access to reputation information is provided to all participants of FL systems via Frontend DApps, which use Ethereum's public blockchain and smart contract technologies to compute and determine trustworthy aggregation of reported reputation scores. D_i although can request, access, and compare the off-chain model parameters from cloud servers as well as F_i and rate their performances, hash them and store in decentralized storage, such as IPFS, in the MEC networks. However, they report the hashes of reputation scores to on-chain smart contracts. The smart contracts then aggregate and calculate the reputation of each F_i and cloud server. Likewise, F_i can rate the performance of connected D_i in terms of data-richness, context-awareness, and ability to provide representative crispy non-redundant model updates in heterogeneous settings, dropped participant ratio, quality of model updates, statistical variations in model updates, and many other performance evaluation parameters. Similarly cloud servers can rate the D_i and F_i in terms of activeness to participate in collaborative model development processes, willingness to share model updates, frequency of model updates, and other performance parameters. Despite varying performance objectives and heterogeneous settings, the need for accurate reputation information remains to ensure trustworthy collaborative FL across MEC environments.

A. *Qualitative Comparison and Evaluation*

Considering two current state-of-the-art and relatively complete studies *i.e.*, BlockDeepNet [21] and DeepChain [22], we present the qualitative comparison of our proposed work in Table I. We found that our proposed system will comply with extraneous and more flexible requirements as it will bring fine-grained personalization whereby the D_i will adapt the L_i based on model updates from all three levels, *i.e.*, D_i, F_i, and cloud servers. BlockDeepNet and DeepChain cater the decentralization at the local dataset levels in D_i, however, our proposed approach will ensure maximum decentralization and it will additionally enable the decentralized global datasets at each F_i. Existing systems such as DeepChain use monetary benefits to incentivize the D_i for active participation, instead, our proposed work will provide reputation-aware incentive models to benefit the honest and high quality D_i and minimize the benefits for dishonest participants in the FL systems. However, we also aim to embed the decentralized trust models to ensure security, privacy, trustworthy L_i training across the MEC systems. The current implementations of blockchain-based FL systems do not monitor the devices actively which results in dropped participants and asynchronized model updates. Therefore, our system will actively monitor the dropped participants and it will execute the proactive model update schemes to ensure maximum synchronization among D_i, F_i, and cloud servers. Existing FL implementations neither infer the contexts nor they cater the multi-level heterogeneity. However, considering three-tier architectures, onboard resources, and data-level heterogeneity, we aim to integrate novel context-aware and heterogeneity-aware FL models. In contrast with two-tier FL models, the communication-efficiency becomes one of the primary challenges. Hence, we aim to optimize the communication model to ensure the latency-minimal fine-grained FL applications in MEC networks. In addition, we also foresee the need for robust and adaptive model compression and aggregation techniques in order to minimize the redundancy and optimize the bandwidth consumption. Last but not the least, we aim to ensure fairness across local and global datasets, D_i, F_i, cloud servers, and local and global L_i. In general, we believe our proposed blockchain-based reputation-aware FL scheme will set a pivot to balance the congregated research works in different domains such as trust models, reputation systems, blockchain, MEC, and federated learning.

V. Conclusion

Federated learning (FL) has drawn a significant attention in recent years and it is being recognised as one of the demanding machine learning technique to preserve privacy in the decentralized datasets. In this paper, we have highlighted the importance of fine-grained FL in order to ensure personalized and fine-grained model training for mobile users. This work is motivated by the recent early adoptions of blockchain technologies for FL schemes and the lack of reputation mechanisms to ensure trustworthy collaborative model training in mobile edge computing environments. In this paper, we proposed the concept of reputation-aware decentralized FL complimented with blockchain technologies. Since the research on blockchain-based reputation systems and FL is still in its infancy, this paper opened a wide range of research questions to motivate interested researchers and practitioners to further investigates this promising research area.

Acknowledgment

D. Svetinovic is the corresponding author. This research work is supported by Center for Cyber-Physical Systems, Khalifa University of Science and Technology, UAE.

References

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *ArXiv*, vol. abs/1602.05629, 2016.

[2] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[3] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu, "Differential privacy-based blockchain for industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2019.

[4] Uber's horovod. [Online]. Available: https://eng.uber.com/horovod/

[5] Openmined. [Online]. Available: https://www.openmined.org/

[6] Paddle federated learning. [Online]. Available: https://github.com/PaddlePaddle/PaddleFL

[7] Federated learning powered by nvidia. [Online]. Available: https://devblogs.nvidia.com/federated-learning-clara/

[8] Tensorflow encrypted. [Online]. Available: https://github.com/tf-encrypted/tf-encrypted/

[9] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, 2009.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[11] M. H. Rehman, A. Batool, and K. Salah, "The rise of proximal mobile edge servers," *IT Professional*, vol. 21, no. 3, pp. 26–32, 2019.

[12] M. H. Rehman, C. S. Liew, and T. Y. Wah, "Uniminer: Towards a unified framework for data mining," in *2014 4th World Congress on Information and Communication Technologies (WICT 2014)*. IEEE, 2014, pp. 134–139.

[13] M. H. Rehman, P. Jayaraman, S. Malik, A. Khan, M. Medhat Gaber *et al.*, "Rededge: A novel architecture for big data processing in mobile edge computing environments," *Journal of Sensor and Actuator Networks*, vol. 6, no. 3, p. 17, 2017.

[14] M. H. Rehman, C. S. Liew, T. Y. Wah, and M. K. Khan, "Towards next-generation heterogeneous mobile data stream mining applications: Opportunities, challenges, and future research directions," *Journal of Network and Computer Applications*, vol. 79, pp. 1–24, 2017.

[15] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, 2019.

[16] K. Sarpatwar, R. Vaculin, H. Min, G. Su, T. Heath, G. Ganapavarapu, and D. Dillenberger, "Towards enabling trusted artificial intelligence via blockchain," in *Policy-Based Autonomic Data Governance*. Springer, 2019, pp. 137–153.

[17] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3548–3558, 2019.

[18] K. Salah, M. H. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for ai: review and open research challenges," *IEEE Access*, vol. 7, pp. 10 127–10 149, 2019.

[19] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7992–8004, 2019.

[20] M. Nassar, K. Salah, M. H. Rehman, and D. Svetinovic, "Blockchain for explainable and trustworthy artificial intelligence," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 1, p. e1340, 2020.

[21] S. Rathore, Y. Pan, and J. H. Park, "Blockdeepnet: A blockchain-based secure deep learning for iot network," *Sustainability*, vol. 11, no. 14, p. 3974, 2019.

[22] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[23] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, 2019.

[24] X. Zhu, H. Li, and Y. Yu, "Blockchain-based privacy preserving deep learning," in *International Conference on Information Security and Cryptology*. Springer, 2018, pp. 370–383.

[25] S. Zhou, H. Huang, W. Chen, Z. Zheng, and S. Guo, "Pirate: A blockchain-based secure framework of distributed machine learning in 5g networks," *arXiv preprint arXiv:1912.07860*, 2019.

[26] S. Awan, F. Li, B. Luo, and M. Liu, "Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2561–2563.

[27] S. Lugan, P. Desbordes, E. Brion, L. X. R. Tormo, A. Legay, and B. Macq, "Secure architectures implementing trusted coalitions for blockchained distributed learning (tclearn)," *IEEE Access*, vol. 7, pp. 181 789–181 799, 2019.

[28] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *arXiv preprint arXiv:1910.06837*, 2019.

[29] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *arXiv preprint arXiv:1909.11875*, 2019.