# Tools and Techniques for Privacy-aware, Edge-centric Distributed Deep Learning

Ziran Min
ziran.min@vanderbilt.edu
Institute for Software Integrated Systems Vanderbilt University
Nashville, TN

Robert E. Canady
robert.e.canady@vanderbilt.edu
Institute for Software Integrated Systems, Vanderbilt University
Nashville, TN

Akram Hakiri
akram.hakiri@issatm.rnu.tn
University of Carthage, ISSAT Mateur
Mateur, Tunisia

Uttam Ghosh
uttam.ghosh@vanderbilt.edu
Vanderbilt University
Nashville, TN

Aniruddha Suresh Gokhale
a.gokhale@vanderbilt.edu
Institute for Software Integrated Systems, Vanderbilt University
Nashville, TN

## ABSTRACT

Training and Inferencing phases of Deep Learning (DL) are compute-intensive that require huge cloud-hosted resources. However, real-time needs of some edge-based applications as well as the variable and wildly-fluctuating edge-cloud latency require new ways to exploit clusters of edge devices in a decentralized and federated manner that perform On-Device or edge-based DL training/inferencing. However, Edge-based DL is fraught with many challenges including the need to discover the right resources, heterogeneity in resource types leading to non-uniform execution times among cluster members, increased incidences of failures and network disconnectivity leading to consistency issues, preserving privacy of data used in DL tasks, the type of distributed DL algorithm used and its performance on the chosen resources, and many others. To address this plethora of challenges, this paper explained the objectives of expected solutions, which include privacy-preserving, inferencing on heterogeneous edge devices and CAP-driven Trade-offs and Blockchain-based Consensus. In the future, we will propose a privacy-preserving, edge-centric federated machine learning solution that blends ideas from model/data-parallel machine learning, resource discovery, distributed consensus using Blockchain, dynamic resource management and use of hardware accelerator devices.

## CCS CONCEPTS

• **Computer systems organization** → *n-tier architectures*; • **Computing methodologies** → **Multi-agent systems**; • **Theory of computation** → Self-organization.

## KEYWORDS

Distributed machine learning, edge computing, consensus, resource management.

## 1 INTRODUCTION

An increasing number of Deep Learning (DL) algorithms run on distributed systems, which offer sufficient cloud-hosted resources. Today's cloud provides multiple machine learning options which help user to design their experiment. However, some edge-based real-time applications can not tolerate the variable and long network latencies between edge nodes and the cloud infrastructure. Edge computing provides a new paradigm to place lightweight cloud resources on clusters of resource-intensive and interactive edge devices, such as autonomous cars, smartphones, medical center servers, pandemic drones, and the emerging IoT [14]. Despite the promise, Edge-based DL still faces many challenges. Consider a disaster response management scenario (e.g. an accident, natural disaster, or terrorism) where first responders need real-time assistance in locating survivors and finding suitable routes for rescuing them. Although deep learning-based models for object detection and image recognition/classification have been developed and a generic model for disaster scenarios can also be pre-built, the vagaries of a given situation will often render these models ineffective thereby requiring model updates using data specific to the situation.

For example, surveillance cameras may have captured images of collapsing structures during an earthquake while any surviving sensors within and around the rubble may still be able to emit useful information such as structural integrity, weight, etc that may provide first responders with the right approach towards sifting through the rubble. Similar other rescue operations can be more effectively planned on-the-fly and executed for a variety of other scenarios, such as flooding or fire. In such situations, given the timeliness constraints as well as the contested operating conditions, it is infeasible to leverage the centralized cloud resources for the model update and inferencing tasks. Rather, there is a need to be able to dynamically discover sensors and edge-based compute resources that can participate in the model update and inferencing tasks. One may assume a command and control station belonging to the first responders that might provide slightly powerful fog-level capabilities than purely edge devices.

