BAFFLE : BLOCKCHAIN BASED AGGREGATOR FREE FEDERATED LEARNING

Paritosh Ramanan [1] **Kiyoshi Nakayama** [2] **Ratnesh Sharma** [2]

ABSTRACT

A key aspect of Federated Learning (FL) is the requirement of a centralized aggregator to select and integrate models from various user devices. However, infeasibility of an aggregator due to a variety of operational constraints could prevent FL from being widely adopted. In this paper, we introduce BAFFLE, an aggregator free FL environment. Being powered by the blockchain, BAFFLE is inherently decentralized and successfully eliminates the constraints associated with an aggregator based FL framework. Our results indicate that BAFFLE provides superior performance while circumventing critical computational bottlenecks associated with the blockchain.

1 INTRODUCTION

Federated Learning (FL) (Konecny et al., 2016) is a distributed machine learning paradigm that has been designed with the primary purpose of preserving data privacy. FL paradigm accomplishes large scale learning tasks (Bonawitz et al., 2019) with data sets fully localized on end user devices ensuring data privacy. Computationally, the entire FL process is divided into rounds. In each round, an aggregator selects a set of user devices and integrates copies of their locally trained machine learning model into a globally held model.

A central assumption of the FL paradigm is the presence of the aggregator meant to coordinate the global computational progress. An aggregator discharges four main functions in the FL paradigm. First, it is responsible for delineating the global computational process into distinct rounds. Second, it maintains a global estimate of the machine learning model which is updated after every round. Third, the aggregator is responsible for selecting the devices and sending a copy of the global model estimate to each. Lastly, the aggregator is responsible for performing the critical step of updating the global model estimate with the aggregate of the selected local copies.

Despite its privacy benefits, the FL paradigm faces numerous computational and operational challenges. First, in FL applications, end users are assumed to trust their aggregator's ability to carry out the selection and aggregation of local models in a fair and impartial manner. However, such an assumption could be highly misplaced since an aggregator can potentially exhibit bias towards a select few user devices thereby poisoning the FL process (Bagdasaryan et al., 2018). Second, the scope of FL is restricted to applications where a centralized aggregator is orchestrated on the cloud (Konecny et al., 2016). As a result, an aggregator's resiliency depends on the robustness of the cloud infrastructure. Owing to a rather centralized nature of cloud based systems, a failure at the cloud level could precipitate the collapse of the entire FL mechanism as well. Lastly, the same cloud based computational framework needs to be individually instantiated for every FL task which in turn leads to separate instantiations of the centralized aggregator as well. Therefore, a scalable, cloud based FL framework for a single learning task demands a high barrier for participation in terms of resources and expertise. Such restrictions render the benefits of FL inaccessible to micro enterprises and social organizations which might lack the scale and sophistication to launch and manage their own large scale FL tasks.

In this paper, we investigate the use of blockchain as a means to eliminate the computational and operational limitations induced by a cloud based aggregator. Our framework uses blockchain based Smart Contracts (SC) that are designed to generalize the concept of transactions in cryptocurrencies using distributed ledger technologies (DLT). SCs are written in specialized Turing complete languages and are executed on top of the existing cryptocurrency layer through a blockchain based virtualized environment (Christidis & Devetsikiotis, 2016). Therefore, by their very design, such virtualized environments offer a versatile platform on which SC driven decentralized applications (dApps) can be created, tested, and deployed. Internally powered by a peer-to-peer

[1]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA [2]NEC Laboratories America Inc., San Jose, CA, USA. Correspondence to: Paritosh Ramanan <paritoshpr@gatech.edu>, Kiyoshi Nakayama <knakayama@nec-labs.com>.

consensus protocol of the underlying blockchain, such a platform becomes ideal for implementing a purely decentralized FL framework.

Our approach leverages SCs to effectively decentralize the rounding, selection and aggregation mechanisms of the FL paradigm. We assume that the global model copy and the associated computational state can be maintained on the SC which results in numerous interesting benefits. First, based on the global state of all peers, user devices could self determine their selection in rounds. Second, the global model copy can be used by the selected devices to perform the aggregation step autonomously in every round. Lastly, the selected devices can also update the global model independently, thereby driving the global computational progress forward. The benefits stemming from our approach imply that the aggregation, selection, and delineation of rounds is completely spearheaded by the devices themselves with the SC merely acting as a facilitator. Therefore, a blockchain based decentralized approach effectively renders the entire FL process aggregator free.

Owing to our blockchain based approach the fundamental limitations of trust, resiliency, and accessibility in aggregator driven FL frameworks can be completely eliminated. First, the round delineation, selection, and model aggregation occurs in a fully decentralized manner helping restore trust. Powered by the underlying consensus protocol, the SC helps ensure transparency, fairness, and impartiality in the FL process (Christidis & Devetsikiotis, 2016) Second, blockchain automatically introduces fault tolerance by its very nature which delivers resiliency in computation. Lastly, owing to their deployment on the blockchain itself, a single SC can coordinate round delineation, selection, and aggregation phases for multiple FL tasks simultaneously from varied sets of user devices. Due to such versatility, a blockchain based FL framework saves on cloud based setup and operational costs as well as eliminating expertise and resource requirements to maintain a cloud driven application (Khajeh-Hosseini et al., 2010). Such an operational benefit lowers the entry barrier for smaller players and improves accessibility significantly.

From a social standpoint, a lower entry barrier for adopting FL could have far reaching community centric benefits as well. Micro scale organizations part of the same community can use existing SCs on public blockchains to self organize and leverage FL among their peers. In doing so, each organization in a community can preserve their own data privacy but yet collaborate on building a global model that helps address challenges common to the entire community. Therefore, a blockchain based FL framework can be used to empower communities of users who would otherwise not have the capability to obtain robust machine learning models for their own internal challenges.

However, the usage of SCs requires careful consideration of the computational constraints imposed by the blockchain. First, storage of data and computation on SCs incurs miner fees for sustaining the consensus protocol. Second, pushing an entire machine learning model to the SC becomes computationally bulky potentially incurring heavy communication latency. Lastly, there are limits on transaction size imposed by the blockchain protocols that restrict the amount of data that can be stored and updated on blockchain in a single transaction. These computational constraints place limitations on the model aggregation and update process in FL.

