Proof of Swarm Based Ensemble Learning for Federated Learning Applications*

Ali Raza†
ar718@kent.ac.uk
Univ. Lille, ENSAIT, ULR 2461 – GEMTEX – Génie et Matériaux Textiles, France
Institute of Cyber Security for Society (iCSS) & School of Computing, University of Kent, UK

Kim Phuc Tran
Ludovic Koehl
Kim-phuc.tran@ensait.fr
ludovic.koehl@ensait.fr
Univ. Lille, ENSAIT, ULR 2461 – GEMTEX – Génie et Matériaux Textiles, France

Shujun Li†
S.J.Li@kent.ac.uk
Institute of Cyber Security for Society (iCSS) & School of Computing, University of Kent, UK

ABSTRACT

Ensemble learning combines results from multiple machine learning models in order to provide a better and optimised predictive model with reduced bias, variance and improved predictions. However, in federated learning it is not feasible to apply centralised ensemble learning directly due to privacy concerns. Hence, a mechanism is required to combine results of local models to produce a global model. Most distributed consensus algorithms, such as Byzantine fault tolerance (BFT), do not normally perform well in such applications. This is because, in such methods predictions of some of the peers are disregarded, so a majority of peers can win without even considering other peers' decisions. Additionally, the confidence score of the result of each peer is not normally taken into account, although it is an important feature to consider for ensemble learning. Moreover, the problem of a tie event is often left un-addressed by methods such as BFT. To fill these research gaps, we propose PoSw (Proof of Swarm), a novel distributed consensus algorithm for ensemble learning in a federated setting, which was inspired by particle swarm based algorithms for solving optimisation problems. The proposed algorithm is theoretically proved to always converge in a relatively small number of steps and has mechanisms to resolve tie events while trying to achieve sub-optimum solutions. We experimentally validated the performance of the proposed algorithm using ECG classification as an example application in healthcare, showing that the ensemble learning model outperformed all local models and even the FL-based global model.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**;

KEYWORDS

Privacy, federated learning, ensemble, consensus protocol, evolutionary computing, swarm algorithms, healthcare.

ACM Reference Format:
Ali Raza, Kim Phuc Tran, Ludovic Koehl, and Shujun Li. 2023. Proof of Swarm Based Ensemble Learning for Federated Learning Applications. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27–31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3555776.3578601

1 INTRODUCTION

Machine learning (ML) can improve digital healthcare by providing efficient and accurate solution to different problems [6]. Federate learning (FL) has been applied in many healthcare application to solve the issues of centralised machine learning, where a joint machine learning model is trained by distributed peers. This models is then downloaded and personalised by local devices [11]. FL helps enhance privacy of data owners. Furthermore using FL, distributed data can be used to train robust ML models and used as ML-as-a-service. However, in such applications, results from a single model cannot be trusted completely because of potential negative consequences of false positives and false negatives. One solution for this problem is to query different models for cross-validation before any results are accepted. However, different models can provide different prediction results and confidence scores for a given input sample. Choosing the right results (prediction) among many can be difficult and challenging. In other words, FL enables collaborative training of joint model but cannot perform *consensus* over the distributed predictions once the global training has been completed and deployed at local devices. The local devices generally personalise the distributively trained model using their local data.

Methods like Byzantine fault tolerance can address such issues in distributed computing, nevertheless, such methods work based on simple majority voting [2], without considering confidence score for results. Confidence scores of results play an important roles, which can be explained by an illustrative example: n peers all with high confidence on a result are clearly better than n peers all with less confidence on the same result. Hence, it is useful for consensus algorithms to consider confidence scores of all participating peers in order to produce more confident consensus results.

---

*The full edition of this short poster paper can be found on arXiv.org as a preprint at https://doi.org/10.48550/arXiv.2212.14050.
†Corresponding co-authors: Ali Raza and Shujun Li.

SAC '23, March 27–31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03.
https://doi.org/10.1145/3555776.3578601

To the best of our knowledge, there is only limited related work, which tries to achieve consensus during the training phase of multiple machine learning models [10]. Such methods cannot be applied to scenarios where multiple pre-trained machine learning models work together to achieve a consensus for unseen data, which remains an open research question.

Swarm intelligence, a natural phenomenon in many organisms, has been used to get the (sub-)optimal choice among groups, schools and colonies. Artificial swarm intelligence of distributed models can often achieve superior results over individual models who participate [3].

