IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

Enhancing Central Model Performance: Leveraging Federated Learning Across Virtual Machine Networks for Distributed Training and Synchronization

Ronit Virwani[1] and Shubhangi Bhattacharya[2]
Student, Department of Information Technology[1]
Student, Department of Computer Science and Engineering[2]
School of Computing, MIT ADT University, Pune, India
ronitvirwani1@gmail.com and b.shubhangi108@gmail.com

Abstract: *This project takes a closer look at federated learning as a way of achieving superior machine learning models in a distributed manner while preserving privacy in the datasets that contribute. We have modelled a network of cooperating virtual machines working collectively without explicit sharing of data. Rather than distributing the complete big dataset to each system, we have split it into chunks of 10,000, 5,000, 40,000, 5,000 entries. These systems would then work on their data with learning rates of their model's making and in the decision-making processes to modify their settings, so that the data that systems would work on could allow for building their respective models by them. What this means is that the high point in the project is the combination of these models into one overarching model. The overarching model then gets better because of the small models learning from it without having to access the data associated with the models in a direct sense. This way, a better model can be built, which will intimately understand the data and thereby predict more accurately. Taken as a whole, we have shown how federated learning can improve the models of machine learning in a significantly private manner, and thus the methodology is positively postured with respect to future related work*

Keywords: Federated Learning, Privacy Preservation, Virtual Machine Networks, Data Partitioning, Machine Learning Algorithms, Hyperparameter Optimization, Decentralized Learning, Centralized Model Integration, Data Diversity, Model Performance Enhancement

I. INTRODUCTION

In the modern computing landscape, marked by data-driven pursuits, this relentless aspiration towards better machine-learning models is set parallel with a paramount concern around data privacy. This has been a critical challenge of leveraging precious insights from distributed data sources and refraining from compromising sensitive information within most industries. In such a context, Federated Learning (FL) arises as a revolutionary paradigm that proffers one of the most promising solutions to bridge the gap between the imperative for data-driven innovation and the necessity of preserving individual privacy rights. FL is characterized by the decentralized nature of its collaborative model training and hence aggregation of knowledge from several devices or entities, while ensuring that raw data stays locally stored and protected, hence does not exceed the privacy constraints dictated by several regulatory frameworks or organizational policies.

It is nowhere that the importance of FL comes into play more than in the domains where data silos, today often characterized by isolated datasets and strict privacy regulations, preclude smooth information sharing and analysis. In the area of healthcare, for instance, with the isolation of data islands within the various hospitals, the time is now ripe for collaborative learning approaches that would pool in insights from various databases without breach of patient confidentiality. In financial institutions, and governmental organizations, and data security, and the necessity to fully conform with strict regulations, FL could suggest one quite viable manner of using collective intelligence while keeping

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

the integrity and confidentiality of sensitive information. With the entry of regulations such as the General Data Protection Regulation (GDPR), it has further reinforced the need for methodologies that are robust and preserving of privacy, called to innovate, such as FL methods that enable effective analysis of data and knowledge extraction, and simultaneously respect individual privacy rights.

It has recently gained much attention owing to its development of different FL (Federated Learning) algorithms and systems which intend to support various machine learning models. Support for different models, including deep neural networks (NNs), gradient boosted decision trees (GBDTs), logistics regression, and support vector machines (SVMs) has been realized by integrating privacy-preserving mechanisms and decentralized learning frameworks aimed at the FL algorithms and systems to train them. The current availability of computing resources and the development of more sophisticated FL algorithms has resulted in the recent development of comprehensive FL systems and infrastructures which are anticipated to aid in the development and operationalization of FL methodologies over a variety of application domains. The recent development of these FL systems and infrastructures and the inevitable high-level control of virtual machine networks has motivated us to conduct a study in this research paper. Specifically, we conduct a comprehensive performance analysis of how FL is actually deployed, across virtual machine networks are used to implement (for example), distributed training and, more generally, synchronization, to enhance the performance of some centralized machine learning model. Our intention is to provide in this research paper a detailed, lower-layer inspection which exposes the "guts" and underlying mechanisms of all the FL methodologies which would ultimately need to be successfully deployed, in order to demonstrate FL is, indeed, ready to be attached to the iconic rocket ship of today's most transformative advancements of collaborative machine learning.