In this paper, we propose BAFFLE, a blockchain based aggregator free FL environment. We show that BAFFLE is able to successfully eliminate the limitations of the aggregator driven FL paradigm in a computationally sound manner. We devise a budgeted approach to model update and aggregation and leverage SCs to delineate the rounds. We theoretically show that a classical FL scheme is equivalent to a BAFFLE driven approach with a linear relation between the respective learning rates. We provide a practical, production level implementation of BAFFLE on a private Ethereum network, with Solidity powered SC deployments. We demonstrate the merits of BAFFLE on a real world case study using a large Deep Neural Network(DNN) model. Based on our case study, we perform exhaustive experiments to study the benefits, robustness and scalability of BAFFLE compared to other benchmarks. Our results indicate that BAFFLE provides superior computational performance despite the highly restrictive constraints imposed by the blockchain.

Our paper is organized as follows. In Section 2 we provide an overview of related work pertaining to the fields of blockchain and decentralized ML. Section 3 deals with the various paradigms that govern decentralized computation over the blockchain. Section 4 discusses the novel strategies employed in BAFFLE to circumvent the restrictions imposed by the blockchain. Section 5 provides an overview of the local and global computational perspectives of BAFFLE. Section 6 introduces a real world case study of improving driver revenue where an aggregator free FL mechanism could be highly beneficial. Section 7 deals with the entire set of experiments and their analysis. We conclude the paper in Section 8 in addition to providing a quick overview of future work.

2 Related Work

Improving a global neural network model using distributed data with a privacy-preserving purpose was first studied in (Shokri & Shmatikov, 2015). The authors provide a scheme of jointly learning an accurate model by multiple parties for a given objective. More specifically, they consider a global

shared memory model where parameters of the global model are held. Various agents participating in this framework can update a random subset of global parameters based on their local training.

Federated Learning was later proposed in (McMahan & Ramage, 2017; Konecny et al., 2016) with theoretical basis explored in (Konečnỳ et al., 2015). The authors provide an effective method for building collective knowledge across a set of devices while preserving their individual autonomy and privacy. There are ongoing efforts to scale up the FL framework as presented in (Bonawitz et al., 2019). The framework considers multiple aggregators headed by a master in order to manage the entire FL process. Although the work proposes a distributed network of aggregators coordinated by a master, it is not inherently aggregator free.

(Lalitha et al., 2018; 2019) propose a framework of fully decentralized FL in which users update their belief by aggregating information from neighbors. While the theoretical aspect of decentralized FL is explored in these works, numerous system and architectural issues persist in achieving true decentralization. As a result, such systemic issues need to be dealt with in order to obtain a FL framework that is feasible under practical settings.

Practical efforts to integrate AI onto the blockchain are largely confined to white paper proposals without any tangible real world implementations available. The framework proposed in (Chen et al., 2018) designs an SC based machine learning platform allowing users to upload tasks as well as contribute models to solve existing tasks. A distributed, AI computing platform has also been proposed in (Yong et al., 2017) where mining nodes earn their income from processing AI models.

There are also several projects that integrate federated learning into blockchain technologies. The work done in (Kim et al., 2018) supports implementing the FL framework into the mining mechanism of the underlying blockchain platform. However, owing to modification requirements to the underlying consensus protocols such approaches tend to be cumbersome to implement on off the shelf blockchain platforms. The work done in (Harris & Waggoner, 2019) proposes and implements a decentralized AI framework using the blockchain. However, a key requirement of this framework is that training data from devices needs to be published on the blockchain. As a result, the data privacy benefits of FL paradigm is eliminated. In fact, the authors note that a decentralized, blockchain based AI framework with full user data privacy is a key component of their future work.

Despite the above mentioned attempts, a concrete, practical framework for realizing decentralized aggregator free FL is so far lacking both in research and in industrial domains. To the best of our knowledge, BAFFLE is the first production-level decentralized FL platform that could run over existing blockchain networks such as Ethereum.

3 Blockchain based Decentralized Computation

We now proceed to discuss the foundations of the blockchain that influence the design of BAFFLE. The entire blockchain paradigm can be divided into three distinct parts that are relevant to our discussion of aggregator free FL.

3.1 Distributed Ledger

Blockchain (Wood et al., 2014), which was intruduced as a core technology of Bitcoin (Nakamoto et al., 2008), is one form of peer-to-peer (P2P) distributed ledger technologies (DLT) forged by consensus, combined with a system for smart contracts and other assistive technologies. A distributed ledger is basically an immutable database that resides across several or a number of nodes with computing devices and memories. Each node possesses an identical copy of the ledger of transactions, the record of events cryptographically secured with a digital signature that is verified, ordered, and bundled together into blocks, formed in the blockchain. Individual nodes of the network update their ledger independently.

3.2 Consensus Protocols

The important feature of decentralization by blockchain is that the ledger is not maintained or managed by any central authority. Updates to the ledger are independently constructed and recorded by each mining node in a decentralized fashion. Changes to the DLT are approved based on a vote from a majority of the mining nodes. While there are several consensus protocols, we use Proof-Of-Authority (PoA) for our implementation of BAFFLE. A discussion of consensus protocols can be obtained in Section A.2

3.3 Smart Contracts

Smart Contracts (SCs) are self-executing contracts with the terms of the agreement being directly written into lines of code. SCs reside on the distributed ledger and as a result they are available across the entire blockchain network. They are deployed on the blockchain using the consensus mechanism and referenced by a unique address. SCs are written in contract-oriented languages such as Solidity and any change made to the SC needs to also be validated by the blockchain's consensus mechanism. SCs prove to be highly versatile and can define data, functions, and conditions and allow anyone having access to their address to interact and modify their contents.

3.4 Gas Costs

In blockchain platforms such as Ethereum, the concept of gas plays an important role. Gas value actually signifies the standardized remuneration that miners receive in return for validating a piece of SC code wrapped around a blockchain transaction. As a result, Gas is deemed a standard unit of measuring the computational effort required to execute SC code on the blockchain.

We now discuss specific blockchain system constraints that prove challenging for an aggregator free FL mechanism. In addition, we also describe the key tenets of our solution methedology that allows us to circumvent these constraints.

4 Smart FL Contract Design: Decentralizing Role of Aggregator

As mentioned in Section 1, a number of technical aspects need to be considered in order to make the FL process aggregator free. In this section, we examine the salient features of BAFFLE that allows us to circumvent blockchain based system constraints without compromising on solution quality. Even though BAFFLE has been implemented and evaluated on the Ethereum platform, the same technical principles would extend over to other blockchain based SC platforms as well.

4.1 Chunking

Most blockchain platforms have an upper limit pertaining to the data size of each transaction. For the Ethereum Virtual Machine (EVM) with the version we have used, this limit has been set to 24 kB by default. Such a limitation immediately results in a bottleneck for an aggregator free FL scheme since the underlying machine learning models are usually significantly larger than the transaction limit sizes. Such a system induced constraint necessitates the need for partitioning the machine learning model weight vector into numerous *chunks* such that each chunk size is less than the maximum transaction size.