Many other applications of deep societal importance can be conceptualized similarly. For instance, the recent surge of the deadliest and most widespread COVID-19 pandemic in over a century has affected all spheres of our daily life. Even as we continue to struggle to contain the current COVID-19 pandemic, the second wave of this life-threatening disease is looming on the horizon. The Covid-19 pandemic has caused havoc around the world with the United States being the most impacted country. Schools and colleges are hoping to reopen in-person by taking necessary precautions. However, in most cases, decisions to revert to online mode are predominantly going to be based on reactive decision-making driven by identified cases and the disease spread. We surmise that proactive approaches may be more effective in handling such situations. For instance, surveillance cameras on campus may be able to spot students not wearing masks and not following social distancing guidelines. Without revealing the identity of the person (i.e., privacy protection), it may still be possible to track these people and determine their whereabouts, e.g., in bars and other crowded places where the chance of contracting an infection is high. Data models can then be used to determine the likelihood and spread patterns of the disease, which then can be used proactively in both contact tracing and quarantining so that the reopening process has a better chance of succeeding. Since most of the tracking and analysis is occurring at the edge, such applications will also have to depend predominantly on edge resources, however, they may be able to leverage somewhat powerful resources such as campus-based compute clusters as fog resources.

## 2 CHALLENGES

Addressing the many requirements of our motivating examples and similar real-world problems is fraught with many technical challenges listed below. Developing a holistic set of solutions to address these challenges all at once is our final goal. Recent prior works have also highlighted synergistic set of challenges and initial ideas [3].

- **Privacy-aware, edge resource discovery**: Unlike cloud resources where users typically use RESTful APIs to request and allocate resources, edge resources are highly distributed and almost never under the control of a central authority. The edge resources typically are deployed to execute application-specific tasks, such as surveillance or air/water quality monitoring. However, these resources should be able to take on additional tasks when the need arises, i.e., spare capacity should be available to execute other tasks, such as model updates and inferencing. However, in disaster scenarios, not all the resources may be operating as some may be damaged thereby reducing their availability. Further, even if a resource is available and were to be discovered, it must be able to execute machine learning tasks. Additionally, continuously training, updating, and inferencing learning parameters in the cloud lead to the leakage of users' private and sensitive data, such as confidential data, fingerprints and face scans. Then, users have to give up their data ownership. That is, the cloud must be trusted in the sense of preserving any privacy constraints (e.g., not leaking users' identity). Thus, novel approaches are needed to discover and deploy trusted edge resources.
- **Cohabiting model update and inferencing with other tasks on heterogeneous edge devices**: The discovered edge resources may not be all the same; in fact it is highly likely that there will be significant heterogeneity among the resources ranging from processing capabilities on cameras, to embedded devices such as Raspberry Pi and Beagle-bones or Smartphones, to even hardware accelerator devices such as Field Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPUs), Tensor Processing Units (TPUs) and Intel Neural Compute sticks among others. Additionally, the number of active devices may influence the size of the network, as many active devices could drop out due to intermittent connectivity or energy constraints. Both model update and inferencing are compute-intensive tasks. Further, edge resources may already by executing other tasks. Finally, different edge resources may illustrate different execution times and energy consumption for the same task. Thus, novel approaches are needed to partition model update and inferencing tasks across the different resources in a way that makes effective run-time trade-offs to balance response time constraints, model fidelity, inference accuracy and task schedulability.
- **Addressing the CAP dilemma and achieving consensus in federated machine learning:** An edge environment typically illustrates substantial uncertainty in resource availability, e.g., network connectivity can fluctuate, or devices may fail. Such environments thus illustrate the classic use case of the Consistency-Availability-Partition Tolerance (CAP) dilemma where applications and systems infrastructure must make effective trade-offs. Furthermore, even though federated learning makes a step towards securing private data locally on each device by sharing model update parameters, i.e. gradient descent information and gradient evaluation instead of raw data [20], however, communicating model updates throughout the training process can nonetheless reveal sensitive information [11].
  Despite Privacy-preservation has been improved using the Machine Learning Differentially Private (MLDP) [6] multiparty data model sharing method, yet the presence of centralized curators in differential privacy protocol increases the risk of data leakage specifically for applications involving distributed multiple parties, such as federated learning [9]. First, because there is a huge amount of aggregated data coming from multiple parties, most of them are often unknown that should be processed by the curators, the risk of data leakage is increasingly prominent. Second, often these multiple parties (including the curators) that share their data do not fully trust others, thus increasing the risk of the data breach. That is, the differential privacy-preserving protocol appears to incur limitations when it comes to sharing data between untrusted parties [19]. Thus, a new collaborative scheme between untrusted federated edge parties is needed to enhance privacy-preservation. Blockchain [8] shows promise in providing consensus capabilities, improve trustworthiness and model validation. Security in Blockchain may be enhanced with Homomorphic encryption (HE) [1] to enable computation on encrypted data without leaking any information about the underlying data. Thus, novel solutions are required to make such trade-offs and achieve consensus.

