IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 19, 2021, accepted August 8, 2021, date of publication August 18, 2021, date of current version August 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105929

Benchmarking PySyft Federated Learning Framework on MIMIC-III Dataset

ANDRIUS BUDRIONIS[1], MAGDA MIARA[2], PIOTR MIARA[2], SZYMON WILK[2], AND JOHAN GUSTAV BELLIKA[1]

[1]Norwegian Centre for E-health Research, University Hospital of North Norway, 9019 Tromsø, Norway
[2]Faculty of Computing and Telecommunications, Poznan University of Technology, 60-965 Poznan, Poland

Corresponding author: Andrius Budrionis (andrius.budrionis@ehealthresearch.no)

This work was supported in part by the Research Council of Norway to Norwegian Centre for E-health Research, University Hospital of North Norway, under Grant 248150/O70, and in part by the Institute of Computing Science at the Poznan University of Technology.

ABSTRACT The adoption of the advanced data analytics methods has been limited in industries governed by strict data reuse regulations, such as healthcare. Barriers to data access and sharing have affected numerous research and development initiatives in healthcare resulting in major delays, extensive use of resources for data access and findings originating from datasets that are too small to be generalizable. Federated machine learning presents a solution to the problems health data analytics projects are facing by providing a way of complying with strict regulatory requirements without sacrificing privacy. Computing frameworks supporting federated machine learning are still in their infancy and their performance in realistic settings has been studied only to a limited extent. To expand the existing knowledge on federated learning in realistic deployment settings three groups of experiments comparing the performance of a neural network-based model trained in federated manner to that of an equivalent baseline model trained on centralized data storage were designed. Experiments were conducted on the MIMIC-III dataset and modelled a binary classification problem predicting in-hospital mortality. The effect that varying amounts of data, number of computational nodes, and data distribution in the federated network had on model performance and on training and inference durations were studied. Experiments demonstrated predictive performance comparable to that of the baseline for models trained in federated settings in terms of area under the ROC and F1 scores. Data distribution across computing nodes showed minimal to no effect on model performance or on training and inference durations. However, federated model training and inference took approximately 9 and 40 times longer, respectively, than the equivalent tasks executed in centralized settings. These results indicate that federated learning is a viable solution for enabling advanced data analytics in environments regulated by strict privacy requirements.

INDEX TERMS Federated learning, machine learning, PySyft, MIMIC-III, learning healthcare system.

I. INTRODUCTION

Advanced data analytics methods are revolutionizing industries by making business processes increasingly data-driven. While the use of machine learning (ML) and artificial intelligence (AI) have reached high market penetration and scale in industries such as retail, finance, manufacturing, and education, their use in the field of healthcare is lagging. The potential for using AI in a healthcare context is highly debated and hyped [1], [2]. If this potential is to be realized, however, it is important to acknowledge that the healthcare sector faces challenges unlike those affecting other industries. The complexity of the healthcare landscape—influenced by medical practice norms, commercial interests, and political and economic factors—is difficult to re-engineer using AI algorithms. Medical data infrastructure, which is essential to best-performing, data-greedy algorithms, is often fragmented due to technological, legal, and organizational barriers, restricting access to wider and more representative datasets, especially those accumulated in large, centralized data storage [3], [4]. While solutions for building large data repositories and data reuse infrastructure often exist, organizational barriers and data privacy considerations are more difficult to address. Similarly, sharing the data by storing them in a centralized data warehouse outside of institutional control raises privacy concerns and may potentially have commercial and ethical implications. Health institutions value autonomy and are often reluctant to disclose patient data for research and public

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

health purposes [5]. Since these problems are here to stay, alternative methods for training and testing ML algorithms are required [6].

It is possible to mitigate the consequences of data fragmentation using federated ML approaches [7]–[10]. These approaches circumvent legal and organizational barriers by moving the model to the data instead of moving data to the model (centralized storage), as is the case in approaches using traditional ML algorithms. Federated data access methods, when combined with privacy-preserving techniques, provide a technology stack that supports ML algorithm training while preserving data privacy. The advantages of this approach have been acknowledged by major technology companies such as Google and open-source communities developing federated learning libraries. At the time of writing, three major frameworks supporting federated ML are available: TensorFlow Federated (Google, US, https://www.tensorflow.org/federated), Federated AI Technology Enabler (FATE, Webank AI Department, China, https://www.fedai.org), and PySyft (OpenMined, open-source community, https://www.openmined.org). These frameworks provide means for integrating distributed data sources into model training processes without violating legal and organizational norms.

Federated ML frameworks are still relatively young in comparison to mainstream ML algorithms requiring centralized data storage. Immaturity concerns combined with the nature of distributed datasets raise numerous questions concerning reliability, accuracy, and scalability. These concerns originate from the added complexity of performing data analytics on distributed datasets while preserving privacy. For instance, none of the parties involved in data processing has the complete data used for model training, making it impossible to verify the accuracy of results using established data analytics tools. Variables including distribution of data across the network of computational nodes, as well as the number and size (amount of data) of nodes may influence system performance and the accuracy of results. However, these variables and their effects have only been researched to a limited extent.

A study by Purushotham *et al.* [11] benchmarked deep learning algorithms on the Medical Information Mart for Intensive Care-III (MIMIC-III) dataset and compared deep learning models to more traditional ML approaches. Our study aims to expand upon existing research by reproducing the experiments reported by Purushotham *et al.* [11] in federated settings and comparing model performance metrics for centralized and federated model training.

Similar work has already been done by Lee and Shin, who benchmarked models trained using TensorFlow Federated (Google, US, https://www.tensorflow.org/federated). The experiments demonstrated that models trained in federated settings reach comparable predictive performance to the models trained on a centralized dataset [12]. However, these experiments were performed in rather minimalistic settings (3 computing nodes), that were sufficient to demonstrate the feasibility of the federated model training. Our work supplements these findings with the experimental results from more realistic deployment settings and provide trends showing how predictive performance, model training and inference durations are affected by the varying configurations of the computing node network.