II. OVERVIEW OF FEDERATED LEARNING

FL is a secure distributed learning framework in which a virtual model can be constructed to address the matter of dispersive clients collaborating needless to expose raw information [21]. The virtual model is an optimal global model for aggregating data from all participants and each participant serves the local objective using the obtained model. FL can achieve that the results of this modeling be much of similar to the traditional centralized training model [20], in which data from multiple clients are brought together in a same center server for modeling. In a federated mechanism, it is generally assumed that participants have the same identity and support the establishment of shared data policies. Because the data is not directly transferred, it does not affect data specifications or compromises user privacy.

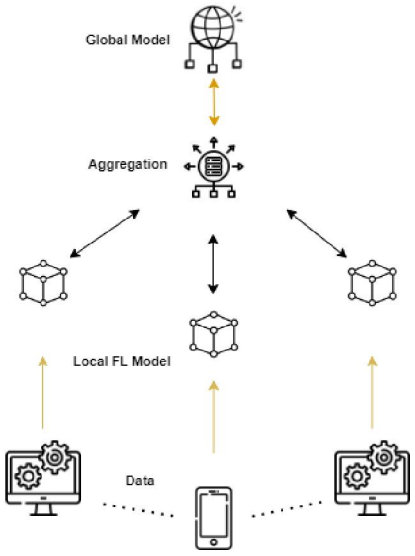

Fig. 1 The basic framework of FL

First, we define the basic concept of FL: Define N participants in FL, all of whom want to merge their data $\{P_1, P_2, \cdots, P_N\}$ to train a global model. A frequently used approach is to gather all the data together and use total data $P = P_1 \cup P_2 \cup \cdots \cup P_N$ to train a model M_{sum} with a performance of V_{sum} . FL is a learning framework where participants co-training a

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

common model $M_{fed}$with a performance of V_{fed}, in which no participant exposes its private data to others. Let ε be non-negative, then the performance loss of FL model is expressed as:$|V_{fed} - V_{sum}| < \varepsilon$
FL allows a small range of performance deviation between the trained global federated model and the centralized training model. After several rounds of efficient federated global training, the performance of global model will continue to enhance and the convergence approaches to the performance of the centralized model..

III. LITERATURE SURVEY

In this section, we have discussed papers related to distributed machine learning, federated learning, and privacy - preserving machine learning. One of the most important parts in research is reading different existing literature's implementation, advantages, and disadvantages. We observed that federated learning is a recent topic which has few existing literature's and articles. A lot of work has already been done in privacy-preserving learning and distributed machine learning algorithms. In federated learning, a star network [where a central server is connected to a network of devices, e.g., in Figure 3(a)] is the predominant communication topology; we therefore focus on the star network setting in this article. We briefly discuss decentralized topologies [where devices only communicate with their neighbors, e.g., in Figure 3(b)] as a potential alternative. In data center environments, de- centralized training has been demonstrated to be faster than centralized training when operating on networks with low bandwidth or high latency. (Tian Li . et al)

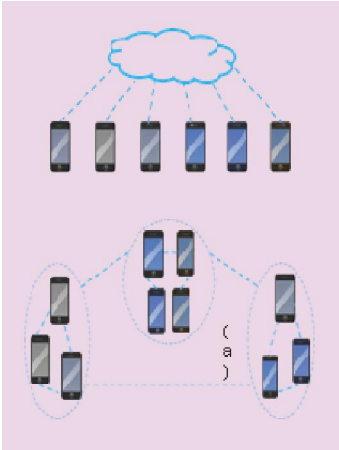

Fig. 2 Centralized versus decentralized topologies. In the typical federated learning setting and as a focus of this article, we assume (a) a star network where a server connects with all the remote devices. (b) Decentralized topologies are a potential alternative when communication to the server becomes a bottleneck.

The sequence diagram of the fog-based Federated Learning (FL) framework shows the increasing interest in decentralized and fog computing strategies for boosting the effectiveness and adaptability of FL systems. Researchers have stressed that such fog-based architectures could deal with connection latencies and bandwidth restrictions in distributed learning settings. They argued for the integration of fog computing with FL, and pointed out that fog nodes might prime local model updates and aggregations to minimize communication overhead and to better preserve privacy. Likewise, a detailed treatment of fog-based FL frameworks, their collaborative model training processes, and how optimal fog nodes might be picked to perform efficient global model aggregations while ensuring data privacy and security was provided. In sum, fog-based FL frameworks show potential to enable robust and efficient collaboration among learning distributed across edge computing environments, and in so that they may enable interesting advances in decentralized machine learning.

