APPENDIX

A. Generalization

It is possible to apply the softmax function to the approximated accuracy instead of the population loss. If we let $\hat{A}_j(w_i^r)$ denote the accuracy of model weights w_i^r using validator j's local data, and the miner computes $\hat{A}(w_i^r) = \frac{1}{V}\sum_{j=1}^{V}\hat{A}_j(w_i^r)$, then the softmax accuracy aggregated next global weight is $\tilde{\mathbf{w}}^{r+1} = \sum_{i=1}^{K}\hat{p}_i^{r,A}\mathbf{w}_i^r$, where $\hat{p}_i^{r,A} = \frac{exp(-\hat{A}(\mathbf{w}_i^r))}{\sum_{s=1}^{K}exp(-\hat{A}(\mathbf{w}_s^r))}$.

The softmax aggregation technique can also be generalized to the traditional FL setting. It is trivial if there exists a large globally available test set. Otherwise, the central server can designate a few clients as validators based on accumulated reputation or some other criteria and send them all workers' model updates. Then the validators can send back the approximated population loss values and the central server can utilize the loss values to complete the softmax aggregation process. However, it is worth noting that without the existing blockchain network, this procedure is at risk of having a larger number of malicious validators when compared to SABFL.

B. Implementation Details

For the experiments we do not use a real blockchain such as Ethereum but instead we implement Algorithm **??** because the goal is to study the performance and robustness of the softmax aggregation method.

To allow participants to have data sets with varying sizes and distributions, we use the data partition scheme in RADFed [1]. Specifically, hyper-parameter λ from RADFed is set to 1 version one of the experiments and 0.1 for version two, and in this case version two has larger data heterogeneity. The local training epoch E is set to 4 in all experiments except version three of experiment 3 where it is set to 8. The learning rate is fixed to 0.01 in all experiments.

In the implementation, to ensure a fair stopping criteria for all the aggregation methods tested, we devise a novel stopping rule. For every single run let us denote the test set accuracy after training for i global rounds as a_i. Then for $i \geq 30$ we consider the ratio of the minimum over the maximum accuracy over the last 30 rounds. Specifically, we compute the sequence $k_i = \frac{\min S_i}{\max S_i}$ where S_i denotes the set $\{a_j\}_{j=i-29}^{i}$. We stop when k_i decreases for the first time, say at round T, and treat $\max a_i$ for $i \leq T$ as the final accuracy for the specific run. Note that k_i decreases if the accuracy of the new round is the new minimum of the past 30 rounds or if the last 30 rounds' accuracy values are not the maximum. In either case, it is reasonable to assume we have already achieved the maximum accuracy.

For training of MNIST, we use a CNN with 32 and 64 filters with size 5×5 and 512 hidden units for the fully connected layer. For FashionMNIST, we use a similar structure but with two hidden layers of size 600 and 120. The feed forward neural network for the Forest Cover Type and eICU data set has two hidden layers of size 1,000 and 500. Finally, the recurrent neural network for IMDB sentiment classification consists of a 64 dimensional embedding layer, 2 layers of LSTM, and a hidden layer of size 256. All of the architectures selected are able to reach a final accuracy similar to the state-of-the-art in the traditional machine learning setting. Finally, the framework has been implemented in PyTorch using Google Colab with Nvidia Tesla T4 GPUs and Intel(R) Xeon(R) CPU @ 2.00GHz.

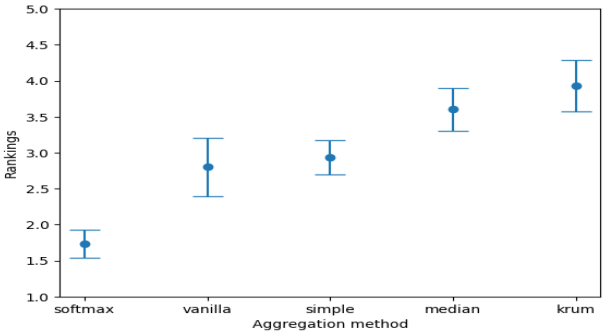
TABLE I: Final Accuracy (%)