However, chunking in turn introduces a few other notable aspects with regards to model sharing.

Serialization: Since storage on the SC is expensive, the machine learning model needs to be stored in a serialized format. However, partitioning the model after serialization leads could lead to inconsistencies. Therefore, for a specific FL task, it is important to first generate a partitioning scheme that must be used by all agents followed by individual serialization of the chunks. Such a *chunk-and-serialize* scheme has numerous benefits. First, the chunks can be read to and written from independently and seamlessly. Second, such an independence among chunks can be exploited for parallel updates from multiple devices at the same time. Lastly, a chunk independence scheme also leads to a potential scoring technique wherein parts of the model can be evaluated for their worth.

Budgets: A potential benefit of chunking is that user devices are empowered to decide their levels of contribution individually. Since, pushing chunks on the blockchain involves a computational cost as well as miner fees, users can independently evaluate their own cost to benefit ratio and decide the number of chunks that they wish to update in a round. The maximum limit on the number of chunks that a user device wishes to update is referred to as the budget for that device. As a result the set of budget values from all user devices can be heterogenous in nature.

4.2 Scoring and Bidding

Each chunk is assigned a score by the end user devices themselves based on a norm difference with respect to the latest available global copy. Depending on a random selection the user device submits bids on a set of chunks as allowed by the budget limit. The SC receives bids on a diverse set of chunks from different user devices in every round. For chunks on which multiple bids were submitted, the device submitting the maximum score is chosen as the sole updater of the chunk.

4.3 Delineation of Rounds

Owing to the decentralized nature of our approach, the onus of delineating the rounds rests with the end user devices themselves aided by the information maintained on the SC. Specifically, a Participation Level (PL) is chosen for every FL learning task which specifies the number of agents which must have submitted their bids in order for the round to start. Once the participation level criteria is met, the round begins and no new devices are allowed to participate. Devices upload the chunks on whom their bids were accepted and proceed to signal a close of their round.

5 Computational Perspectives in BAFFLE

The entire aggregator free blockchain based FL paradigm presented by BAFFLE can be viewed in terms of two important perspectives. The computational steps undertaken by the user devices in their interaction with the blockchain network forms the local perspective. The global perspective details the computational picture pertaining to the role of the SC in orchestrating the BAFFLE framework.

5.1 User Level Actions: The Local Perspective

Locally, model training and aggregation form the two important steps that every user device participating in BAFFLE

must undertake.

5.1.1 *Local Training*

User devices continuously observe new data points from their environment which can be leveraged for the FL task at hand. The user device pulls the latest available model from the blockchain and performs an average with its latest available local copy. The resulting model is used to train on the locally available data to yeild the new local copy.

5.1.2 *Model Aggregation and Update*

In order to aggregate with the other devices and push its update to the chain, every agent considers the local model copy obtained after local training. The steps taken by the user device for model aggregation and update can be traced with the help of the flowchart depicted in Figure 1 and summarized concisely in Algorithm 1. Each user device is initialized on the basis of the same given partition scheme. As soon as local training is complete, user devices average with the global model copy and check for the round status. In case a round is already underway and thus inactive, the user device returns to the task of collecting new data. If a round is active and accepting bids, devices choose randomly from their local chunks based on their budget size. A scalar score is assigned to each chunk based on the norm of the difference of the local weights with the global weights copy. These scores form the basis of the bid submitted to the SC which decides on which user device gets to update which chunk.

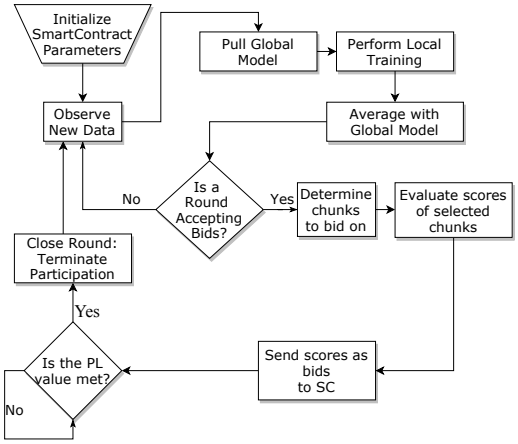


Figure 1. Flowchart depicting the sequence of events at the agent level

5.2 Global Perspective: The Overall Picture

Globally, the computational process employed by BAFFLE is divided into three distinct phases. We illustrate the global computational perspective with the help of an example shown in in Figure 2. In our example we consider a BAFFLE system comprising of 5 asset devices A1, A2, A3, A4, A5 respectively. The model is divided into 5 chunks C1, C2, C3, C4 and C5. For this example, we consider a PL value of 4. In Phase 1, each device performs local training and aggregation to generate new bids. Next, every device attempts to submit bids for its randomly chosen chunks. The bids chronologically arrive in the order A1, A3, A5, A4 and A2. In Phase 2, owing to the PL value being met with the arrival of bids from A4, A2 is rejected from the current round. The accepted devices push the respective chunks for which their bids were accepted. In Phase 3, every device eventually signals the culmination of all its local steps to the SC to mark the end of Phase 3 as well as the current round.

Algorithm 1 Agent based SC interaction

```
for k = 0 ... do
  if round is open for participation then
    choose chunks C̃^k ⊆ C, |C̃^k| = B^j randomly
    calculate scores δ_c = ||Q^c − Q^{j,c}_{k+1}||, ∀c ∈ C̃^k
    submit bids [c, δ_c], ∀c ∈ C^k to SC
    determine accepted chunk set C^k ⊆ C̃^k
    push C^k to blockchain
  end if
end for
```

From a theoretical perspective, using lemma 1, we show that the global computational process is equivalent to classical FL scheme with a learning rate that varies by a constant factor.

Lemma 1. *A learning rate of η_{BFL}, of a model comprising of C total chunks, with L participating agents possessing a maximum budget potential of $B \leq C$ is equivalent to a classical Federated Learning scheme with learning rate η_{FL} such that:*

$$\eta_{BFL} = \frac{2.C.(C-B+1)\alpha_{FL}}{B.L.\alpha_{BFL}}.\eta_{FL}$$

where $\alpha_{FL}, \alpha_{BFL}$ are the probability that an agent is selected for model aggregation for aggregator free decentralized FL and the classical FL respectively.