In 2019, Bonawitz et al. [3] discussed the massive potential of federated learning for training large decentralized datasets, presenting the core framework, addressing fundamental challenges and lessons learned, and discussing future directions for future distributed systems learning. Smith et al. [4] researched multi-task federated learning systems in 2017 to enable collaborative learning of a shared model across clients, where each has one task, without centralizing data, and whenever clients only have access to correlated samples. They designed a system, which (i) aggregates model updates via a novel concentrator, which uses random rotations to change the axes of different clients' updates so they

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

mostly add along a shared axis, (ii) secures aggregation against eavesdropping by a curious central server on a small fraction of user data, and (iii) uses adaptive moment estimation (Adam) to ensure efficient training.

In 2016, Papernot et al. [5] introduced attacks and defenses, where the position is that the deep learning model is not just the weights and architecture, but an entire pipeline, which includes the model, the training procedure, and the training and test data. They introduced attacks on these different levels and demonstrated how to defend against them. Building on this work, Bonawitz et al. [6] detailed a secure protocol for secure aggregation of highly-accurate models, which are generated and aggregated across devices using federated learning, using a single new cryptographic primitive, private information retrieval with low communication overhead. Bost et al. [7] paper in 2014 warned of the pitfalls of securing learning, where they introduced privacy-preserving classification protocols for in the two-party and multi-party settings. Kamarinou et al. [8] today queried: "When data meets model: the European general data protection regulation and the challenges to the right of data portability" in a machine learning world, including looking at individual profiling in the context of the General Data Protection Regulation.

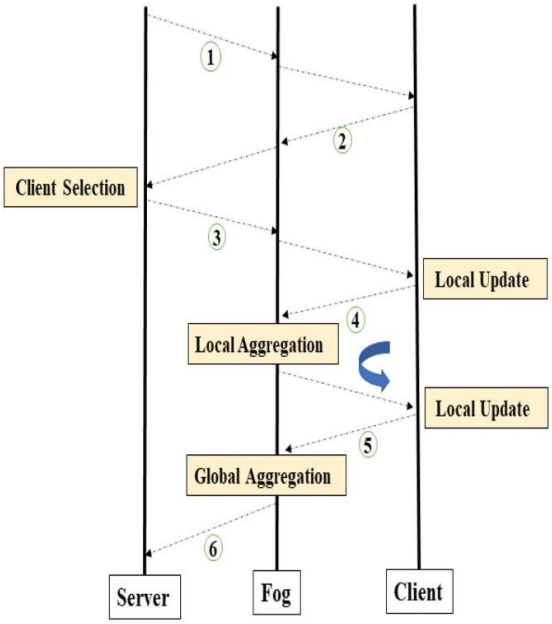

Fig. 3. Different steps of fog-based FL framework.

IV. METHODOLOGY

In this section, we present a comparative study between three distinct classifiers: the basic machine learning classifier, distributed machine learning classifier, and federated learning classifier. We have trained these classifiers using the Fashion-MNIST dataset within the Keras and TensorFlow frameworks. The Fashion-MNIST dataset serves as a drop-in replacement for the traditional MNIST dataset and comprises 60,000 training images and 10,000 testing images, each with dimensions of 28x28 pixels. These images are in grayscale and represent different clothing and garment items. They are labelled from 1 to 10, indicating various item categories. The Fashion-MNIST dataset thus presents a larger, more challenging, and diverse dataset, which is appropriate for evaluating the performance of the classifiers that were employed.