## 3 APPROACH OBJECTIVES

In this section, we will explain the objectives of expected approaches, which can overcome the challenges that we mentioned before.

## 3.1 Objective 1: Privacy-aware Edge Resource Discovery and Deployment

The proposed framework for privacy-aware edge resource discovery and middleware deployment operates as a three layers architecture as shown in Fig. 1. The top layer is the cloud computing layer which hosts application servers. The middle layer is the edge computing layer that offloads some required computational logic and data to edge network to avoid the transfer of data all the way back to central cloud storage. In this layer, we introduce a domain controller (server) for each domain in the bottom layer. The domain controller has the functionalities depicted in Fig. 2. To avoid reinventing the wheel, we will make every attempt to build our ideas on top of existing coordination mechanisms such as Apache Zookeeper, Consul and etcd. A cluster of fog servers or edge resources will be required to execute these coordination services.

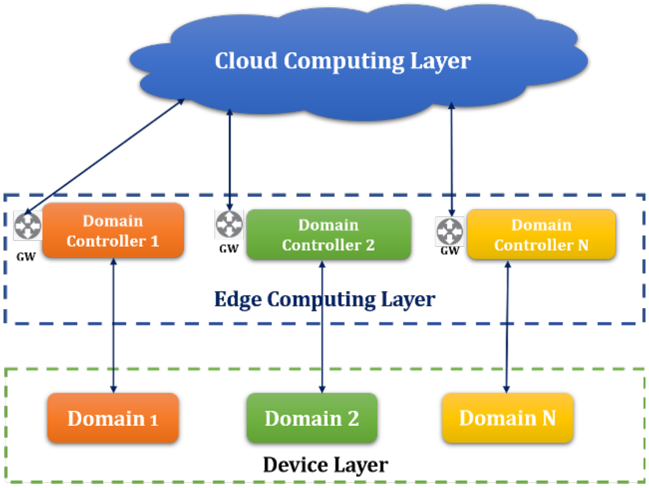

**Figure 1: Proposed Resource Discovery Architecture.**

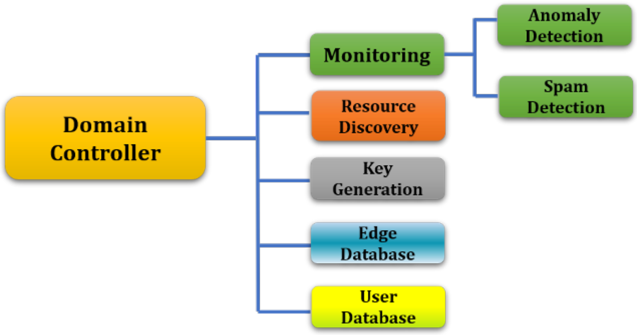

**Figure 2: Functionalities of the Domain Controller.**

- **Monitoring:** The domain controller obtains key information about resources and services from each edge node. Further, it runs deep learning mechanisms to detect anomaly and spam of the device layer.
- **Key generation:** The domain controller enhances privacy and security of the proposed framework by generating a public and private key for all users at the device layer.
- **Resource discovery:** The domain controller is responsible for resource discovery by using a message passing technique. Here, we will use light-weight public key cryptography to provide privacy and security of messages during the discovery process.
- **User database:** The domain controller maintains a user database for the users of each domain by managing their usernames and passwords. The current active users in the domain and their time of activity are logged into this database.
- **Edge database:** The domain controller also maintains a database that stores information of all the discovered edge devices (such as IP address, open port number, name, etc.).

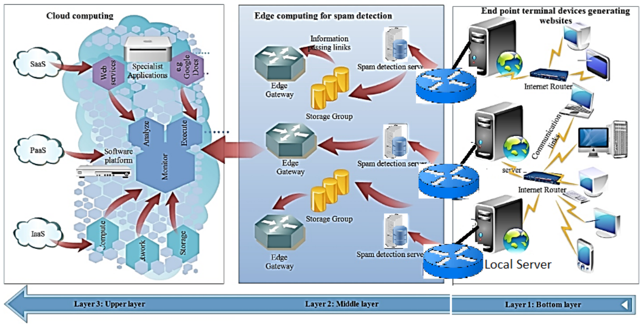

**Figure 3: An example of Spam Detection Framework.**

The main objectives for this dimension of the technical problems are the following:

- We will design a generalized framework leveraging privacy and security for edge computing devices using a machine learning technique and a light-weight cryptography.
- We will propose a web spam detection framework for edge computing in an edge (IoT) environment as depicted in Fig. 3.
- A spam detection algorithm will be developed where the feature extraction will be performed using real-time collected images.
- The framework validation will be conducted using deep learning techniques, i.e., LSTM and CNN with respect to various parameters.