Proof. Proof provided in Section A.1 □

6 Case Study: Improving Taxi Driver Revenue with BAFFLE

A key problem in the taxi and ride sharing industry is to improve driver revenue by reducing idle time (Han et al., 2016). Drivers are often unable to find passengers at certain locations in the city at varying points of time during the day due to low demand (Han et al., 2016). As a result, they usually hover around the same location until they find a

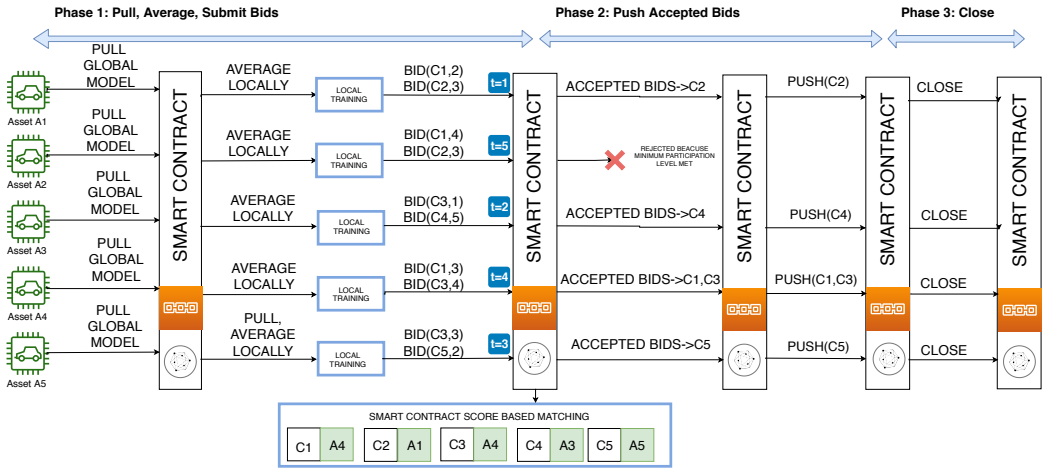


Figure 2. Global computational steps

passenger. Idling time reduces vehicle utilization and leads to potential loss in revenue for the individual driver (Verma et al., 2017).

The application of machine learning to improve driver revenue by reducing idle time has been studied before (Han et al., 2016; Verma et al., 2017; Shi et al., 2018). Based on existing work, a Deep Reinforcement Learning (DRL) scheme is demonstrated to provide good quality improvement in driver revenues (Shi et al., 2018). However, these approaches assume the presence of a centralized coordinator to steer the RL process. A central repository of ride information presents several privacy issues which have been successfully exploited to de-anonymize passenger information (Douriez et al., 2016). The work done in (Shi et al., 2019) as an extension of (Shi et al., 2018) introduces privacy preserving features and distributed computation as a means to improve driver revenue. However, (Shi et al., 2019) assumes a hierarchical computational setup that prevents all the benefits of decentralized computations from being realized in their entirety. The requirements of multiple control centres to perform the learning tasks leads to limited applicability of such approaches.

6.1 Benefits of Aggregator Free FL for Improving Driver Revenue

The taxi and ride sharing industry is a perfect example of micro scale enterprises that could benefit significantly from an aggregator free FL approach. The ride sharing and taxi industry remains largely an unorganized market where setting up a trusted coordinator remains a challenging proposition. Even in case of a central data repository, extracting intelligence from the anonymized data proves to be a futile exercise (Douriez et al., 2016). Moreover, drivers usually also do not have access to sophisticated computing platforms on which they could orchestrate learning tasks to improve their revenue. Therefore, a decentralized aggregator free FL environment allows drivers to leverage their collective ride experiences and improve their revenue without sharing their private ride data itself.

6.2 Deep Batch Reinforcement Learning for Taxis

We use a batch DRL paradigm to learn the Q function values and employ the Deep Neural Fitted Q (Riedmiller, 2005) method to accomplish our learning task. Specifically, we define our states and actions as follows:

- Pickup State s_i:<*pickup_location, pickup_time* >
- Dropoff State s_i':<*dropoff_location, dropoff_time* >
- Action *a: action (dropoff_location)*
- Reward *r: fare*

State is defined by $S \times T$, where S is set of discrete cells that divide the city into distinct grids. T is set of 96 discrete intervals of 15 mins each for 24 hours. Therefore, given N rides, we denote the ride set $\mathcal{H} = \{(s_i, a_i, s_i', r_i), \forall i \in \{1, \ldots, N\}\}$.

$$\tilde{Q}_k(s_i, a_i) = r_i + \gamma \max_b Q_k(s_i', b), \forall i \in \mathcal{H} \quad (1a)$$

$$Q_{k+1} \leftarrow \tilde{Q}_k - \eta \nabla \tilde{Q}(s_i, a_i) \quad (1b)$$

Equations 1a and 1b govern the functioning of the batch DRL framework at the k^{th} round. The Q function is updated based on Equation 1a before being trained on the DNN using Equation 1b.

In Algorithm 2, we consider P taxis and begin by initializing all user devices to the same initial state. Next the partition

Algorithm 2 BAFFLE for Improving Driver Revenues

for taxi: $j = 1 \dots P$ **do**
  initialize model $Q_0^j = Q^{init}$, budget B
  initialize chunk set $\mathcal{C}$ based on given partition scheme.
  for $k = 0 \dots$ **do**
    observe new ride set $\mathcal{H}^k$
    pull latest available model Q from blockchain
    perform averaging $Q_k \leftarrow \frac{Q_k^j + Q}{2}$
    update $\tilde{Q}_k^j$ based on Equation 1a
    locally train Q_{k+1}^j via Equation 1b
    employ Algorithm 1 to push updates to SC
  end for
end for

information and SC details is loaded on each device. The user devices utilize a new set of rides accumulated locally in every round. The local estimate of the Q function is updated and trained locally based on Equations 1 before being pushed onto the blockchain using Algorithm 1.

6.3 Data and Benchmarking Techniques

For our case study, we used the NYC taxi data set (Travel & of New York) for our experiments. Specifically, we randomly chose 2 million rides pertaining to May 2018 which was divided into two equal parts to denote the training and testing data sets. Restricting the rides specifically for the area of lower Manhattan resulted in approximately a little more than half million rides each in training and test data sets. The training set was used to assign rides to taxis participating in the FL process.

On the basis of the test set, we determine 50 taxi trajectories which form benchmark for FL tasks based on work done in (Verma et al., 2017). Each trajectory comprises of 50 rides and assumes idling in case no ride is found. The sum total of fares accrued from the 50 benchmark trajectories is referred to as the Aggregated Simulation Revenue (ASR) which forms the *No Learning (NL)* baseline for our case study.

The benchmark trajectories and the accompanying simulation procedure are also used to calculate ASR values for various DRL models as well. However in this case, instead of hovering in the same location upon not finding a ride, the DRL model in question is used to determine a new location to transition into (Verma et al., 2017). The sum total of fares from the ensuing trajectories denotes the ASR value for the DRL model being considered. For robustness purposes, we perform this simulation multiple times for any DRL model and report the average ASR value.