4.1 Basic Machine Learning classifier

We present the detailed implementation of the basic machine learning classifier in this section using a Neural Network architecture. The model implementation methodology comprises four main stages: (1) Data Pre-processing, (2) Neural Network Architecture Definition, (3) Model Training, and (4) Model Testing as illustrated in Figure 4.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

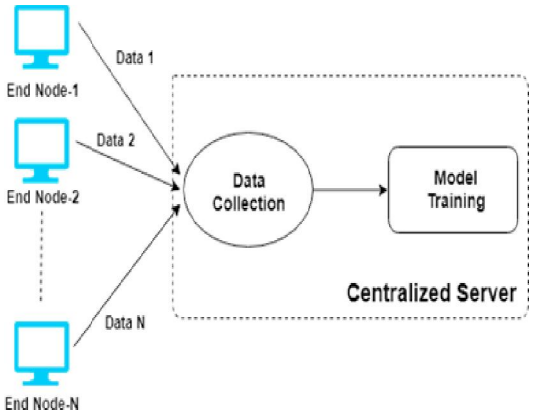

Figure 4: Working of the Basic Classifier

To start, we import the necessary libraries that are required to carry out the data processing and model training, including:

TensorFlow and Keras, two open-source libraries that facilitate the creation and training of neural network models to process and classify image data at scale.

Matplotlib, a plotting library that is utilized to generate a visual representation of the data. Numpy library used for working with arrays and matrices of multidimensional data. Pandas library that provides data structures and data analysis tools. Time library that provides various time-related functions used for motion-details updates including pausing or stopping the time n seconds.

We load the Fashion-MNIST dataset with four datasets i.e., train image and test image, train label, and test label. We normalize the data - we scale the data so that it lies in the range 0 to 1. Then, we create a training and testing set from Fashion MNIST! It will now be easier for our model to learn and generalize as we have used two separated sets each for training and testing.

After the data has been processed, we need to define the architecture of our neural network model. Here it contains three layers.The first layer is the flatten layer in which the input image matrix is converted to a vector. The second layer is the dense layer with rectified linear unit (ReLU) function that is responsible for effectively transforming our input data to suitable features. It will consist of 128 neurons. The last layer is the dense layer with softmax function which will classify the input into the specific class. It has 10 nodes/neurons because of 10 different classes of input in the fashion mnist dataset.Up till now, we have defined the Neural Network architecture. Now, we have to train the model.This is done by training the model and then testing the model to see how it performs on the given dataset. Here, first, we start a TensorFlow interactive session followed by an example use of tf.train. Monitored Training Session. It takes care of a lot of bookkeeping that is required to execute a TensorFlow computation. During training the Classification model, We configure a Monitored Training Session and tf.estimator.train_and_evaluate. We loop over train data until limiting cases. At the end, test set accuracy and loss will be assessed.

4.2 Distributed Machine Learning classifier

In order to address the challenges of large datasets that are bigger than the available memory in traditional machine learning setups, the trend has shifted towards distributed machine learning algorithms that can handle large datasets by leveraging the powers of many devices. This is the case for our distributed machine learning classifier, where we have transformed the basic classifier implementation we had before to work in a distributed manner, simulating many devices on different ports of the same computer. In a real-world implementation, of course, the devices would be spread out across the world, each with their own IP address, but we simplified the setup using different localhost ports to run in parallel on a single computer.

We had three devices running concurrently: the first device acted as a parameter server to keep the global variable states. The other two devices acted as workers which ran the training operations and updated the weights. We were able to give different roles to the parameter server and worker by assigning them each a job-name and task-index. We

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

formed a cluster with one of each of the three devices and described the communication between the nodes with the tf.train. ClusterSpec class. The parameter server listened to volumes at all times and the workers computed gradients and updated a shared model on the parameter server after an initial data partition and normalization. The chief worker was responsible for the synchronization of the gradients between the replicas and responsible for updating a model with the gradients and into adding the newest computed gradients while maintaining a cohesive distribution on the worker. The graph was defined and the model was trained with the knowledge that it ran over multiple devices. We added regular interval checkpoints after ourselves to ensure the state of the model outlived the lifecycle of the session. We continued to evaluate our model as we did before.