In this paper, inspired by swarm intelligence, we propose a novel consensus algorithm called Proof of Swarm (PoSw). The proposed algorithm can be used to obtain (sub-)optimal consensus among all the peers by effectively considering output probability distributions (confidence scores) over all the candidate outputs to obtain an agreement (consensus) among all the peers over the output results. The proposed algorithm does not involves complex computation, so it can be used in resource constraint edge device(s).

The rest of the paper is organised as follow. Section 2 presents the proposed PoSw consensus method. Section 3 shows experimental results. The last section concludes the paper.

2 PROPOSED METHOD

We will use an indicative example to explain the proposed PoSw consensus method. Let us suppose that five edge devices have trained a global model for classification of ECG signals into five classes, S, V, F, N, and Q, using federated learning. After receiving the globally trained model, each edge device personalises the global model using its local data. Now, an input sample is given to each of the edge device. Each edge device will output a classification result (confidence score over candidate classes). Here, the output is actually the probability distribution given by the softmax function of the trained model for the given input at each edge device E_i ($i = 1, 2, 3, 4, 5$). Suppose that edge devices 1 and 2 predicted class N, while edge devices 3, 4 and 5 predicted class V, Q and F, respectively. This is because the model considers the class with the highest probability as the predicted class. Since not all of the peers have the same output class for the same input, trusting any particular result is not feasible. Therefore, to achieve consensus in this situation, our proposed PoSw method considers the confidence scores of all edge devices' results so that results with higher confidence can be prioritised. Assuming there are n edge devices, the general workflow of the proposed PoSw method can be described as follow.

1. Each edge device E_i broadcasts (C_i, p_i) to the whole network, where C_i is the local "best" class label with the maximum probability p_i. If more than one class label has the same maximum probability, randomly choose one.
2. For each unique class label $c \in \{C_i\}_{i=1}^{n}$, count the number of votes it receives among all edge devices and denoted it by $\#(c)$. Denote the maximum number of votes by $M = \max\{\#(C_i)\}_{i=1}^{n}$. Now each edge device calculates a set of global "best" class labels C as follows:
    (a) If there is a single class label c with the maximum number of votes M, $C = \{c\}$.
    (b) If there are more than one class label with the maximum number of votes M, then calculate the sum of the probabilities of each such class label c according to Eq. (1). If a single class label c has the maximum sum, then $C = \{c\}$.
    $$P(c) = \sum\nolimits_{\forall i, C_i = c} p_i. \tag{1}$$
    (c) If more than one class label with the maximum probability sum, then set C to be the set of all such labels.
3. Each edge device satisfying $C_i \notin C$ performs the *move* function, by assigning C_i to be the next class label C_i' with the next highest probability p_i', and then re-broadcasting (C_i', p_i'). If an edge device exhausts all class labels, it goes back to the class label with the highest probability (i.e., "resets" the whole process).
4. Repeat the above two steps until the status of the whole network converges, e.g., $\forall i$, $C_i \in C$.

For the proposed PoSw algorithm, we can prove the following important theorem.

Theorem 1. *Assuming there are $N > 1$ edge devices and $K > 1$ class labels, the above-described PoSw algorithm will converge to reach a consensus after at most $K(K-1)$ rounds.*

Proof. For the i-th round of the algorithm, denote the set of the global best labels by C_i, and assume that n_i edge devices that vote for one of the labels in C_i. If $n_i = N$, the algorithm reaches the end so can stop. Therefore, we now only consider the case of $n_i < N$. In the following, we show for all possible cases, after a finite number of steps, n_i will increase by at least one, i.e., $n_{i+j} \geq n_i + 1$, where j is a finite number.

According to the proposed PoSw algorithm, only the $N - n_i$ edge devices that did not vote for any class labels in C_i should perform the *move* function. Assume after the moves, the new class labels of $N - n_i$ edge devices choose are $C_1, \ldots, C_{N-n_i}$. Consider two different scenarios.

Scenario 1) $\exists c \in C_i$, which appears at least once in $C_1, \ldots, C_{N-n_i}$: In this case, $n_{i+1} \geq n_i + 1$ will always hold since no matter which class label(s) (C_i or one or more in $C_1', \ldots, C_{N-n_i}'$) is/are selected, the number of votes will be no less than $n_i + 1$, the minimum number of votes c gets in the new round.