Our main contributions to the existing research come directly from the experiments described in this manuscript. Using our experimental setup, we demonstrated that a neural network model trained in federated settings achieves predictive performance that is comparable to the baseline model trained on a single data repository. Model performance is minimally affected by the number of nodes in the system, amount of data hosted by a single node and data distribution on the computing nodes. Performance trends provide indications on the scaling potential of the federated learning system in terms of model training and inference durations.

The paper is organized as follows: we start the paper with the Introduction to the problem statement and suggest a potential solution that requires more research and evaluation in realistic settings. We then present the Method for evaluating the selected Federated ML framework, including data, evaluation metrics and baseline measures. Results section presents the results from the performed experiments contrasting them to the corresponding baseline measures. Discussion section places our findings in a broader context, comparing them to the existing research. The paper is concluded by summarizing our findings at a more general level and providing implications for further research.

II. METHOD

A. SELECTED FRAMEWORK AND SETUP

Experiments were performed using PySyft version 0.2.9, Python version 3.7.6, and PyTorch version 1.4.0. PySyft virtual and real workers were deployed on a Google Compute Engine (n1-stadard-32 instance, 32 virtual CPUs of type Intel(R) Xeon(R) CPU @ 2.30GHz, 120GB RAM, no GPU). Virtual workers are a construct in PySyft used for simplifying experimental setups. These workers run as separate processes on a single virtual machine and receive their own datasets at runtime. Real computational nodes are standalone services and can be deployed on one or several physical computing nodes. Given PySyft's immaturity, experiments were begun using virtual workers only and were later repeated using both virtual and real workers. To simplify the experimental setup, real nodes residing in Docker containers, deployed on a single Google Compute Engine instance, were used as a substitute for physical computational nodes. All experiments were repeated 5 times using random data splits and average values for the selected metrics have been reported.

A model-centric setup was used. In this approach, the model is hosted by a coordinating node, while computational nodes download the model and train it on local datasets. After model training is complete, the difference in weight between the downloaded and trained models is calculated and sent back to the coordinating node. The coordinating node collects model differences from all computational nodes, aggregates them, and uses them to update the global model.

B. DATA

A dataset equivalent to the one used by Purushotham *et al.* [11] was used in the present study. Following detailed preprocessing instructions, feature sets were derived from MIMIC-III dataset. To limit the scope of our experiments, only one feature set, referred to as Feature set A [11], was used. Feature set A contains 35,637 data rows (examples) and 17 features, such as chronic diseases, admission type, and patient age. These features were extracted for the first 24 hours after ICU admission. A detailed description of data preprocessing and feature engineering is available in the original paper [11].

C. EXPERIMENTS AND METRICS

To shed more light on how federated learning framework performance is affected by realistic deployment scenarios—characterized by varying amounts of potentially skewed and unbalanced data residing in some computational nodes—a series of experiments was designed. These experiments were split into three groups based on their system configuration:

1. Data – the number of nodes in the network was kept constant (n = 32), while the amount of data in the system increased. Data was uniformly distributed across the nodes.
2. Nodes – the amount of data in the system was kept constant (maximum), while the number of computational nodes increased from 1 to 128. Data was uniformly distributed across the nodes.
3. Distribution – the amount of data in the system (maximum) and the number of computational nodes (n = 32) were kept constant, while data were distributed across the nodes following these distributions: uniform (baseline), linear, beta left skewed, beta centered, and beta right skewed (Figure 1).

Predictive performance of the models trained in the aforementioned experimental setups was measured in terms of ROC AUC and F1 score. Model training and inference durations were measured in seconds. Experimental setups are summarized in Table 1.

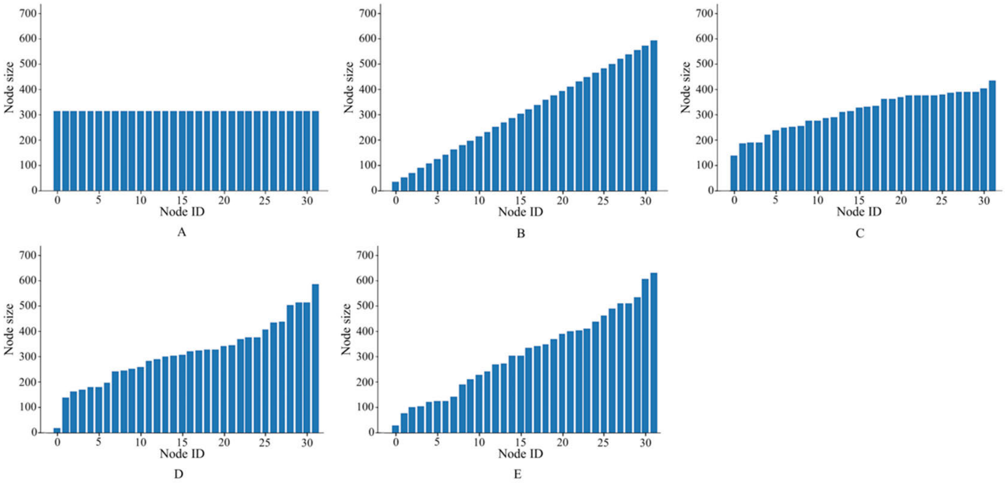


FIGURE 1. Data distribution in computational nodes. A – uniform, B – linear, C – beta left skewed, D – beta centered, E – beta right skewed.

TABLE 1. Configuration of experiments.