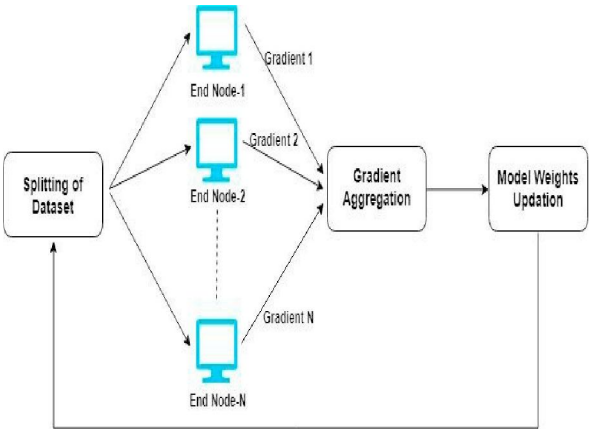

Fig. 4.1 Distributed Classifier

4.3 Federated Machine Learning Classifier

We have described the federated learning classifier, in which each worker updates its local weights independently. The workers transmit their weights, not gradients, to the parameter server at fixed step intervals without direct communication of gradients. The chief worker calculates the average of the transmitted weights, updates the shared model on the parameter server, and ensures each worker's weights are synchronized. The complete workflow of the federated machine learning classifier is depicted below in Figure 4.2.

Our initial implementation of the federated learning classifier is quite similar to the distributed machine learning classifier. There are two key differences. We added the federated-average-optimizer to handle the optimization of the federated averaging process. The other addition is the interval-steps variable, which regulates the frequency with which the workers will be updating their weights — i.e., after interval-steps.

As in distributed learning, we only use the replica-device-setter to pass it to our custom optimizer. In this case, we pass a custom optimizer whose only iteration runs the federated average logic. We specify it as one of the optimizer's arguments. The placement of each trainable variable is then taken care of by the custom optimizer, which places them in our parameter server where their collective average from all the local models is stored. The rest of the training process is managed by MonitoredTrainingSession, which each worker initializes independently so that each one does its own training without needing a designated chief worker.

We finish the training phase by evaluating the accuracy of the neural network model and its capacity to generalize to the training dataset. Thus we show the federated learning viable and robust for an illustrative programming language model.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

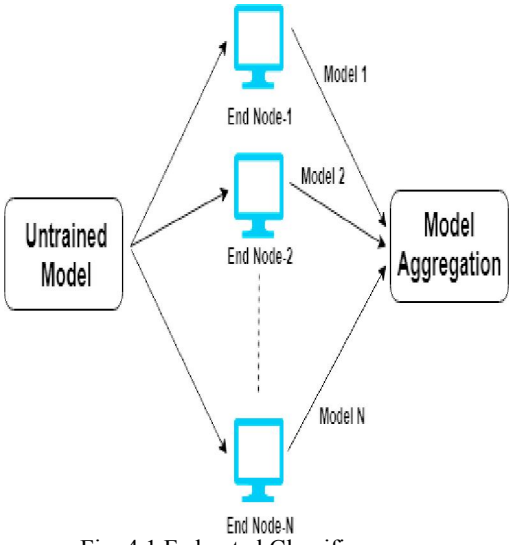

Fig. 4.1 Federated Classifier

Title must be in 24 pt Regular font. Author name must be in 11 pt Regular font. Author affiliation must be in 10 pt Italic. Email address must be in 9 pt Courier Regular font.

V. EXPERIMENTAL RESULTS

In this experiment, we tested three classifiers — a basic machine learning classifier, a distributed machine learning classifier, and a federated machine learning classifier — on the Fashion MNIST dataset. Each classifier was trained over 5 epochs with a batch size of 32 images. We used categorical cross-entropy to evaluate the loss after each epoch, which consistently decreased every epoch.

The basic machine learning classifier was trained on a single node without any data distribution. It took approximately 26.4 seconds to train, achieving a fair test accuracy of 87%. Without communication overhead, a single node's limited computational bounds held it back from being faster.

For the distributed machine learning classifier, three nodes were employed: a parameter server and two worker nodes. The training time for each worker node was approximately 14.11 seconds, resulting in a slight decay of accuracy to 86.23%. The increased overhead of communication — more specifically, the noise of gradients adding to the parameter server for averaging — was the cause of decreased accuracy; over one node, communication was virtually non-existent.

On the other hand, the federated learning classifier took a different tack, generating two local models, one on each worker node individually — effectively having each worker node act as if it were the only system performing model training — and then transmitting the weights back to the parameter server for aggregation after a defined interval. This earned the federated learning classifier an additional 2 seconds of setup time, but its more streamlined training time of 16.75 seconds was just enough to put the system over the edge, albeit with a slightly reduced accuracy of 85.15 percent. It was the extra manoeuvre in the federated learning classifier — most notably, the fellow application of weights to the local models via averaging — that led to the slightly longer training duration.