Experiment	Softmax	Vanilla	Simple	Median	Krum
Exp1-LH	97.43 (0.21)	**98.56** (0.11)	97.36 (1.00)	96.42 (1.29)	97.17 (0.07)
Exp1-HH	96.37 (0.33)	**98.40** (0.11)	95.30 (1.82)	88.29 (0.52)	89.19 (0.33)
Exp2-LH	96.95 (0.40)	96.91 (1.31)	**97.32** (0.25)	96.57 (1.53)	97.03 (0.09)
Exp2-HH	95.16 (0.86)	**95.63** (2.12)	94.75 (1.65)	85.93 (1.45)	88.72 (0.60)
Exp3-LH	**97.24** (0.56)	86.10 (8.53)	92.93 (1.49)	95.61 (1.73)	97.18 (0.10)
Exp3-HH	**95.18** (0.55)	84.69 (7.80)	86.45 (6.47)	85.11 (4.57)	57.11 (38.39)
Exp3-HHE	**98.01** (0.33)	86.16 (14.17)	93.81 (2.51)	97.70 (0.20)	80.17 (34.36)
Exp4-LH	92.95 (3.01)	**96.65** (1.21)	93.53 (2.58)	90.77 (2.43)	78.02 (33.33)
Exp4-HH	83.21 (3.46)	**92.73** (2.58)	83.10 (2.34)	67.51 (3.54)	64.74 (26.44)
Exp5-LH	89.87 (0.61)	82.29 (12.14)	92.02 (1.80)	88.90 (2.88)	**94.68** (0.33)
Exp5-HH	81.77 (4.41)	**83.61** (10.35)	76.25 (6.40)	65.83 (5.07)	64.91 (25.97)
Exp6-LH	**98.04** (0.58)	83.25 (24.98)	93.53 (2.16)	96.18 (1.68)	39.72 (47.06)
Exp6-HH	**97.31** (0.69)	82.25 (12.00)	90.22 (1.42)	96.83 (0.41)	59.08 (45.53)
Exp7-LH	**89.05** (0.41)	83.78 (1.66)	82.42 (3.43)	88.57 (0.63)	52.13 (41.68)
Exp8-LH	**72.78** (0.17)	71.85 (0.31)	71.18 (0.20)	71.70 (0.22)	72.54 (0.36)
Exp9-LH	**81.92** (0.45)	80.85 (0.64)	80.45 (1.03)	79.77 (1.07)	56.83 (20.15)
Exp10	90.74 (0.11)	**90.80** (0.18)	90.50 (0.15)	90.71 (0.17)	61.93 (23.98)

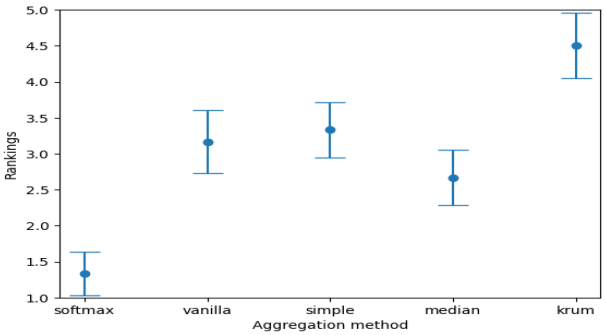
C. Complete Results

Table I contains the accuracy results with standard deviations provided in parentheses from all experiments and aggregation methods. The results are consistent with other results, specifically, the softmax aggregation method is the most robust in varying situations. It either is the best performer or has similar accuracy to the best aggregation method. It is also worth noting again that the vanilla aggregation method, which performs well in a few tasks, utilizes private sample size information that is beneficial to the performance, and ignoring the vanilla method, softmax aggregation is the best performer in 13 out of the 17 tasks.

We find softmax aggregation has consistent relative performance across the various settings in our experiments. We rank the performance of each method from 1 to 5 across the 16 different experimental settings, then we compute the average ranking statistics and present them in Figure 1. Figure 1a contains the average rankings for all experiments, and



(a) All experiments



(b) With 40% malicious

Fig. 1: Rankings statistics.