## 3.2 Objective 2: Cohabiting and Distributing Model Update and Inferencing

With the availability of powerful and reliable GPU-accelerated edge AI products, such as NVIDIA's Jetson family (TX2, Nano, Xavier) and Google's edge TPU (Coral), continual learning is now viable using edge devices. In particular, the edge devices are suitable for performing the model update due to the following reasons:

- The computational power of edge devices is enough for updating small to medium-sized DNNs (up to 150 million parameters), such as VGG, Resnet, Inception, and Mobilenet.
- The duration of the model update task is far less than the initial model training time, as fewer full data iterations (epochs) are required.
- Performing model update at the edge avoids costly data transfers to the cloud.
- Using edge devices for the model update also handles data privacy concerns and reduces data security threats.

Training a DL model is a resource-intensive and time-consuming task. Several machine learning frameworks such as MXNet, TensorFlow, PyTorch and Ray run distributed training across two multi-gpu nodes where tasks are divided among multiple workers. Primarily, there are two types of parallelism associated with distributed training: 1) *Model Parallelism* has all workers learn a part of the DL model parameters while working on the complete dataset; 2) *Data Parallelism* involves sharding the dataset among different workers such that each worker learns the complete DL model parameters while working on the part of the dataset. Most available literature focuses on data parallel model learning, however, more recently frameworks such as Horovod [13] are emerging to support model parallel machine learning.

There exists another classification in distributed training based on how the knowledge (i.e. hyper-parameters) learned by individual workers is shared across the group. Most DL frameworks implement either centralized or decentralized architectures for storing and sharing the updated parameters of a DL model. In the centralized architecture, all workers compute forward and backward passes locally and send the gradients to a central entity, called the parameter server, for updating the parameters based on an optimization algorithm such as Stochastic Gradient Descent (SGD). The parameters are then pulled back by the workers to continue the next training step. In the decentralized architecture, no central entity exists; instead the workers exchange among each other the locally learned gradients. Training different hyperparameters can be restrictive, as federated training systems may fail to fully utilize the power of the model distribution due to the limited bandwidth available on those devices. Thus, the decentralized architecture is usually not suitable for the model update at the edge because it incurs high transfer costs.

On another level, there are two types of training loops associated with data-parallel, distributed model update with centralized parameter server: 1) Synchronous training loop, where each worker waits for the others to finish a training step before starting another step, i.e., the training progress is synchronized at each step; 2) Asynchronous training loop, where the training progress is not synchronized, and the parameter server updates the model parameters upon receiving the gradients from each worker. Using an asynchronous training loop is more favorable for the model update at the edge as it avoids the costly synchronization overhead.

Furthermore, resource heterogeneity further complicates the problem. Typically, an equal amount of data is distributed among the workers in multi-machine training. However, this approach can result in a longer time to complete due to the heterogeneity of edge devices. For instance, in our experiments, we observed that NVIDIA TX2 is 30% faster on an average in completing a training step than NVIDIA Nano when updating a state-of-the-art Inception model using the Caltech-256 Object category dataset [7]. Our experiments show the cumulative distribution of time to complete one step, where the average step times for TX2 and Nano are 1.89 and 2.69 seconds, respectively. Thus, equal distribution can lead to underutilization of edge resources. Moreover, the performance of the model update task can be impacted by the state of the node (e.g., CPU, GPU, Memory utilization).

Additionally, our experiments show an initial GPU utilization of 88% and 66% for the two devices increases the average step time by almost 20%. Hence, an intelligent data sharding policy is required by considering the actual states of the workers. This will require lightweight solutions to monitor the state of discovered resources.