Scenario 2) $\forall c \in C_i$, c does not appear in $C_1, \ldots, C_{N-n_i}$: In this case, the number of votes of each global best class label in C_i remains unchanged. Now let us consider two sub-scenarios.

Scenario 2a) If one or more of $C_1, \ldots, C_{N-n_i}$ get more votes than n_i, then C_{i+1} will change to the set of those new class label(s), and $n_{i+1} \geq n_i + 1$ after just one round.

Scenario 2b) If none of $C_1, \ldots, C_{N-n_i}$ get more votes than n_i, let us consider all future rounds of the algorithm. If for any round $j > i$, Scenario 1 or 2a happens then $n_j \geq n_i + 1$ will hold, therefore, the only possibility for a consensus to not take place will be when the algorithm is "trapped" within Scenario 2b forever. Now let us assume that the algorithm is indeed trapped in Scenario 2b forever. In this case, the global best class labels appear in all future rounds as follows:

$$\overbrace{C_1, \ldots, C_1}^{i_1+1 \text{ to } i_1+f_1 \text{ rounds}} \quad , \ldots, \quad \overbrace{C_K, \ldots, C_K}^{i_{K-1}+1 \text{ to } i_{K-1}+f_K \text{ rounds}} \quad , \ldots$$

Assume $\exists k \in \{1, \dots, K\}$ so that $f_k \geq K - 1$. Then, all the edge devices that did not vote for any class label in C_k in the $i_k + 1$-th round would have exhausted all the remaining $K - 1$ candidate class labels, which must include at least one label $c \in C_k$. If so, the number of votes c gets should have increased by at least one before reaching the $i_k + f_k$-th round. This means that since the i_1-th round the number of votes of the global best must have increase at least by one in at most $K(K - 1)$ rounds. On the other hands, if the $\forall k \in \{1, \dots, K\}$ so that $f_k < K - 1$, let us prove that $\forall i > j, C_i \cap C_j = \emptyset$. The nature of being trapped in Scenario 2b is that the global best class label(s) can only change if $P(C_j) > P(C_i)$ since the number of votes remains n_i. This means that $P(C_K) > \cdots > P(C_1)$. Since the probability of any class label is static, the inequality implies $\forall i, j \in \{1, \dots, K\}$, $C_i \neq C_j$. Given there are only K class labels, we have $\cup_{i=1}^{K} C_i = \{1, \dots, K\}$. Now, after C_K, the output of the algorithm will not change since none of the class labels in $\{1, \dots, K\} - C_K$ will have a higher probability than $P(C_K)$. Therefore, C_K will be the output forever, i.e., $f_K = \infty$, which contradicts to the previous assumption that $f_K < K - 1$. Therefore, the algorithm cannot be trapped forever in Scenario 2b and will go to other scenarios in at most $K(K - 1)$ rounds, at which point the number of votes will increase by at least one.

Combining all the above scenarios together, we can see the maximum rounds needed to let the number of votes the global best class label(s) to increase by at least one is $\max(1, K(K - 1))$. Since $K > 1$, we can get $K(K - 1) \geq 2$ so the maximum number of rounds is $K(K - 1)$. □

A corollary from Theorem 1 is the following.

Corollary 1. *Once the PoSw algorithm produces an output* C *with* $\lfloor K/2 \rfloor + 1$ *(a simple majority of all votes),* C *will be the final converged solution so the algorithm can stop.*

3 EXPERIMENTAL RESULTS

In this section, we show some experimental results of a performance analysis of the proposed PoSw method. To evaluate the proposed method, we trained a convolutional neural network-based five-class classifier for ECG classification in a federated setting. We used five edge deceives in the FL setting to collaboratively train a global model. After training the global model, each edge device downloads the global model and fine tunes it for further classification. We tested each locally tuned global model using a test dataset. Then we used the proposed PoSw method to achieve a consensus among the edge devices for the same test dataset. We used the widely known MIT-BIT arrhythmia dataset [8] to test the proposed algorithm. The training samples were equally independent and identically distributed among each client. We also kept 1,000 samples for testing which were not used by any client. The PoSw method was implemented with TensorFlow 2.9.0 as the machine learning library, our simulations were run on a computer with an Intel core i-6700HQ CPU and 32 GB RAM. Figure 2 presents the number of rounds taken by 1,000 simulations of the PoSw method to reach a mutual consensus for each input sample. For most simulations, the PoSw algorithm was not needed because all five edge classifiers predicted the same class label. For other cases, the PoSw algorithm was run to obtain the results, mostly within just one or two rounds and in one case after 20 rounds (which is the maximum number of rounds according to Theorem 1).