Experiment group	Number of nodes	Training data	Data distribution	Metrics
1. Data	Fixed (n = 32)	Increasing (50, 100, 500, 1000, 5000, 10000)	Fixed (uniform)	ROC AUC, F1 score, training and inference durations (seconds)
2. Nodes	Increasing (n = 2, 4, 8, 16, 32, 64, 128)	Fixed (maximum)	Fixed (uniform)	
3. Distribution	Fixed (n = 32)	Fixed (maximum)	Changing (uniform, linear, beta left, beta center, beta right)	

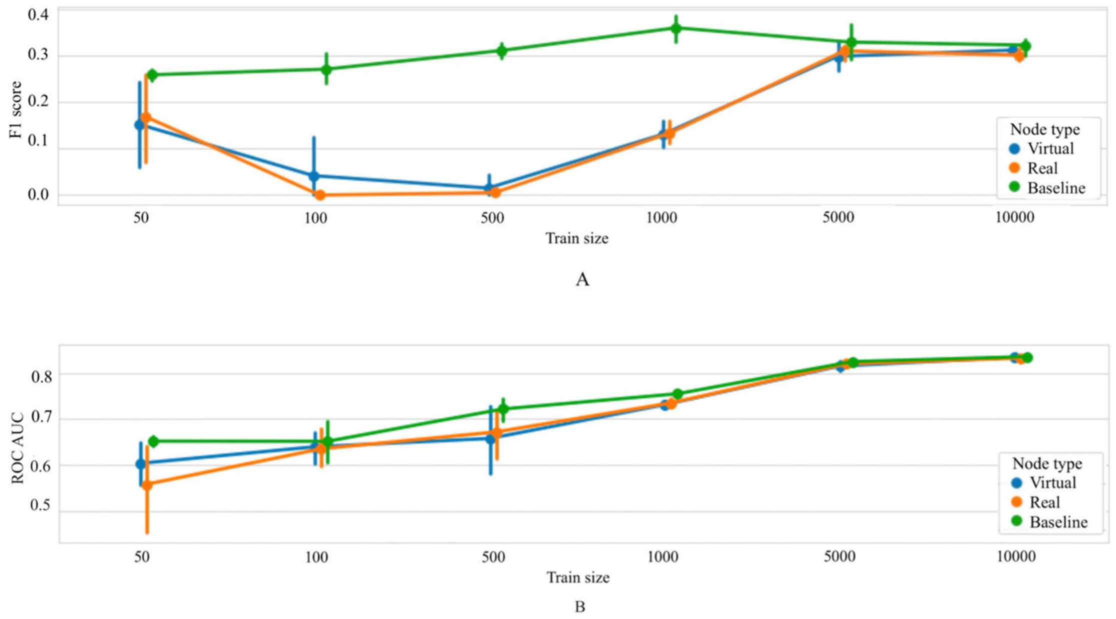


FIGURE 2. Influence of data amount on the performance of the federated model. A – F1 score. B – ROC AUC. Node type refers to the type of computational nodes used in the experiment: Virtual – PySyft Virtual workers, Real – PySyft computational nodes residing in separate docker containers. Baseline – baseline measure referring to model performance trained in centralized settings.

D. PREDICTION TASK AND NEURAL NETWORK MODEL

While several variations of mortality prediction (e.g., short-term, long-term) were modelled by the authors of the original paper [11], the experiments described herein used a binary classification task of predicting in-hospital mortality. A neural network with the same architecture as the one reported by Purushotham *et al.* [11] (i.e., a feed-forward network with sigmoid output layer) was used for this task.

E. BASELINE

Model performance in a centralized training scenario was used as a baseline for evaluating the models trained in federated settings. Results reported by Purushotham *et al.* [11] presented sufficient benchmark values for model performance evaluation, however, model training and inference duration metrics were lacking. Therefore, training scripts and parameters reported in the original paper [11] were reused to reproduce the performance benchmark which was supplemented with the lacking measures. Running these scripts on the entire dataset resulted in performance metrics values that were very close to those reported in the original paper [11]. Minor discrepancies were attributed to randomness in the training process, for instance random network initialization and random order of data samples.

III. RESULTS

Experiment results were grouped according to system configuration, as described in the Method section.

A. DATA

This group of experiments studied how model performance is affected by the amount of data residing in the federated learning node network. A fixed number (n = 32) of PySyft workers were allocated an increasing amount of data used for model training. Figure 2 demonstrates how predictive performance in terms of ROC AUC and F1 scores was affected by the amount of data in the system. Baseline refers to the corresponding model performance in a centralized training scenario.

Increasing the amount of data used for model training resulted in a better-performing model (Figure 2). As soon as a sufficient amount of data was provided, ROC AUC and F1 score values for a federated model were close to those of the model trained in a centralized manner.

Figure 3 shows how model training and inference durations were affected by the increasing amount of data in the network. Subgraph A illustrates the time required for collecting pointers to the nodes that participate in model training. The remaining subgraphs show how model training and inference durations were affected by the increasing number of data rows used for model training.

Figure 3 shows that higher amounts of training data increased the time required for training the model, as expected. Subgraph A suggests that time overheads for collecting pointers to the nodes participating in model training and hosting data are minimally influenced by the amount of data. While training duration increases with the amount of training data, prediction time remains relatively stable.

B. NODES

This group of experiments studied how model performance was influenced by the number of computational nodes in the network. The amount of data distributed among the nodes was kept constant (maximum) while the number PySyft workers