Despite these insights, we cannot rely only on device characteristics to decide on the partitioning strategy. Novel approaches are needed to partition model update and inferencing tasks across the different resources, in a way that makes effective run-time trade-offs to balance response time constraints, model fidelity, inference accuracy and task schedulability. Such approaches should (i) anticipate a low amount of participation, (ii) tolerate heterogeneous hardware, and (iii) be robust to dropped devices in the network. There are several resource managers (e.g., Borg [17], Tetricsched [16], and our prior work on Barista [5]) for different kinds of workloads in the cloud environment that perform static and dynamic scheduling. However, most of the approaches do not apply to edge clusters, which illustrate higher levels of edge device heterogeneity. This heterogeneity can be the result of different physical characteristics of the devices, such as number of processors, CUDA cores, memory, etc. or due to the workload associated with the devices.

Another challenge is caused by the performance interference resulting from resource contention due to tasks running in the background. Running resource-intensive model update tasks can cause the background latency-critical tasks (e.g., inferencing) to miss deadlines leading to Service-Level Objective (SLO) violations. Hence, the selection of an edge device to participate in the model update task should be contingent on the latency constraints of the background tasks. Resolving these challenges calls for a custom resource manager for DL model update workloads at the edge by considering the timing constraints of the background applications, the computational capabilities, and workloads of the individual edge devices along with the structure and characteristics of the DL jobs.

## 3.3 Objective 3: CAP-driven Trade-offs and Blockchain-based Consensus

Beyond its use in cryptocurrencies, Blockchain shows particular appeal for federated on-device ML given its emphasis on sharing, distributing, and encrypting device data across geographically distributed zones without compromising their content [18]. Blockchain enables trust-less disintermediation of on-device data making it possible for multiple decentralized islands of devices, who do not trust each other, to exchange digital assets while preserving their anonymity and their privacy from each other. It allows multiple federated on-device ML to agree, at regular intervals, about the true state of shared data [8]. Such shared data can represent credentials and attributes of transactions, information about individuals, entities, etc.

We will address these challenges using blockchain, hybrid consensus algorithm and smart contract technology. In particular, we will focus on the following activities:

- Use blockchain to create distributed and immutable audit trail review for federated model to improve trustworthiness and model validation;
- Enhance blockchain with homomorphic encryption, high-performance data distribution, and scalable communication

between edge nodes and federated model to provide better privacy-preservation; and

- Demonstrate the proposed consensus architecture through the implementation and the deployment of scalable FedML training platform in a fully decentralized, secure and trusted manner, while ensuring anonymity and privacy-preservation as shown in Fig. 4.

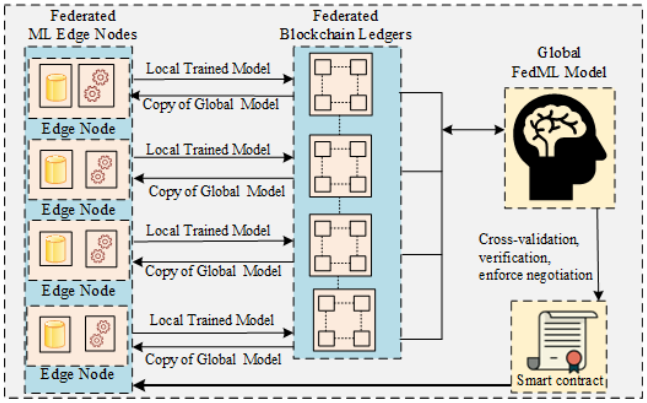

**Figure 4: Federated Machine Learning Architecture.**

### Contribution 1: Hybrid consensus algorithm (CAP-driven Trade-offs and Consensus)

Blockchain shows promise in providing consensus capabilities particularly for the parameter server approach we discussed earlier and when the operating conditions are fluctuating [12]. Nonetheless, a blockchain-based approach must overcome the following challenges, which will be the focus of our investigations:

a) Computation efficiency requires consuming substantially less computing power and less energy for which we propose the asynchronous federated learning scheme with bounded-delays assumptions [10], where the goal is to design and implement an asynchronous federated learning scheme for learning models from the edge data, and further improve the efficiency of federated learning by selecting the participating nodes to minimize the total cost.

b) Shared **data reliability** that must be guaranteed along with robustness and fault-tolerance due to low number of participants, heterogeneous hardware, and dropped devices in the network. To that end, our approach must account for the following:
   - Edge devices have heterogeneous storage, computational, and communication capabilities due to variability in hardware (CPU, memory), network connectivity (i.e. 3G, 4G, 5G, Wi-Fi), and power (battery level) due to which edge devices may be unreliable to their connectivity limitation and their battery level.
   - Current consensus protocols (such as PBFT-like, PoW, PoS, PoET, RAFT, PoA) are not well-suited for improving reliability while providing high performance and increase scalability [4].