Figure 1b only contains experiments where there are 40% malicious participants, specifically experiments 3,5,7,8 and 9. We find that softmax aggregation not only has the best average ranking, it also has the smallest ranking deviation in both studies, indicating it has consistent superior performance in varying settings. We find that even when there are 40% malicious clients, median aggregation performs relatively well in these previously untested situations. Krum suffers the most from robustness issues. Here although vanilla method has the highest robustness, its performance suffers in presence of malicious participants, and there are quite a few experiments with many malicious clients.

The outcomes of our real-world simulation experiment utilizing the eICU dataset shows that softmax aggregation ranks narrowly behind vanilla aggregation and has the smallest standard deviation. It is crucial to reiterate that this dataset features clients with a wide range of sample sizes, from 200 to over 7,000 patients, giving vanilla aggregation an undue advantage by utilizing this sensitive information regarding patient numbers.

We also conducted a study of the accuracy based softmax aggregation introduced in the Appendix with the same setting as in Exp3-HH. We find accuracy based softmax aggregation achieves 94.52% final accuracy, which is superior to the benchmark methods. We want to point out that softmax aggregation with accuracy loses the convergence property introduced previously.

D. Proof of Theorem **??**

To prove Theorem **??**, we need to first introduce the following results. For non-negative real numbers $x_1, x_2, x_3, \ldots, x_n$, it holds that

$$\frac{1}{\sum_{j=1}^{n} exp(x_j)} \sum_{i=1}^{n} exp(x_i) x_i \geq \frac{\sum_{i=1}^{n} x_i}{n}.$$

Additionally, the equality only holds when $x_1 = x_2 = x_3 = \cdots = x_n$. *Proof.* We prove by induction. For the base case of $n = 1$ the statement simplifies to $\frac{exp(x_1)}{exp(x_1)} x_1 \geq x_1$, and it is trivially true. For the induction step, without loss of generality, let us assume $x_{k+1} \geq x_k \geq x_{k-1} \geq \cdots \geq x_1$. To simplify the notation, let $S_m = \sum_{i=1}^{m} exp(x_i)$. We have the following

$$\begin{aligned}
\sum_{i=1}^{k+1} \frac{exp(x_i)}{S_{k+1}} x_i &= \frac{exp(x_{k+1})}{S_{k+1}} x_{k+1} + \sum_{i=1}^{k} \frac{exp(x_i)}{S_{k+1}} x_i \\
&= \frac{exp(x_{k+1})}{S_{k+1}} x_{k+1} + \frac{S_k}{S_{k+1}} \sum_{i=1}^{k} \frac{exp(x_i)}{S_k} x_i \\
&\geq \frac{exp(x_{k+1})}{S_{k+1}} x_{k+1} + \frac{S_k}{S_{k+1}} \frac{\sum_{i=1}^{k} x_i}{k} \\
&\geq \frac{1}{k+1} x_{k+1} + \frac{k}{k+1} \frac{\sum_{i=1}^{k} x_i}{k} \\
&= \frac{\sum_{i=1}^{k+1} x_i}{k+1}.
\end{aligned}$$

The first inequality is by the induction hypothesis and the equality holds only when $x_1 = x_2 = x_3 = \cdots = x_k$. For the second inequality, note that we assume $x_{k+1} \geq x_k \geq x_{k-1} \geq \cdots \geq x_1$, therefore, we have $\frac{exp(x_{k+1})}{S_{k+1}} \geq \frac{1}{k+1}$, $x_{k+1} \geq \frac{\sum_{i=1}^{k} x_i}{k}$, and the equality is true only when $x_{k+1} = x_i$, for every i. It also uses the fact that $f(s) = s x_{k+1} + (1-s) \frac{\sum_{i=1}^{k} x_i}{k}$ is an increasing function. ■

This proposition shows the softmax weighted average is no less than arithmetic average, an immediate corollary follows. For non-negative real numbers $x_1, x_2, x_3, \ldots, x_n$, we have

$$\frac{1}{\sum_{j=1}^{n} exp(-x_j)} \sum_{i=1}^{n} exp(-x_i) x_i \leq \frac{\sum_{i=1}^{n} x_i}{n}.$$

Additionally, the equality only holds when $x_1 = x_2 = x_3 = \cdots = x_n$. The proof follows by applying Lemma with $-x_1, -x_2, -x_3, \ldots, -x_n$.