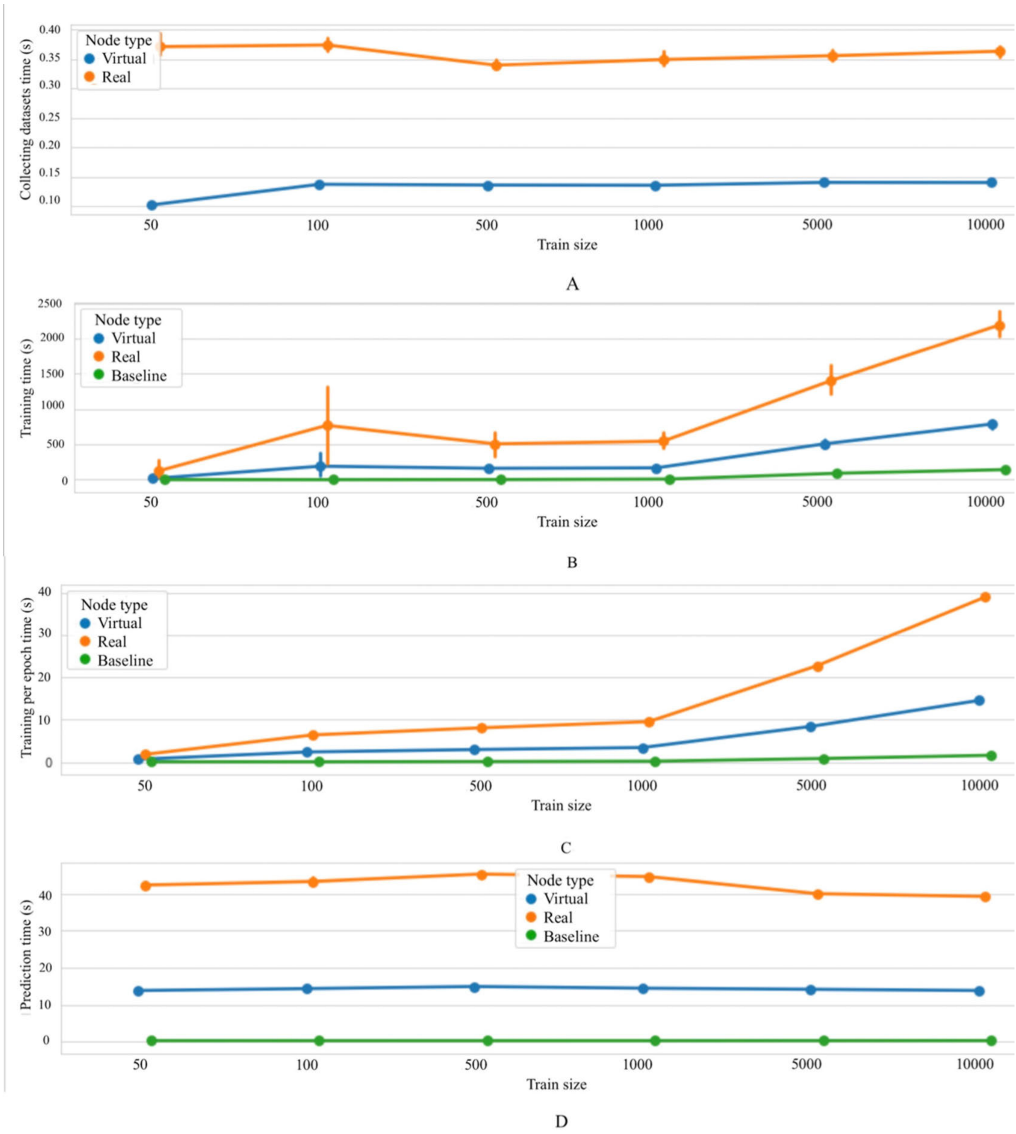


FIGURE 3. **Influence of the amount of data in the network on training and prediction times when number of computing nodes is constant (n = 32). A – Collecting pointers time (s). B – Training time (s). C – Training per epoch time (s). D – Prediction time (s). Node type refers to the type of computational nodes used in the experiment: Virtual – PySyft Virtual workers, Real – PySyft computational nodes residing in separate docker containers. Baseline – baseline measure referring to model performance trained in centralized settings.**

ranged from 1 to 128 for virtual nodes and from 1 to 32 for real nodes (Figure 4). Memory constraints prevented experiments with more than 32 real PySyft worker nodes from being performed. The baseline subgraph depicts model performance in a centralized training scenario.

The number of virtual workers (computational nodes) had a minor influence on model performance (Figure 4). ROC AUC and F1 score values were similar for models trained in federated and centralized settings (Figure 4). A noticeable drop in performance appeared when scaling up the experiment to a maximal configured number of nodes. This drop would suggest that a network consisting of a large number of nodes, each hosting little data, may not be an optimal scenario for training the best-performing model (each node hosted approximately 557 and 278 examples in 64 and 128 node setups, respectively). However, it is important to note that the observed drop in performance was relatively small ($<6\%$ in F1 score and $<1\%$ in ROC AUC) and may not have major consequences in real-life settings.

Similarly, the way in which the number of workers in the network influenced model training and inference times was tested while keeping a constant amount of data in the system (Figure 5).

Increasing the number of computational nodes in the network showed interesting trends. Naturally, the time required to collect pointers to the nodes increased when the number

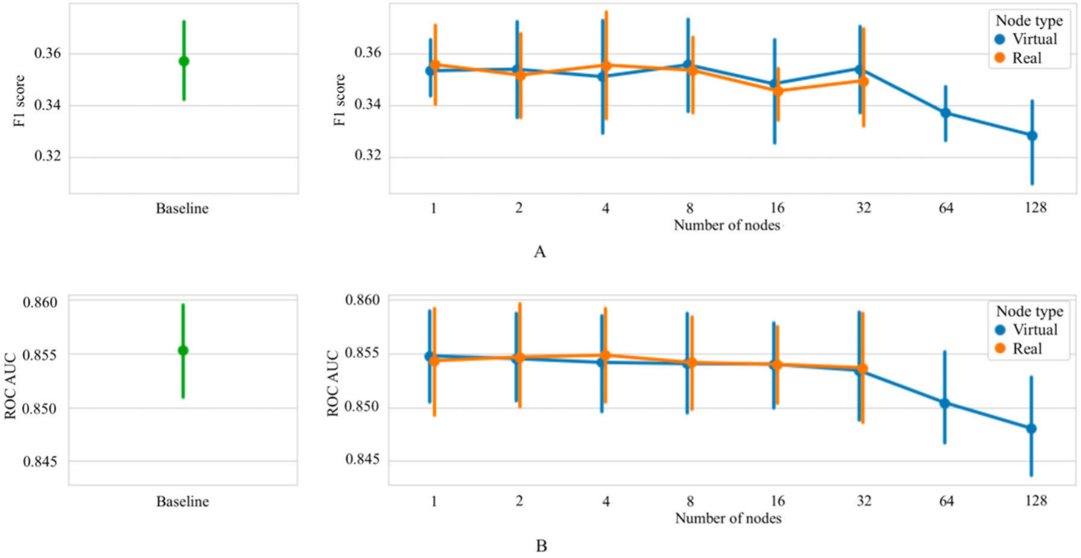


FIGURE 4. Dependency of model performance on the number of computational nodes in the system. A – F1 score. B – ROC AUC. Node type refers to the type of computational nodes used in the experiment: Virtual – PySyft Virtual workers, Real – PySyft computational nodes residing in separate docker containers. Baseline – baseline measure referring to model performance trained in centralized settings.

of nodes grew. Model training per epoch followed a slightly increasing trend that could be explained by the computational overhead required for aggregating models trained on each node into a global model (Figure 5).