   The goal is thus to design a hybrid consensus protocol to better support Reliable, Replicated, Redundant, and Fault-Tolerant as well as providing scalable and higher performance communication support for distributed federated parties.

c) Improvements are needed for scalability by supporting increasing number of participants without overwhelming or imposing undue overhead on the federated system. For example, PoA consensus can be adapted to support lower network overhead and scale up and down of participants compared to traditional PBFT algorithms for which some degree of consistency may have to be sacrificed to keep the network scalable. The research goal will be to propose a high-throughput framework that encompasses a lightweight consensus algorithm to support federated on-device ML and which is configurable to support a range of operating conditions. To that end, we can explore the following possibilities:
   - Both RAFT and PoA should be integrated into a novel hybrid consensus algorithm which maintain the advantages of reliability and fault tolerance in RAFT and the performance and lower overhead in PoA;
   - To maintain the strengths of both approaches in a novel hybrid decentralized and federation-enabled consensus; and
   - finally, we will be using replicated logging approach to supervise anomaly detection in different locations in the network. Thanks to an improved hybrid consensus algorithm that combines both Reliable, Replicated, Redundant, And Fault-Tolerant (Raft) and Proof of Authority (PoA) algorithms.

### Contribution 2: Privacy-preserving homomorphic encryption scheme for the blockchain

Blockchain as a transparent, immutable, and validated-by-design distributed ledger can offer a potential solution to address the key security and privacy-preserving challenges in federated learning, i.e. by providing secure and pseudo-anonymous transactions in a fully distributed and decentralized manner. Blockchain could also enhance privacy-preservation and enforce security and trust between untrusted participants. Owing to the blockchain encryption algorithms (elliptic function and hashing) that hide data content, the contributed models can be distributed in a form of blockchain transactions coded and encrypted inside blocks.

However, smart contracts are often open to public and perform only computations over plaintext data that are available on the Internet so that users can simply access them by selecting the right URL. Data analyzed over edge devices are usually more complex and more sensitive to storage in the public blockchain (e.g., healthcare data). Security in Blockchain should be enhanced with Graph Deep Learning scheme (potentially using Graph Neural Network (GNN)) along with Homomorphic encryption (HE) [15] to enable computation on encrypted data without leaking any information about the underlying data. GNN-based homomorphic encryption schemes will enable users to perform processing on encrypted data and enforce data privacy and data security on local models (parameters).

The integration of Blockchain transactions (i.e. local model parameters) with homomorphic encryption can secure on-device ML data with high privacy in a decentralized mode. That is, HE performs calculations on encrypted data without decrypting it first and data is analyzed without breaching personal privacy (i.e., when decrypting the output is the same as if the operations had been performed on the unencrypted data). The goal will be to investigate:

- How HE methods implemented in existing libraries [2] (e.g. Lattigo, HEAAN, Pyfhel, PySEAL, etc.) can be used to implement secure smart contracts that will compute encrypted data, and how to enhance these libraries to hide inputs to a smart contract?
- How do we keep local model states and parameters hidden from everyone yet ensure program correctness, while executing arbitrary computations (yet consistent and efficient) on encrypted (or hidden) data, i.e. edge users encrypt their private data, send to the cloud to perform global computation (global learning model), and return them back to end users which are only and restrictively able to decrypt these outputs?

**Contribution 3: Blockchain-based data distribution and training platform**

We aim at developing a blockchain-based data distribution and training platform, that will enable all participants to contribute data and train models in fully decentralized, secure and trusted manner, while ensuring anonymity and privacy-preservation. Compared to traditional approaches in distributing learning tasks among federated nodes, blockchain offers efficient federated ML since instead of contributing with raw data, each participant can locally train its model and only contribute with parameters to the blockchain after which the blockchain nodes aggregate these local data to build a global model.

Specifically, we can use blockchain to create an immutable audit trail enabling trust and transparency, privacy-preserving encryption enhancements in communication, and demonstrate the use of smart contracts for automation and governance. We can include the following activities and objectives as part of our investigations:

- Implement and deploy a collaborative federated machine learning platform to support on-device ML using blockchain technology and smart contracts.
- Ensure that the platform enables trust on the produced federated ML model along with fully automated network management and orchestration (offering automated horizontal and vertical scalability) during the computation.