We derive a RandomDFL mechanism that is inspired by the work done in (Shokri & Shmatikov, 2015) that can be directly applied for orchestrating a naive aggregator free FL approach. RandomDFL is described in detail in Section A.5

7 EXPERIMENTS

In order to evaluate the efficacy of BAFFLE, we focus on four key experiments. We perform a benchmark study where we compare the potential benefits from BAFFLE with respect to classical FL as well as other non FL paradigms. Next, we examine the trends arising from varying number of chunks as well as budget sizes of user devices. We then move onto a scalability analysis that demonstrates the impact of varying the total number of active user devices on the model quality. Lastly, we demonstrate the robustness of BAFFLE to the participation level (PL) parameter of BAFFLE. Further, we also show superior computational performance of BAFFLE compared to the best possible aggregator free approach inspired by the current state-of-the-art.

7.1 Experimental Setup

BAFFLE was implemented and evaluated on a private Ethereum blockchain setup exclusively for our computational experiments. We employed `go-ethereum`, an official `go` based implementation of the Ethereum protocol (GETH) to orchestrate our private blockchain comprising of 16 Ethereum nodes. Proof-Of-Authority was used as the primary consensus protocol for all our experiments. The SC layer was developed using the Solidity programming language and deployed on the private blockchain using `go-ethereum`. The private blockchain was deployed on an Intel Xeon CPU with a clock rate of 2.40 GHz with 16 cores and 2 threads per core. We used `OpenMPI` (Gabriel et al., 2004) in conjunction with `mpi4py` (Dalcin) to spawn multiple distributed memory client processes intended to simulate the user devices on the field. Client processes were deployed on Intel Core i7 CPUs with 12 cores each. We used a 2 layer DNN with 500 perceptron in each layer for our experiments on Keras (Chollet et al., 2015) with TensorFlow (Abadi et al., 2015).

7.2 Benefits Study

Table 1. Benefit Analysis

Category	ASR (USD)	Benefit (%)
No Learning (NL)	13387.31	-
Local Learning (LL)	16106.02	20.31
Classical FL (CFL)	18495.94	38.16
BAFFLE	**18442.21**	**37.75**

In this experiment we compare the benefits accrued by drivers participating in BAFFLE with respect to two other types of learning paradigms. The first comprises of a *Local*

Learning (LL) mechanism, wherein no model aggregation is involved. The second paradigms pertains to an aggregator driven Classical FL(CFL) scheme. We considered each taxi having accumulated approximately 700 rides in each round for a total of 50 rounds. For the FL cases we considered 16 taxis whereas for the LL case, we considered a single taxi. Table 1 presents the results with respect to LL, CFL and BAFFLE mechanisms in terms of their ASR value and benefit relative to the NL baseline.

The trends depicted in Table 1 provide numerous key insights into the performance of BAFFLE. Primarily, we observe that BAFFLE is able to provide a benefit of approximately 38% which rivals the CFL approach. Further, we observe that BAFFLE and CFL approaches improve driver benefit by close to 18% as compared to the LL case. Overall, the results demonstrate that blockchain driven FL paradigms are highly capable of delivering good quality machine learning models in an aggregator free, decentralized fashion.

7.3 Sensitivity Analysis

Table 2. Final Benefit (%) based on Average ASR

Chunk Size (kB)	No. Of Chunks	Budget Size		
		16	24	32
2	738	38.32	38.18	36.51
4	356	36.37	36.87	39.17
8	181	40.23	34.79	38.11
16	88	39.07	38.82	38.02

Table 3. Average Total Training Time(in secs) (Std Dev.)

Chunk Size (kB)	Budget Size		
	16	24	32
2	87.48(2.89)	85.97(3.26)	73.49(2.70)
4	79.92(3.18)	77.63(2.83)	73.22(3.87)
8	74.16(3.70)	71.90(2.53)	69.79(3.09)
16	73.44(4.01)	76.39(3.37)	71.79(3.51)

We perform a robustness study to analyze the impact of variation in chunk sizes as well as local budget sizes on the overall model quality. For this experiment, we considered a total of 64 taxis, with each taxi having accumulated approximately 70 rides in each round for 125 rounds overall. Table 2 shows the benefit percentage calculated for varying chunk and budget sizes. Figure 3 represents the overall trends with Figures 3(a), 3(b) depicting the boxplots pertaining to Gas Costs, Push Time respectively. Table 3 shows the mean and standard deviation with respect to the training time incurred by the individual agents.

The results for all the combinations in Table 2 depict benefits that closely mirror that of the CFL approach shown in Table 2 on the same training set. Therefore, on the basis of data presented in Table 2 one can conclude that BAFFLE is significantly resilient to varying degrees of budget and chunk sizes.

On the basis of Table 3, we conclude that time incurred for training is marginal compared to the push time depicted in Figure 3(b) for all combinations of budget and chunk sizes. The relatively small training time implies that reducing the total push time is critical in ensuring a computationally efficient performance for a blockchain based FL mechanism.

We draw upon the trends shown in Figure 3 to reveal numerous key insights which elucidate the high computational efficiency of BAFFLE.

Primarily, in Figure 3(a) we observe a smaller variation in gas costs for the 2 kB chunk size irrespective of budget sizes. However, as the chunk size increases we see the variation in gas costs also increase substantially for all budget sizes. Second, despite the increased variation, the mean gas cost appears to saturate for higher chunk sizes. We also observe that for the budget size of 32 after the initial uptick there is a relatively more pronounced downward trend for higher chunk sizes. This trend can be clearly attributed to the scoring and bidding mechanism incorporated in BAFFLE. Since a higher chunk size implies lesser number of chunks, there is relatively more competition among user devices to update the same set of chunks. As a result for higher chunk sizes, only user devices which are able to consistently contribute higher scoring chunks will incur a higher gas cost. Therefore, owing to its underlying scoring and bidding mechanism, BAFFLE is able to achieve significant savings in gas costs for the users.