Figure 1 presents the time taken (in seconds) by 1,000 software-based simulations of the proposed PoSw method to finally achieve a mutual consensus for each input sample. It can be seen that the proposed PoSw method took on average less than a seconds to achieve a mutual consensus among the participating edge devices.

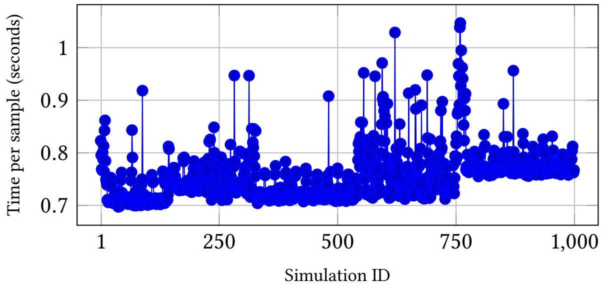

Figure 1: Performance of the proposed PoSw consensus method (in terms of the time taken by 1,000 software-based simulations to achieve a mutual consensus).

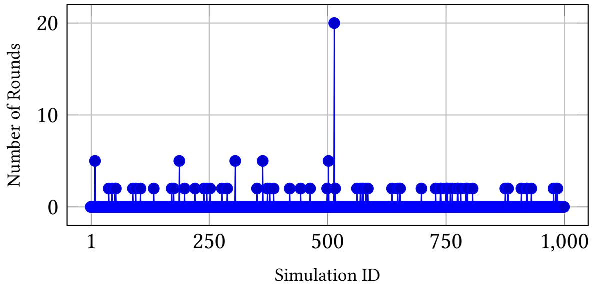

Figure 2: Performance of the proposed PoSw consensus method (in terms of the number of rounds needed by each of 1,000 simulations to achieve a mutual consensus).

In order to compare the classification performance (in terms of accuracy, defined by #(classification errors)/#(samples)) of the proposed PoSw-based consensus (ensemble learning) model against the five local models and the FL-based global model, we calculated the accuracy metrics of all the seven models using the same test dataset. Figure 3 presents the results. It can be observed that the proposed PoSw-based model has the best accuracy, among all models. This indicates that using multiple local models to collectively make a decision can help reduce error rates, even outperforming the FL-based global model.

4 COMPARISON

In this section, we compare key features of our proposed PoSw with the most commonly used ensemble learning methods for classification, i.e., bootstrap aggregation or bagging [1], boosting [4], stacking [9] and BFT [2]. Bagging trains a number of models on different samples of the same training dataset. The predictions

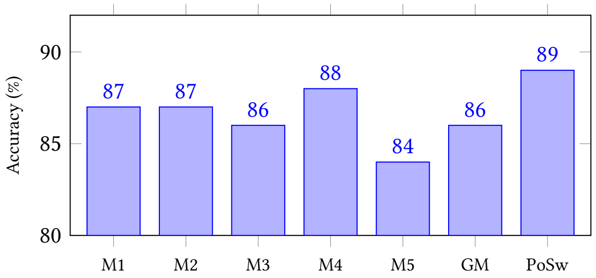

Figure 3: Comparison of the classification accuracy of the PoSw-based consensus (ensemble learning) model with the global model and the five local models.

made by all ensemble member are then combined using a statistical method like (weighted) majority voting. Bagging can partially solve the tie problem by reducing the probability of having a tie if weighted voting is used, and a rule can be set to decide which result to output in case of a tie. Boosting combines several weak models sequentially by assigning weights to outputs of each model. Then it inputs the incorrect result from the first model in sequence to the subsequent model. Similarly, stacking involves training several weak models and then training a meta model using the outputs of the weak learners. In a simple BFT, a majority voting is used to determine the final output. In addition to ensemble learning methods, many swarm intelligence (SI) and blockchain-based methods can be used to achieve a mutual consensus among multiple parties, but we are not aware of any SI-based and blockchain-based methods that can address the tie problem in our application area[1].