Although each training epoch took longer when the number of nodes increased, the time required for overall model training showed an opposite pattern—the more nodes in the network, the faster the entire model was trained. Faster convergence and early stopping may explain faster model training in a larger network of nodes. Prediction time increased when the number of computational nodes in the network grew, however this increase was relatively small.

C. DISTRIBUTION

Five data distributions (uniform, linear, beta left skewed, beta centered, and beta right skewed, Figure 1) were followed when assigning partial datasets to computational nodes. Mortality prediction models were trained using these data distributions across the nodes to check the effect on predictive model performance and time required for training and inference (Figure 6).

Figure 6 shows that data distribution across the network had minimal influence on model performance—both ROC AUC and F1 scores were minimally affected, and no specific trends were observed. The same patterns are visible in training and prediction times, showing minimal effect of data distribution across the computing nodes.

IV. DISCUSSION

The experiments presented in this paper illustrate the initial results of comparing ML models trained in centralized and federated manners. Models were evaluated in terms of predictive performance and training and inference durations. The effects of data amount, number of nodes, and data distribution in computational nodes were studied, giving an indication for model performance and system scalability in scenarios where the computational node network is unbalanced in terms of node size.

There were minimal differences in model performance, measured by ROC AUC and F1 scores, between centralized and federated approaches, as long as a sufficient amount of data was provided during model training. Model performance was not affected by the number of computational nodes in the system. Data distribution across the nodes (in terms of the amount of hosted data) did not have an influence on model performance.

Our findings support existing research on the performance of federated learning frameworks. Similar to our experiments, Zhu *et al.* compared the performance of federated PySyft and TFF models to equivalent models trained on centralized data for a text recognition task. Their results showed that federated learning models could achieve comparable (and in some cases even higher) accuracy [13]. A recent study modelled in-hospital mortality prediction task using MIMIC-III dataset and reported that neural network models trained using TensorFlow Federated framework reached comparable predictive performance to the equivalent state-of-the-art models trained in centralized settings [12]. Ziller *et al.* presented a system for privacy-preserving medical image analysis based on federated ML. Benchmarking system performance against locally-trained models showed minor differences across several reported metrics [14]. Our results for different models, training tasks, and data showed the same trends.

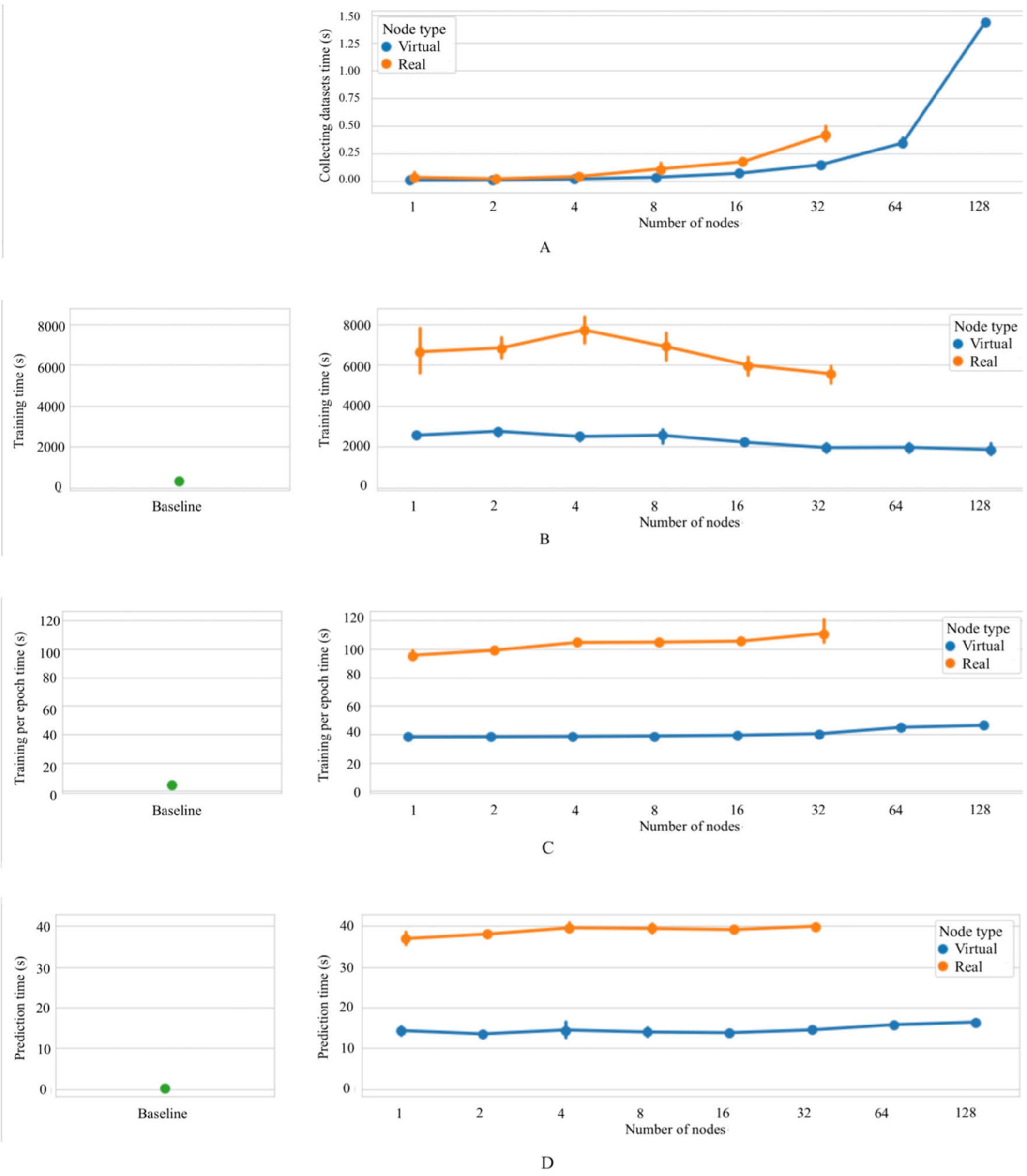


FIGURE 5. Influence of the number of computational nodes on model training and inference time using maximum amount of training data available in the dataset. A – Collecting datasets time (s). B – Training time (s). C – Training per epoch time (s). D – Prediction time (s). Baseline – baseline measure referring to model performance trained in centralized settings. Node type refers to the type of computational nodes used in the experiment: Virtual – PySyft Virtual workers, Real – PySyft computational nodes residing in separate docker containers.