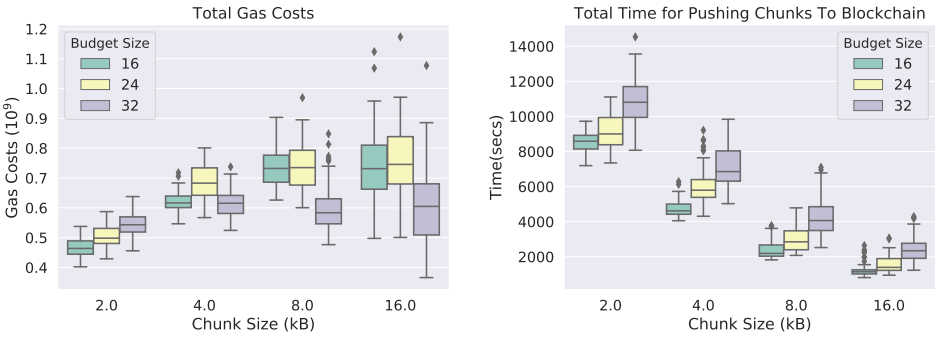
We observe that in Figure 3(b) despite the budget size increasing, the total push time increases only marginally owing to the scoring and bidding mechanisms. Therefore, we can safely say that BAFFLE is successfully able to circumvent the computational bottleneck posed by the push step of BAFFLE .

7.4 Scalability Analysis

Table 4. Scalability analysis with varying No. of Taxis

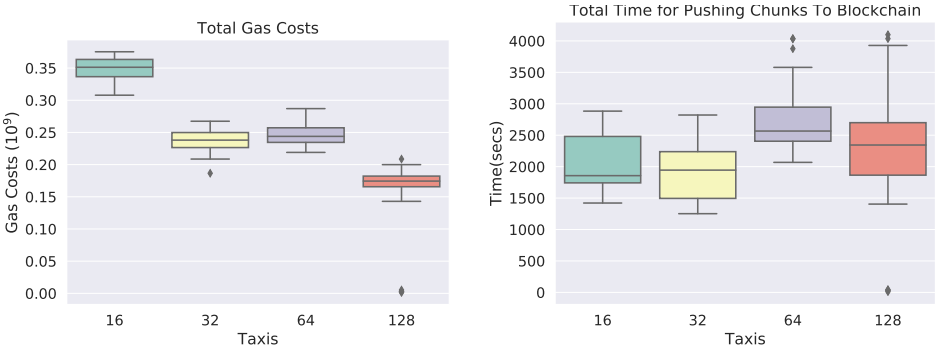
Taxis	Average ASR (USD)	Benefit (%)
16	14489.59	8.2
32	16547.20	23.6
64	18266.72	36.44
128	18414.48	37.55

We attempt to gauge the impact of the total number of active user devices on the performance of BAFFLE. For this experiment, we assumed each taxi having accumulated approximately 70 rides in each round for 62 rounds overall.



(a) Gas Costs (b) Push Time

Figure 3. Performance analysis with respect to Chunk Size and Budget



(a) Gas Costs (b) Push Time

Figure 4. Weak Scaling trends

Table 4 represents the ASR value and the ensuing benefit percentages for 16, 32, 64 and 128 taxis respectively. From the trends presented in Table 4 it is apparent that increasing number of user devices results in a sizeable improvement in the model quality. However, the trends in Table 4 also reveal that the improvement in model quality eventually saturates with increasing active devices potentially indicating a convergence to a globally superior model.

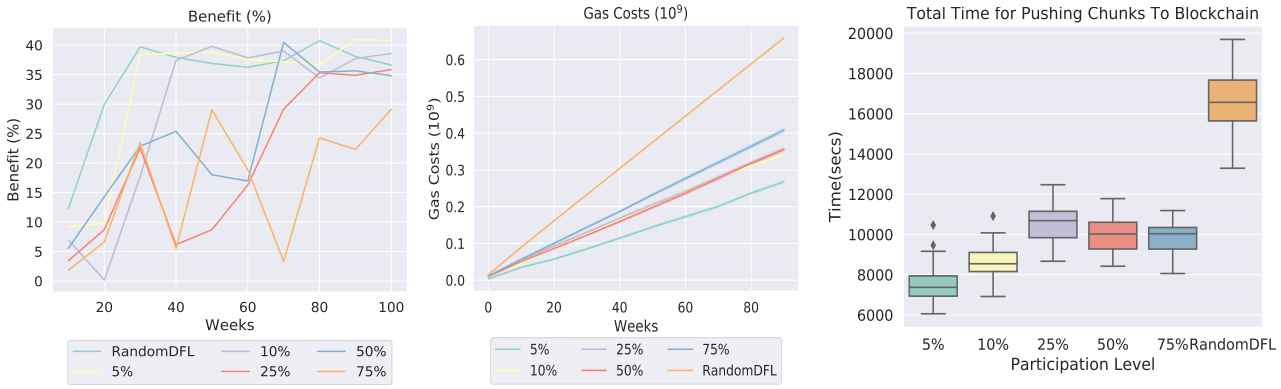
Figure 4 depicts the trends pertaining to the gas costs as well as the push time with varying active user devices in Figures 4(b) and 4(a) respectively. Figure 3(a) shows a reduction in gas costs with increasing number of active devices. However, Figure 4(b) reveals little variation of push time with increase in active devices.

The reduction in gas costs in Figure 4(a) can be attributed to greater competition arising from an increase in total number of devices. Moreover, owing to a constant push time depicted in Figure 4(b) we infer that increase in number of participants leads to reduction in gas costs in BAFFLE.

7.5 Participation Level (PL) Analysis

In this experiment we study the impact of varying the PL on the performance of BAFFLE with 64 taxis, approximately 70 rides per round and a total of 62 rounds. Figure 5 presents results pertaining to PL values ranging from 5% to 75%. Further, we also compare the RandomDFL case in which devices update the global copy without any global coordination. Figures 5(a), 5(b) and 5(c) represent the trends pertaining to the growth in model quality, gas costs and the total push time pertaining to varying PL in every round.

From Figure 5(a), we observe that the fastest convergence of the model quality occurs in case of the RandomDFL case. However, the convergence characteristics of BAFFLE with a 5% PL value closely mirrors the RandomDFL case. Overall, the trends in Figure 5(a) generally indicate that a lower PL value leads to a faster convergence. Figure 5(b) shows that that a lower PL value in BAFFLE incurs a lower gas cost as well. Trends similar to Figure 5(b) are also exhibited in Figure 5(c) wherein a lower PL value in BAFFLE corresponds to a lower total push time as well. We



(a) Model Quality Progression (b) Gas Cost Growth (c) Total Push Time

Figure 5. Performance analysis with respect to Participation Level (PL)

observe that in general, BAFFLE incurs barely half the gas cost and push time as compared to the RandomDFL case. In fact, BAFFLE outperforms the RandomDFL case by a factor of more than 2 with a PL value of 5% both in terms of the gas cost as well as the push time.

Since fewer devices are pushing to the global model copy every round, the chances of multiple devices pulling the same global model is significantly higher in case of a lower PL value. This leads to greater stability in the decentralized process which ultimately leads to a faster convergence for a low PL value as shown in Figure 5(a).