Table 1 presents the comparison of our proposed method PoSw with the above-mentioned state-of-the-art ensemble learning methods. It can be observed that PoSw has more desirable features by providing a mutual consensus among all the participants unlike bagging and BFT where the result is achieved with a simple (weighted) majority voting. Moreover, in case of FL, boosting is not suitable because of its sequential training nature. Similarly, stacking involves training a meta model using weak learners. In FL, a global model is achieved by aggregation local models, which are then fine-tuned locally. Hence, applying stacking again would make no difference.

Table 1: Comparison of PoSw with selected state-of-the-art ensemble learning and distributed consensus methods

Method	Mutual Consensus	Tie resolution
Bagging	No	Partially
Boosting	Not Applicable	Not Applicable
Stacking	Not Applicable	Not Applicable
BFT	No	No
Other SI-based methods	Yes	No
Other Blockchain-based methods	Yes	No
PoSw	Yes	Yes

5 CONCLUSIONS

In this article, we proposed a novel distributed consensus algorithm called PoSw (Proof of Swarm) to achieve ensemble learning in federated learning applications. Using the proposed PoSw method, distributed peers can always converge to reach a consensus in $K(K-1)$ steps, where K is the number of classes of the classification problem. Additionally, the proposed PoSw method can efficiently solve tie events. Unlike the classical distributed consensus algorithm, such as Byzantine fault tolerance the proposed algorithm does not makes consensus based on a simple majority voting, instead, it considers confidence scores of predicted class labels of all peer classifiers and tries to achieve a more optimised consensus decision among all the peers in the network. Using experimental results of an ECG classification task with five classes, we show that the proposed PoSw-based ensemble learning model outperformed all local models and also the FL-based global model, in terms of the overall accuracy.

ACKNOWLEDGMENTS

This study was supported by the grant (REF LABEX/EQUIPEX), a French State fund managed by the National Research Agency under the frame program "Investissements d'Avenir" I-SITE ULNE / ANR-16-IDEX-0004 ULNE.

REFERENCES

[1] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140. https://doi.org/10.1007/BF00058655
[2] Miguel Castro and Barbara Liskov. 2002. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems* 20, 4 (2002), 398–461. https://doi.org/10.1145/571637.571640
[3] Amrita Chakraborty and Arpan Kumar Kar. 2017. Swarm intelligence: A review of algorithms. *Nature-Inspired Computing and Optimization* (2017), 475–494.
[4] Sergio González, Salvador García, Javier Del Ser, Lior Rokach, and Francisco Herrera. 2020. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion* 64 (2020), 205–237.
[5] Heiko Hamann. 2013. Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance. *Swarm Intelligence* 7, 2 (2013), 145–172. https://doi.org/10.1007/s11721-013-0080-0
[6] DonHee Lee and Seong No Yoon. 2021. Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges. *International Journal of Environmental Research and Public Health* 18, 1 (2021), 271. https://doi.org/10.3390/ijerph18010271
[7] Xu Mengfan and Li Xinghua. 2022. FedBC: An Efficient and Privacy-Preserving Federated Consensus Scheme. In *Proceedings of the Security and Privacy in Social Networks and Big Data: 8th International Symposium*. Springer Nature, 148.
[8] George B. Moody and Roger G. Mark. 2001. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50.
[9] Bohdan Pavlyshenko. 2018. Using Stacking Approaches for Machine Learning Models. In *Proceedings of the Second International Conference on Data Stream Mining & Processing*. 255–258. https://doi.org/10.1109/DSMP.2018.8478522
[10] Stefano Savazzi, Monica Nicoli, and Vittorio Rampa. 2020. Federated learning with cooperating devices: A consensus approach for massive IoT networks. *IEEE Internet of Things Journal* 7, 5 (2020), 4641–4654. https://doi.org/10.1109/JIOT.2020.2964162
[11] Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. 2021. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research* 5, 1 (2021), 1–19. https://doi.org/10.1007/s41666-020-00082-4

---

[1] There are other SI-based and blockchain-based methods that can address ties, however, their application is limited in our application area. This is because such methods are not directly applicable to federated learning, where typical features such as a noisy swarm and rankings of clients/edges are lacking in case of SI-based methods and stacking [5], and high computational power in case of blockchain-based methods [7].