Model training and inference durations showed interesting trends. As expected, experiments using virtual and real PySyft computing nodes took longer than when running them in centralized settings. In general, more data in the federated node network resulted in longer training times. Corresponding duration values for models trained on centralized data followed a similar pattern, however, the increase in durations was much slower when the amount of data grew. In comparison to the baseline, training in federated settings was approximately 3 and 9 times slower using PySyft virtual and real node setups, respectively.

Conversely, increasing the number of computational nodes while keeping the amount of data constant resulted in decreasing model training durations. Inference time showed a slightly increasing trend. Differences in inference times between centralized and federated setups were higher than differences in model training times. Making predictions for the entire testing dataset in federated settings took approximately 15 and 40 times longer than in a centralized scenario using virtual and real computing nodes, respectively. Differences in prediction durations may be explained by computational overheads introduced by the federated infrastructure.

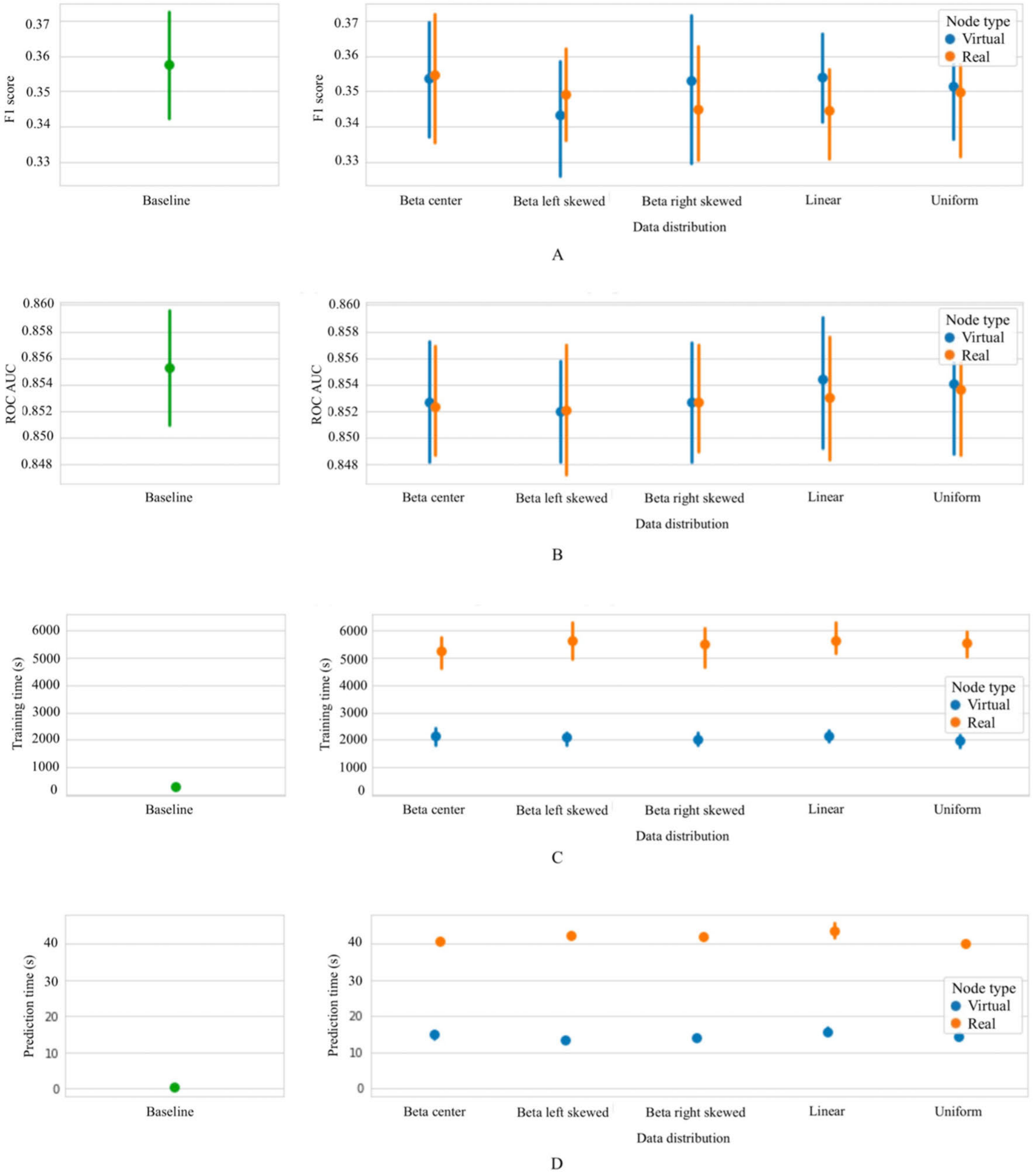


FIGURE 6. Influence of data distribution across the nodes on predictive model performance and training and inference durations. A – F1 score. B – ROC AUC. C – Training time (s). D – Prediction time (s). Baseline – baseline measure referring to model performance trained in centralized settings. Node type refers to the type of computational nodes used in the experiment: Virtual – PySyft Virtual workers, Real – PySyft computational nodes residing in separate docker containers.

Even though inference duration for an entire test set may seem relatively high compared to the baseline, in real-life deployments predictions are likely to be made for individual patients rather than big patient groups. Therefore, longer times for making predictions will likely become negligible.