We have to introduce another lemma to prove our main result. It is a special case of a result from the proof of Theorem 1 in [2] when there is only one participant. *(Result from the proof of Theorem 1 in [2])* Suppose Assumptions **??-??** hold. For fixed $\mathbf{w}^0$, let $\tilde{\mathbf{w}} = \mathbf{w}^0 - \alpha \sum_{t=1}^{E} \sum_{b=1}^{B} \nabla f(\mathbf{w}^{t-1}; \xi_i^{r,t})$, and let α be such that

$$\frac{L^2 \alpha^2 (E+1)(E-2)}{2} + L\alpha E \leq 1, and\ 1 - \delta \geq L^2 \alpha^2$$

for some constant $\delta \in (0,1)$. If $B = |\xi_i^{r,t}|$, then we have

$$\begin{aligned}
&\mathbb{E}F(\tilde{\mathbf{w}}) - F(\mathbf{w}^0) \\
\leq& -\frac{(E-1+\delta)\alpha}{2}||\nabla F(\mathbf{w}^0)||^2 \\
&+ \frac{L\alpha^2 EM}{2B}(E + \frac{L(2E-1)(E-1)\alpha}{6}).
\end{aligned}$$

With the previous results, we can now start the proof of the main result.

Proof of Theorem **??**. Consider the random variables of two consecutive rounds' global weights, $\tilde{\mathbf{w}}^r$ and $\tilde{\mathbf{w}}^{r-1}$, and let us define $p_i^r = \frac{exp(-F(\mathbf{w}_i^{r,E}))}{\sum_{s=1}^K exp(-F(\mathbf{w}_s^{r,E}))}$. Then we can bound the global objective by

$$\begin{aligned}
&F(\tilde{\mathbf{w}}^r) - F(\tilde{\mathbf{w}}^{r-1}) \\
=& F(\sum_{i=1}^K \hat{p}_i^r \mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}^{r-1}) \\
\leq& \sum_{i=1}^K \hat{p}_i^r F(\mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}^{r-1}) \\
=& \sum_{i=1}^K (p_i^r + (\hat{p}_i^r - p_i^r))F(\mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}^{r-1}) \\
=& \sum_{i=1}^K p_i^r F(\mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}^{r-1}) \\
&+ \sum_{i=1}^K (\hat{p}_i^r - p_i^r)F(\mathbf{w}_i^{r,E}).
\end{aligned} \tag{1}$$

The first equality is by definition and the first inequality is by the convexity assumption. Since the softmax function is 1-Lipschitz according to Proposition 4 in [3] and if we let $m_r = \min_j m_j^r$, then Algorithm **??** implies $|F(\mathbf{w}_i^{r,E})| \leq m_r$ and $|\hat{F}(\mathbf{w}_i^{r,E}) - F(\mathbf{w}_i^{r,E})| < \frac{\epsilon}{2^r K^{5/2} m_r}$, and then we have

$$\begin{aligned}
&\sum_{i=1}^K (\hat{p}_i^r - p_i^r)F(\mathbf{w}_i^{r,E}) \\
\leq& \sum_{i=1}^K |\hat{p}_i^r - p_i^r||F(\mathbf{w}_i^{r,E})| \\
\leq& \sum_{i=1}^K K^{1/2}||\hat{\mathbf{p}}^r - \mathbf{p}^r||_2|F(\mathbf{w}_i^{r,E})| \\
\leq& \sum_{i=1}^K K^{1/2}||\hat{\mathbf{F}}(\mathbf{w}^{r,E}) - \mathbf{F}(\mathbf{w}^{r,E})||_2|F(\mathbf{w}_i^{r,E})| \\
\leq& \sum_{i=1}^K K^{1/2}||\hat{\mathbf{F}}(\mathbf{w}^{r,E}) - \mathbf{F}(\mathbf{w}^{r,E})||_1|F(\mathbf{w}_i^{r,E})| \\
\leq& \sum_{i=1}^K K^{1/2} K \frac{\epsilon}{2^r K^{5/2} m_r} m_r \\
=& \frac{\epsilon}{2^r}.
\end{aligned} \tag{2}$$