BAFFLE incurs significantly lower gas costs compared to the RandomDFL case owing to minimization of redundant updates. Due to the decentralized round delineation and a robust scoring and bidding process, devices only push chunks that are among the best in the round. As a result, collision among devices for the same chunk is completely eliminated leading to a much lower gas cost and push time.

We now move towards the concluding our work and discussing future directions.

8 Conclusion and Future Work

In this paper we investigate the use of the blockchain for realizing a decentralized aggregator free FL mechanism. We design and develop BAFFLE, a custom made blockchain based framework for aggregator free FL. In our framework, we successfully eliminate the role of a centralized aggregator by effectively decentralizing the concepts of round delineation, user device selection and model aggregation with the help of an SC. Further, in order to circumvent the computational restrictions imposed by the blockchain, we employ an effective model partitioning and serialization mechanism that enables independent and parallel model updates. We orchestrate BAFFLE on a private Ethereum blockchain network with a Solidity driven SC implementation.

We argue that the operational and computational benefits of aggregator free FL has significant potential for solving business problems for micro scale enterprises. We support our claims by applying BAFFLE to a case study pertaining to the ride sharing and taxi industry which serves as a perfect example of a micro scale enterprises. Our case study utilizes the BAFFLE framework to improve driver revenue based on a DRL model that is collectively augmented by all drivers using FL. We show that BAFFLE yields approximately a 40% improvement in driver revenues compared to non FL approaches. We further show that despite being aggregator free, BAFFLE's result quality matches that of classical FL schemes that require investment in an aggregator. Moreover, BAFFLE performs significantly better compared to other aggregator free approaches that are inspired by the current state of the art.

The issue of aggregator free FL opens up new avenues for research especially in the blockchain domain. Effective aggregator free techniques for more complex models like CNNs and LSTM will go a long way to enable wider adoption of FL. Therefore, extending BAFFLE for handling such models forms our immediate future work. We also wish to investigate the use of differential privacy in an aggregator free setting.

Our work shows that an aggregator free approach to FL offers significant potential for revolutionizing small scale organizations and their businesses by delivering quality machine learning models at lower costs. Driven by a robust decentralized platform like the blockchain, the benefits of FL could impact a variety of domains leading to its widespread adoption.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*, 2018.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C. M., Konen, J., Mazzocchi, S., McMahan, B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. Towards federated learning at scale: System design. In *SysML 2019*, 2019. URL https://arxiv.org/abs/1902.01046.

Chen, Z., Wang, W., Yan, X., and Tian, J. Cortex - AI on blockchain: The decentralized AI autonomous system. *Cortex White Paper*, 2018. URL https://www.cortexlabs.ai/Cortex_AI_on_Blockchain_EN.pdf.

Chollet, F. et al. Keras. https://keras.io, 2015.

Christidis, K. and Devetsikiotis, M. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016. ISSN 2169-3536. doi: 10.1109/ACCESS.2016.2566339.

Dalcin, L. Mpi for python. URL https://mpi4py.readthedocs.io/en/stable/.

Douriez, M., Doraiswamy, H., Freire, J., and Silva, C. T. Anonymizing nyc taxi data: Does it matter? In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 140–148, Oct 2016. doi: 10.1109/DSAA.2016.21.

Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., et al. Open mpi: Goals, concept, and design of a next generation mpi implementation. In *European Parallel Virtual Machine/Message Passing Interface Users Group Meeting*, pp. 97–104. Springer, 2004.

GETH. Go ethereum: Official go implementation of the ethereum protocol. URL https://geth.ethereum.org/.

Han, M., Senellart, P., Bressan, S., and Wu, H. Routing an autonomous taxi with reinforcement learning. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 2421–2424. ACM, 2016.

Harris, J. D. and Waggoner, B. Decentralized & collaborative AI on blockchain. *arXiv preprint arXiv:1907.07247*, 2019.

Khajeh-Hosseini, A., Sommerville, I., and Sriram, I. Research challenges for enterprise cloud computing. *arXiv preprint arXiv:1001.3257*, 2010.

Kim, H., Park, J., Bennis, M., and Kim, S.-L. On-device federated learning via blockchain and its latency analysis. *arXiv preprint arXiv:1808.03949*, 2018.

Konečnỳ, J., McMahan, B., and Ramage, D. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

Konecny, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL https://arxiv.org/abs/1610.05492.

Lalitha, A., Shekhar, S., Javidi, T., and Koushanfar, F. Fully decentralized federated learning. *Proceedings of third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

Lalitha, A., Kilinc, O. C., Javidi, T., and Koushanfar, F. Peer-to-peer federated learning on graphs. *arXiv preprint arXiv:1901.11173*, 2019.

McMahan, B. and Ramage, D. Federated learning: Collaborative machine learning without centralized training data, 2017. URL https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

Nakamoto, S. et al. Bitcoin: A peer-to-peer electronic cash system. 2008. URL https://bitcoin.org/bitcoin.pdf.

Riedmiller, M. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pp. 317–328. Springer, 2005.

Shi, D., Ding, J., Errapotu, S. M., Yue, H., Xu, W., Zhou, X., and Pan, M. Deep q-network based route scheduling for transportation network company vehicles. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Dec 2018. doi: 10.1109/GLOCOM.2018.8647546.

Shi, D., Ding, J., Errapotu, S. M., Yue, H., Xu, W., Zhou, X., and Pan, M. Deep q-network based route scheduling for tnc vehicles with passengers location differential privacy. *IEEE Internet of Things Journal*, pp. 1–1, 2019. ISSN 2327-4662. doi: 10.1109/JIOT.2019.2902815.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321. ACM, 2015.

Travel and of New York, L. C. Nyc 2018 yellow taxi data. URL https://data.cityofnewyork.us/Transportation/2018-Yellow-Taxi-Trip-Data/t29m-gskq.

Verma, T., Varakantham, P., Kraus, S., and Lau, H. C. Augmenting decisions of taxi drivers through reinforcement learning for improving revenues, 2017. URL https://aaai.org/ocs/index.php/ICAPS/ICAPS17/paper/view/15746.

Wood, G. et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

Yong, H., Lee, C., and Wang, D. Artificial intelligence computing platform driven by blockchain. Technical report, DeepBrain Chain, Singapore, 2017. URL https://www.deepbrainchain.org/assets/pdf/DeepBrainChainWhitepaper_en.pdf.