Solutions to optimize model training in terms of faster convergence and shorter overall training durations were already suggested for centralized setups [15], [16]. For instance, non-iterative approaches for model training challenge the back-propagation algorithm and show clear advantages, especially when the number of hidden layers increases. The suggested methods freeze a randomly selected number of nodes in the network throughout the training process while determining their output weights analytically. The strengths of non-iterative approaches have been widely discussed in academic community and some applications solving practical problems have been demonstrated [15], [17]. The intersection of these approaches and federated learning frameworks presents numerous open questions, calling for more research.

Data distribution across the nodes suggested minimal influence on both predictive model performance and training and inference durations. This is a very positive finding, especially when considering the transition of relatively immature federated learning frameworks to production environments. In a realistic deployment, computing nodes are likely to be very different in terms of hosted data amounts, therefore it is important to clarify that network topology has little to no influence on the performance of the trained model and training and inference times. These findings are aligned with existing research on imbalanced and skewed datasets used for training ML models in federated settings. Comparable model performance was recently demonstrated on three datasets, including MIMIC-III used in our study, regardless of how unbalanced or skewed data hosted by the computing nodes was [12]. These results show that models trained in federated settings are able to achieve performance comparable to the models trained on centralized data.

A. LIMITATIONS

The results reported in this paper should be considered with the following limitations in mind.

Model performance experiments used a relatively simple neural network model that was inherited from the publication reporting the original experiments in centralized settings [11]. While this model was sufficient for the purpose of our experiments, it may not be generalized for more advanced deep learning models. Additional studies using various neural network architectures are needed to support our conclusions in a broader context.

Furthermore, our data distribution experiments do not consider data correlation within the nodes, which could become an important factor in certain cases. For instance, one doctor's office could mainly serve an elderly population, while another may only treat children. This and other less obvious data correlation problems could exist in data processing nodes and may affect the performance of the overall model.

Due to limited computing resources and additional complexity associated with infrastructure setup, we have not performed the experiments using real PySyft workers deployed on dedicated computing instances or physical hardware—a setup closest to real-life deployment settings. However, we do not expect that a move to dedicated computing instances or physical hardware would affect the performance (ROC AUC and F1 score) of the model. Model training and inference durations are expected to increase, adding time required for network operations to the reported numbers.

V. CONCLUSION

Federated learning frameworks are attracting major attention from industries, like healthcare, where centralizing data in a single repository is not possible. Federated learning provides a means of harnessing the power of ML in a regulation-compliant way and accelerating continuous learning from data generated in routine care—the Learning Healthcare System [18]–[20].

This paper demonstrated that the performance of ML models trained in a federated environment is comparable to those trained on centralized data storage. Federated models are not affected by unbalanced data distributions across network nodes.

However, training ML models in a federated environment has its cost. The efficiency of model training and inference suffers due to the added complexity of node orchestration, privacy preservation, and extra steps that are not existent in centralized approaches. Experiments using PySyft workers deployed in separate Docker containers within a single cloud compute instance showed that model training may take up to 9 times longer, and that inference time may increase by a factor of 40 in comparison to the model trained on centralized data. It is important to note that these numbers include minimal overheads for network operations, since all worker nodes reside in the same cloud compute instance. Real-life deployment will increase durations for both model training and inference.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu, who are the authors of the original experiments that were reproduced in this publication and who kindly helped to prepare datasets for their experiments.

They also wish to thank Google Cloud Platform for supporting this study and for providing them with the computing resources necessary to performing the experiments described herein.

REFERENCES

[1] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nature Biomed. Eng.*, vol. 2, no. 10, pp. 719–731, Oct. 2018, doi: 10.1038/s41551-018-0305-z.

[2] T. Davenport and R. Kalakota, "The potential for artificial intelligence in healthcare," *Future Healthcare J.*, vol. 6, no. 2, pp. 94–98, Jun. 2019, doi: 10.7861/futurehosp.6-2-94.

[3] T. Panch, H. Mattie, and L. A. Celi, "The 'inconvenient truth' about AI in healthcare," *NPJ Digit. Med.*, vol. 2, no. 1, Dec. 2019, Art. no. 77, doi: 10.1038/s41746-019-0155-4.

[4] T. S. Toh, F. Dondelinger, and D. Wang, "Looking beyond the hype: Applied AI and machine learning in translational medicine," *EBioMedicine*, vol. 47, pp. 607–615, Sep. 2019, doi: 10.1016/j.ebiom.2019.08.027.

[5] K. El Emam, J. Mercer, K. Moreau, I. Grava-Gubins, D. Buckeridge, and E. Jonker, "Physician privacy concerns when disclosing patient data for public health purposes during a pandemic influenza outbreak," *BMC Public Health*, vol. 11, no. 1, p. 454, Dec. 2011, doi: 10.1186/1471-2458-11-454.

[6] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, Dec. 2020, Art. no. 119, doi: 10.1038/s41746-020-00323-1.

[7] M. A. Hailemichæl, K. Y. Yigzaw, and J. G. Bellika, "Emnet: A system for privacy-preserving statistical computing on distributed health data," presented at the 13th Scandinavien Conf. Health Inform., Troms, Norway, Jun. 2015.

[8] J. G. Bellika, T. S. Henriksen, and K. Y. Yigzaw, "The snow system: A decentralized medical data processing system," in *Data Mining in Clinical Medicine* (Methods in Molecular Biology), vol. 1246. Clifton, NJ, USA: Springer, 2015, pp. 109–122, doi: 10.1007/978-1-4939-1985-7_7.

[9] J. S. Brown, J. H. Holmes, K. Shah, K. Hall, R. Lazarus, and R. Platt, "Distributed health data networks: A practical and preferred approach to multi-institutional evaluations of comparative effectiveness, safety, and quality of care," *Med. Care*, vol. 48, no. 6, pp. S45–S51, Jun. 2010, doi: 10.1097/MLR.0b013e3181d9919f.

[10] K. Y. Yigzaw and J. G. Bellika, ''Evaluation of secure multi-party computation for reuse of distributed electronic health data,'' in *Proc. IEEE-EMBS Int. Conf. Biomed. Health Informat. (BHI)*, Jun. 2014, pp. 219–222, doi: 10.1109/BHI.2014.6864343.