Here the first two inequalities are by the properties of the $L1$ and $L2$ norms, the third inequality is by softmax function being 1-Lipschitz, the fourth inequality is by $L1$ norm being no less than $L2$ norm, and finally the last inequality is by Algorithm 2 lines 12-16. The bolded letters $\hat{\mathbf{p}}^r, \hat{\mathbf{p}}^r, \hat{\mathbf{F}}(\mathbf{w}^{r,E}), \mathbf{F}(\mathbf{w}^{r,E})$ denote the K dimensional vectors. Next we apply Corollary to $F(\mathbf{w}_1^{r,E}), F(\mathbf{w}_2^{r,E}), F(\mathbf{w}_3^{r,E}), \ldots, F(\mathbf{w}_K^{r,E})$, plug inequality (2) into inequality (1), and obtain

$$\begin{aligned}
F(\tilde{\mathbf{w}}^r) - F(\tilde{\mathbf{w}}^{r-1}) &\leq \frac{1}{K}\sum_{i=1}^K F(\mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}^{r-1}) + \frac{\epsilon}{2^r} \\
&= \frac{1}{K}\sum_{i=1}^K (F(\mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}_{r^1})) + \frac{\epsilon}{2^r}.
\end{aligned}$$

We are interested in expected objective decrease, where the expectation is with respect to the set of random variables $\Xi_r = \{\xi_i^{r,t}\}_{(i,t)}$. We use the law of total expectation with respect to r. Participant j's update is computed as

$$\mathbf{w}_j^{r,E} = \tilde{\mathbf{w}}^{r-1} - \alpha_r \sum_{t=1}^E \nabla f(\mathbf{w}_j^{r,t-1}; \xi_j^{r,t}).$$

We have

$$\begin{aligned}
&\mathbb{E}_{\Xi_r}[F(\tilde{\mathbf{w}}^r) - F(\tilde{\mathbf{w}}^{r-1})|\Xi_{r-1}] \\
\leq& \frac{1}{K}\sum_{i=1}^K \mathbb{E}_{\Xi_r}[F(\mathbf{w}_i^{r,E}) - F(\tilde{\mathbf{w}}^{r-1})|\Xi_{r-1}] + \frac{\epsilon}{2^r}.
\end{aligned} \tag{3}$$

Now we apply Lemma and the expectation of all random variables on both sides to obtain

$$\begin{aligned}
&\mathbb{E}_{\Xi_r}[F(\tilde{\mathbf{w}}^r) - F(\tilde{\mathbf{w}}^{r-1})] \\
=& \mathbb{E}_{\Xi_{r-1}}\mathbb{E}_{\Xi_r}[F(\tilde{\mathbf{w}}^r) - F(\tilde{\mathbf{w}}^{r-1})|\Xi_{r-1}] \\
\leq& -\frac{(E-1+\delta)\alpha_r}{2}\mathbb{E}_{\Xi_{r-1}}||\nabla F(\tilde{\mathbf{w}}^{r-1})||^2 \\
&+ \frac{L\alpha_r^2 EM}{2B_r}(E + \frac{L(2E-1)(E-1)\alpha_r}{6}) + \frac{\epsilon}{2^r}.
\end{aligned}$$

Combining with the fact that $F^* - F(\tilde{\mathbf{w}}_0) \leq F(\tilde{\mathbf{w}}_r) - F(\tilde{\mathbf{w}}_0)$, we obtain the following bound

$$\begin{aligned}
&\mathbb{E}\sum_{r=1}^R \alpha_r ||\nabla F(\tilde{\mathbf{w}}^{r-1})||^2 \\
\leq& \frac{2(F(\tilde{\mathbf{w}}_0) - F^*)}{(E-1+\delta)} + \frac{2\epsilon}{(E-1+\delta)} \\
&+ \sum_{r=1}^R \frac{LE\alpha_r^2 M[6E + L(2E-1)(E-1)\alpha_r]}{6B_r(E-1+\delta)},
\end{aligned}$$

which completes the proof. ■

REFERENCES

[1] Y. Xue, D. Klabjan, and Y. Luo, "Aggregation delayed federated learning," in *2022 IEEE International Conference on Big Data*, 2022, pp. 85–94.

[2] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 3219–3227.

[3] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv*, vol. 1704.00805, 2017.