A Appendix

A.1 Proof of Lemma 1

Proof. We know that for SGD the following relation holds:

$$\hat{w}_i^{k+1} = \hat{w}_i^0 - \eta \sum_{t=1}^{k} \nabla f(\hat{w}^t)_i \tag{2}$$

where $\hat{w}_i^k$ is the estimate of the i^{th} component of the weight vector at round k, η is the learning rate. Further, $\nabla f(\hat{w}^k)_i$ is i^{th} component of the gradient estimated based on the globally available weight vector.

In case of BAFFLE, we can say that

$$\hat{w}_i^{k+1} = \frac{1}{2}[\hat{w}_i^k + \hat{w}_i^k - \eta_{BFL} \nabla f(\hat{w}^t)_i] \tag{3a}$$

$$\hat{w}_i^{k+1} = \hat{w}_i^k - \frac{\eta_{BFL}}{2} \nabla f(\hat{w}^t)_i \tag{3b}$$

Therefore, if at the t^{th} round, device j_t is active and the i^{th} component is chosen, it follows that the expected value of the weight vector is given by:

$$E[\hat{w}_i^{k+1}] = \hat{w}_i^0 - \frac{\eta_{BFL}}{2} E\left[\sum_{t=1}^{k} \nabla f_{j_t}(\hat{w}^t)_i\right] \tag{4}$$

At every round, we also assume that the probability of user device j_t being selected is denoted by α_{BFL}. Given a budget size B, the total number of chunks C, devices choosing their chunks randomly subject to budget B, the probability of picking the chunk containing the i^{th} weight element, is then determined as follows:

$$\binom{C-1}{B-1} / \binom{C}{B} = \frac{B}{(C-B+1)C} \tag{5}$$

Therefore, Equation 4 is equivalent to:

$$E[\hat{w}_i^{k+1}] = \hat{w}_i^0 - \frac{\eta_{BFL}}{2}\left[\sum_{t=1}^{k} \alpha_{BFL} \sum_{j=1}^{n} \frac{B \nabla f_{j_t}(\hat{w}^t)_i}{(C-B+1)C}\right] \tag{6}$$

which leads to:

$$E[\hat{w}_i^{k+1}] = \hat{w}_i^0 - \frac{B.\eta_{BFL}\alpha_{BFL}}{2(C-B+1)C} E\left[\sum_{t=1}^{k}\sum_{j=1}^{n} \nabla f_{j_t}(\hat{w}^t)_i\right] \tag{7}$$

On the other hand, with aggregator driven FL, with L user devices aggregated in each round, we can similarly state:

$$E[\hat{w}_i^{k+1}] = \hat{w}_i^0 - \frac{\eta_{FL}.\alpha_{FL}}{L}\left[\sum_{t=1}^{k}\sum_{j=1}^{n} \nabla f_{j_t}(\hat{w}^t)_i\right] \tag{8}$$

where α_{FL}, η_{FL} is the probability of choosing a device and the learning rate of aggregator driven FL respectively.

Therefore, equating 7 and 8, we can say that with a learning rate of

$$\eta_{BFL} = \frac{2.C.(C-B+1)\alpha_{FL}}{B.L.\alpha_{BFL}}.\eta_{FL} \tag{9}$$

BAFFLE is equivalent to classical FL with learning rate η_{FL}. □

A.2 Consensus Protocols

As an extension of Section 3.2, we continue the description of popular consensus protocols powering the blockchain.

Proof of Authority (PoA): PoA uses a set of *authorities*, nodes that are explicitly allowed to create new blocks and secure the blockchain. A validator's identity performs a key role in this consensus mechanism. The block needs to be signed off by the majority of authorities, which makes the block become a part of the permanent record of the distributed ledgers. PoA is better suited for consortium settings as it is more secure, less computationally intensive, more performing, and more predictable.

Proof of Work (PoW): This consensus algorithm is used to select a miner for the next block generation. The idea behind

Table 5. SC Attributes in BAFFLE

Attribute	*Description*
Model ID	Unique identifier assigned for every FL task by the SC
Round Registration Details	List of users with submitted bids for the upcoming round
Participation Level	The minimum number of users with submitted bids required to begin a round
Chunk Core Array	A data structure for every chunk holding: last updated time; last user to update; set of submitted scores & their owners

this PoW algorithm is to solve a complex mathematical problem that requires a lot of computational power and thus, the first node who solves the problem is eligible for mining the next block with a reward. Bitcoin adopts this PoW consensus mechanism.

Proof of Stake (PoS): In this type of consensus algorithm, the nodes called validators look into blocks to be added to the chain and invest in the coins of the system as stake instead of solving a complex math problem. The validators bet on the blocks that would most likely be added to the chain. Based on the actual blocks added to the Blockchain, all the validators get a reward in proportion to their bets and their stake increases accordingly. In the end, a validator is selected to create a new block based on their economic stake in the network. This is the most common alternative to PoW and Ethereum has shifted from PoW to PoS.

A.3 Smart FL Contract Data Structure

The Smart FL Contract follows contract-oriented design principles that required to function on the blockchain network. Fields of significance contained in the Smart FL Contract are listed in Table A.1.

A.4 Centralized Deep Batch Q Learning

Algorithm 3 Centralized Deep Neural Fitted Q

for $k = 0 \dots$ **do**
  observe new ride set $\mathcal{H}^k$
  pull latest available model Q from blockchain
  perform averaging $Q_k \leftarrow \frac{Q_k^j + Q}{2}$
  update $\tilde{Q}_k^j$ based on Equation 1a
  locally train Q_{k+1}^j via Equation 1b
end for

Algorithm 3 details the centralized batch Deep Q Learning for improving driver revenue. It starts with observation of a new ride set every epoch. For every ride in the ride set, the existing Q-value estimate is updated with the fare collected for the ride and a discounted future reward. The discounted future reward is based on the action that gives highest Q value originating from the destination state. Based on the observed set of rides, a Deep Neural Network (DNN) is used to calculate the next Q function estimate.

A.5 Random Decentralized FL (RandomDFL)

Algorithm 4 Randomized Decentralized Deep Neural Fitted Q

for taxi: $j = 1 \dots P$ **do**
  for $k = 0 \dots$ **do**
    observe new ride set $\mathcal{H}^k$
    pull latest available model Q from blockchain
    perform averaging $Q_k \leftarrow \frac{Q_k^j + Q}{2}$
    update $\tilde{Q}_k^j$ based on Equation 1a
    locally train Q_{k+1}^j via Equation 1b
    push random set of chunks $C^k \subseteq \mathcal{C}, |C^k| = B$
  end for
end for

In the randomized version represented in Algorithm 4, the SC is considered to be naive. User devices are free to update any chunks subject to their own budget values. In this naive randomized version, some chunk updates are bound to get wasted owing to the fact that they may be overwritten by another user device's contribution before the previous update has had a chance to be read by the other agents.