[11] S. Purushotham, C. Meng, Z. Che, and Y. Liu, ''Benchmarking deep learning models on large healthcare datasets,'' *J. Biomed. Informat.*, vol. 83, pp. 112–134, Jul. 2018, doi: 10.1016/j.jbi.2018.04.007.

[12] G. H. Lee and S.-Y. Shin, ''Federated learning on clinical benchmark data: Performance assessment,'' *J. Med. Internet Res.*, vol. 22, no. 10, Oct. 2020, Art. no. e20891, doi: 10.2196/20891.

[13] X. Zhu, J. Wang, Z. Hong, T. Xia, and J. Xiao, ''Federated learning of unsegmented Chinese text recognition model,'' in *Proc. IEEE 31st Int. Conf. Tools with Artif. Intell. (ICTAI)*, Portland, OR, USA, Nov. 2019, pp. 1341–1345, doi: 10.1109/ICTAI.2019.00186.

[14] A. Ziller. (Dec. 2020). *Privacy-Preserving Medical Image Analysis*. [Online]. Available: https://hal.inria.fr/hal-03065933

[15] X. Wang and W. Cao, ''Non-iterative approaches in training feed-forward neural networks and their applications,'' *Soft Comput.*, vol. 22, no. 11, pp. 3473–3476, Jun. 2018, doi: 10.1007/s00500-018-3203-0.

[16] W. Cao, Z. Xie, J. Li, Z. Xu, Z. Ming, and X. Wang, ''Bidirectional stochastic configuration network for regression problems,'' *Neural Netw.*, vol. 140, pp. 237–246, Aug. 2021, doi: 10.1016/j.neunet.2021.03.016.

[17] W. Cao, X. Wang, Z. Ming, and J. Gao, ''A review on neural networks with random weights,'' *Neurocomputing*, vol. 275, pp. 278–287, Jan. 2018, doi: 10.1016/j.neucom.2017.08.040.

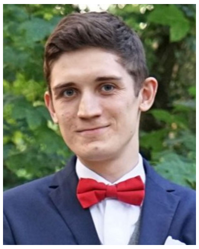
[18] Institute of Medicine (US) Roundtable on Evidence-Based Medicine. (2007). *The Learning Healthcare System: Workshop Summary. Washington (DC): National Academies Press (US)*. Accessed: Nov. 17, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK53494/

[19] (2011). *Institute of Medicine (US) and National Academy of Engineering (US) Roundtable on Value & Science-Driven Health Care, Engineering a Learning Healthcare System: A Look at the Future: Workshop Summary. Washington (DC): National Academies Press (US)*. Accessed: Nov. 18, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK61965/

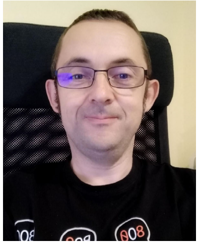
[20] A. Budrionis and J. G. Bellika, ''The learning healthcare system: Where are we now? A systematic review,'' *J. Biomed. Informat.*, vol. 64, pp. 87–92, Dec. 2016, doi: 10.1016/j.jbi.2016.09.018.

ANDRIUS BUDRIONIS received the B.S. and M.S. degrees in software engineering from Vilnius University, Lithuania, in 2008 and 2010, respectively, and the Ph.D. degree in computer science from the UiT—The Arctic University of Norway, Tromsø, Norway, in 2015. He is currently a Senior Research Fellow with the Norwegian Centre for E-health Research, University Hospital of North Norway, Tromsø. His interests include large scale e-health solutions, data-driven healthcare, and advanced data analytics methods.

MAGDA MIARA received the M.S. degree in computer science from Poznan University of Technology, Poznan, Poland, in 2020. During her studies, she did several internships in Amazon and Google. She is currently with the Faculty of Computing and Telecommunications, Poznan University of Technology. She was a member of the academic machine learning group called GHOST.



PIOTR MIARA received the M.S. degree in computer science from Poznan University of Technology, Poznan, Poland, in 2020. During his studies, he did several internships in Amazon and Google. He is currently with the Faculty of Computing and Telecommunications, Poznan University of Technology. He was a member of the academic machine learning group called GHOST.



SZYMON WILK received the M.Sc. and Ph.D. degrees in computer science from Poznan University of Technology, Poznan, Poland, in 1997 and 2003, respectively. He is currently an Associate Professor with the Faculty of Computing and Telecommunications, Poznan University of Technology, and a member of the Division of Intelligent Decision Support Systems. He was a Postdoctoral Research Fellow with the Mobile Emergency Triage, (MET), University of Ottawa, Canada, where he is also an Adjunct Professor. His research interests include clinical decision support systems, with a special focus on multi-agent architectures, machine learning techniques with symbolic representation of discovered knowledge, multi-criteria decision analysis and support, with a special focus on mitigating interactions in multiple clinical practice guidelines and on improving patient's adherence, and engagement in treatments. He is also involved in the CAncer Patients Better Life Experience (CAPABLE) project founded by the EU within the Horizon 2020 program.

JOHAN GUSTAV BELLIKA was born in Alta, Norway, in 1962. He received the M.Sc. and Ph.D. degrees in computer science from the UiT—The Arctic University of Norway, Tromsø, Norway, in 1997 and 2006, respectively.

He was with the Department of Community Medicine, UiT—The Arctic University of Norway, from 1992 to 1997. From 1997 to 2015, he was with the Norwegian Centre for Integrated Care and Telemedicine, University Hospital of North Norway, Tromsø. From 2007 to 2013, he was an Associate Professor with the Department of Computer Science, UiT—The Arctic University of Norway. During his employment at the department, he was a Researcher and a Lecturer for the M.Sc. program in telemedicine and eHealth. He is currently a Professor of medical informatics with the Department of Clinical Medicine, UiT—The Arctic University of Norway, and Norwegian Centre for E-health Research, University Hospital of North Norway. He has broad research experience at the intersection between medicine, medical research, and informatics. His current research interest includes methods for enabling privacy preserving reuse of health data.

Prof. Bellika is a Board Member of Norwegian Society for Medical Informatics, and a Norwegian Representative of the IMIA Working Group on Health Information Systems.