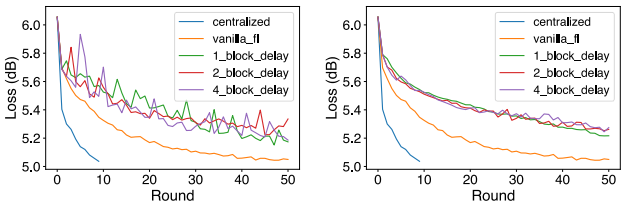
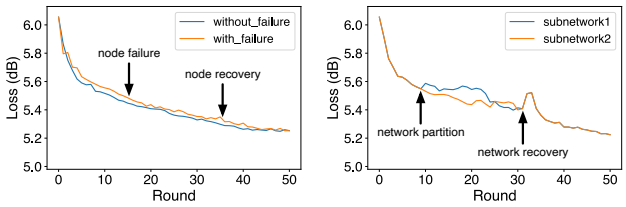


(a) Without EWMA. (b) With EWMA.

Fig. 5. Impact of model aggregation with EWMA on model convergence and accuracy.



(a) Miner node failure. (b) Miner network partition failure.

Fig. 6. Impact of failures on model training.

A. Microbenchmarks

In order to verify the correctness and robustness of Fiesta, we use the following simulation setting. We have 5 miners and 10 mobile devices in the system, which is the same as the real-world deployment. The learning mode of mobile devices is simulated through Python code. One round of local training in a mobile device contains 10 epochs. All miners and mobile devices processes run in parallel in a single 28-core machine so that each process occupies roughly the same hardware resource, i.e. one core. We use **App1** for benchmarking. For simplicity, we only use the readings of channel 19 (500-506 MHz) in [31]. The data are randomly sampled with 10% probability as the validation set first. For the rest 90%, they are sorted by timestamp and then allocated to the mobile devices as 10 consecutive time periods of a same number of data records. Although the number of data records on each mobile device is the same, there is a certain degree of heterogeneity in terms of spatial distribution.

In the following subsections, "baseline" refers to the naive synchronous FedAvg. The label "centralized" refers to local training with the full dataset in a contrast to the federated learning fashion.

1) Mining difficulty: We first decide the mining difficulty to balance the first confirmation time and the forking probability of blockchain. The difficulty d means, in order to generate a valid block, the number of leading zeros of the block hash must equal d. We calculate forking probability as the total number of block replacements occurring due to knowing a longer chain from other miners divided by the total number of blocks in the blockchain. Fig. **??** shows the results.

Considering the results of first confirmation time and forking probability, we fix the difficulty as 3 for the rest of our simulations and the real-world deployment. Note that forking probability is also related to the scale of the miner network. The larger scale of the miner network, the higher the forking probability it can get.

2) Effectiveness of EWMA: We study the effect of model aggregation with EWMA introduced in § II-B on model convergence and accuracy. Detailedly, we let the blockchain contain 5 model updates in each block precisely and divide the 10 mobile devices into two groups. There are 5 faster mobile devices that upload their updates immediately after their local training round is finished, i.e. no block delay. The other 5 mobile devices are the slower ones, whose updates are at least $b(b >= 1)$ blocks delayed (specified in the legend of Fig. 5. We use the loss on the validation set to check the convergence and accuracy.

Fig. 5 illustrates the results. If EWMA is not used when doing model aggregation, we can see from Fig. 5(a) that the loss would fluctuate significantly. Fig. 5(b) shows that if EWMA is used when doing model aggregation, the learning curves are very smooth without sacrificing accuracy, compared with not using EWMA. Having large fluctuations in the learning curve is even more undesirable if we consider the possibility of mobile devices joining/leaving, the heterogeneous data distribution on each device, and new data collected. Additionally, Fig. 5 also contains the results of centralized learning (marked as "centralized") and vanilla federated learning (marked as "vanilla_fl") in the synchronous setting. Vanilla federated learning converges slightly faster than the results of the asynchronous settings and has a marginally better estimation accuracy.

Note that α in EWMA is set as 0.5 for all the simulations and the real-world deployment to balance past and present, which can be further fine tuned if needed.

3) Robustness: We study the robustness introduced by blockchain, which does not exist in centralized learning and ordinary federated learning. We use an asynchronous setting where each block contains 4˜10 model updates with EWMA enabled.

We first study system's resistance toward miner node failure. Fig. 6(a) illustrates the validation loss as a function of the number of rounds with miner node failure and recovery. A randomly chosen miner node fails after round 15 is finished and is back online after round 35 is finished. In Fig. 6(a), the curve standing for the node failure experiment converges to a similar accuracy at approximately same speed compared to the non-failure one.

By using a distributed miner network, another type of network failure is possible, which is network partitions. The miner network is partitioned into two subnetworks after the birth of block 10, with one subnetwork containing 2 miners and the other containing 3 miners. After block 31, the two subnetworks merge back into one. Further, the validation loss still gradually goes down, almost the same as the no failure case in Fig. 6(a). We admit that limited failure rehearsals don't speak loudly enough for a perfect robustness, but it does demonstrate Fiesta's enhancement in its resistance towards failures in a qualitative manner.