## 4 CONCLUSION

In this position paper, we showed the unaddressed problems in Edge-based DL, which include real-time needs of some edge-based applications, and the variable and long latencies between the edge and the cloud. We also analyzed the potential challenges in exploring Edge-based DL, which include the need to discover the right resources, heterogeneity in resource types leading to non-uniform execution times among cluster members, increased incidences of failures and network disconnectivity leading to consistency issues, preserving privacy of data used in ML tasks, the type of distributed ML algorithm used and its performance on the chosen resources, and many others. We pointed out the objectives of the expected approaches. In the future, we will demonstrate a systematic approach to real-time, privacy-aware distributed machine learning on heterogeneous edge devices and in the process also demonstrate the novel application of blockchain and distributed consensus in AI/ML.

## ACKNOWLEDGMENTS

This work was partially funded by the Tunisian Ministry of Higher Education and Scientific Research (MES) under the Young Researchers Incentive Program (19PEJC09-04) and the AFOSR DDDAS FA9550-18-1-0126. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of MES or AFOSR.

## REFERENCES

[1] 2019. Chapter Ten - Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities. In *Role of Blockchain Technology in IoT Applications*, Shiho Kim, Ganesh Chandra Deka, and Peng Zhang (Eds.). Advances in Computers, Vol. 115. Elsevier, 293 – 331.

[2] Rashmi Agrawal, Lake Bu, Alan Ehret, and Michel A Kinsy. 2020. Fast Arithmetic Hardware Library For RLWE-Based Homomorphic Encryption. *arXiv preprint arXiv:2007.01648* (2020).

[3] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed. 2020. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Access* 8 (2020), 140699–140725.

[4] Seyed Mojtaba Hosseini Bamakan, Amirhossein Motavali, and Alireza Babaei Bondarti. 2020. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications* (2020), 113385.

[5] Anirban Bhattacharjee, Ajay Dev Chhokra, Zhuangwei Kang, Hongyang Sun, Aniruddha Gokhale, and Gabor Karsai. 2019. BARISTA: Efficient and Scalable Serverless Serving System for Deep Learning Prediction Services. *2019 IEEE International Conference on Cloud Engineering (IC2E)* (Jun 2019).

[6] M. Gong, Y. Xie, K. Pan, K. Feng, and A. K. Qin. 2020. A Survey on Differentially Private Machine Learning [Review Article]. *IEEE Computational Intelligence Magazine* 15, 2 (2020), 49–64.

[7] Gregory Griffin, Alex Holub, and Pietro Perona. 2007. Caltech-256 object category dataset. (2007).

[8] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[9] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang. 2020. Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics. *IEEE Transactions on Industrial Informatics* 16, 3 (2020), 2134–2143.

[10] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. 2020. Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles. *IEEE Transactions on Vehicular Technology* 69, 4 (2020), 4298–4311.

[11] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*.

[12] Md Sadek Ferdous, Mohammad Jabed Morshed Chowdhury, Mohammad A Hoque, and Alan Colman. 2020. Blockchain Consensus Algorithms: A Survey. *arXiv* (2020).

[13] Alexander Sergeev and Mike Del Balso. 2018. Horovod: fast and easy distributed deep learning in TensorFlow. arXiv:1802.05799 [cs.LG]

[14] W. Shi, G. Pallis, and Z. Xu. 2019. Edge Computing [Scanning the Issue]. *Proc. IEEE* 107, 8 (2019), 1474–1481.

[15] Rakesh Shrestha and Shiho Kim. 2019. Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities. In *Advances in Computers*. Vol. 115. Elsevier, 293–331.

[16] Alexey Tumanov, Timothy Zhu, Jun Woo Park, Michael A. Kozuch, Mor Harchol-Balter, and Gregory R. Ganger. 2016. TetriSched: Global Rescheduling with Adaptive Plan-Ahead in Dynamic Heterogeneous Clusters. Association for Computing Machinery, New York, NY, USA, 16.

[17] Abhishek Verma, Luis Pedrosa, Madhukar R. Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proceedings of the European Conference on Computer Systems (EuroSys)*. Bordeaux, France.

[18] Q. Wang, Y. Guo, X. Wang, T. Ji, L. Yu, and P. Li. 2020. AI at the Edge: Blockchain-Empowered Secure Multiparty Learning with Heterogeneous Models. *IEEE Internet of Things Journal* (2020), 1–1.

[19] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor. 2020. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

[20] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2 (2019).