With this in mind, it's clear that the federated learning classifier, which produced and aggregated two local models in the end, showed a marked balance between training speed and accuracy. Certainly, it served as a suitable intermediary between both the basic and distributed machine learning classifiers and would be a viable alternative to each.

VI. DISCUSSION

This study looked into Federated Learning, a way to do machine learning while keeping data private and spread out over many places. It compared how well it works against other common ways of doing machine learning, focusing on how fast it learns and how accurate it is.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

The research found that the usual single-location machine learning method was pretty accurate but slow because it could only do so much at once. On the other hand, when machine learning was done over many locations working together, it got faster but wasn't quite as accurate because it was hard for all the parts to communicate efficiently.

Federated Learning was a middle ground, balancing speed and accuracy. It worked by spreading out the learning process but in a way that reduced the problems of communication between parts. Even though it wasn't the top performer in pure accuracy or speed, it struck a good balance, making it a solid choice for situations where keeping data private and working efficiently across different locations is important.

VII. CONCLUSION

In conclusion, the study focussed into the use of Federated Learning as a privacy-preserving and efficient approach. The comparison of basic, distributed, and federated learning classifiers revealed useful information on the trade-offs between training efficiency and model correctness in the setting of various datasets. The findings highlighted federated learning's potential as a feasible option for combining the imperatives of data privacy and computational efficacy in modern machine learning systems.

The research not only clarified the performance features of each strategy but also underlined the importance of distributed collaboration in boosting the overall efficacy of machine learning models through its full exploration of the decentralized learning paradigm. The findings emphasized the need of establishing a compromise between model accuracy and training efficiency, and they advocated for the use of federated learning as a possible method for reaching this balance.

VIII. FUTURE SCOPE

In the future, the research opens up new options for further investigation and improvement in the field of Federated Learning. Future research could look into increased privacy-preserving techniques, such as differential privacy and secure multi-party computation, to strengthen the security of federated learning systems. Furthermore, adapting federated learning approaches to accommodate heterogeneous data sources and unique learning architectures can provide useful insights into their applicability across multiple domains and businesses. Furthermore, the development of more sophisticated aggregation approaches, such as hierarchical aggregation and adaptive learning rate scheduling, has the potential to dramatically improve the efficiency and speed of convergence of federated learning models.

Integration of federated learning with new technologies such as blockchain and edge computing is another exciting research route, potentially enabling the creation of decentralized, safe, and efficient machine learning ecosystems. Future research endeavors can help to the development and widespread adoption of Federated Learning by pursuing these pathways, hence supporting its incorporation into varied real-world applications and use cases.

REFERENCES

[1]. Kunal Chandiramani, Dhruv Garg, N Maheswari,Performance Analysis of Distributed and Federated Learning Models on Private Data,Procedia Computer Science,Volume 165,2019,Pages 349-355,ISSN 1877-0509.J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[2]. Bonawitz, Keith. "Towards Federated Learning at Scale: System Design." arXiv preprint arXiv:1902.01046 (2019).

[3]. Smith, Virginia, et al. “Federated-multi-task learning.” Advances in Neural Information Processing Systems. 2017.

[4]. Papernot, Nicolas, et al. “Towards the science of security and privacy in machine learning.” arXiv preprint arXiv:1611.03814 (2016).

[5]. Bost, Raphael, et al. “Machine learning classification over encrypted data.” NDSS. Vol. 4324. 2015.

[6]. T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, May 2020.

[7]. A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, Protection against reconstruction and its applications in private federated learning. 2018. [Online]. Available: arXiv:1812.00984

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

[8]. S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi and M. Guizani, "A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5476-5497, 1 April1, 2021, doi: 10.1109/JIOT.2020.3030072.

[9]. Kairouz, Peter, et al. "Advances and open problems in federated learning." arXiv preprint arXiv:1912.04977 (2019).

[10]. Gupta, R., Alam, T. Survey on Federated-Learning Approaches in Distributed Environment. Wireless Pers Commun 125, 1631–1652 (2022).

[11]. Federated Learning for Edge Computing: A Survey. Alexander Brecko, Erik Kajati, Jiri Koziorek and Iveta Zolotova. Journal: Applied Sciences, 2022, Volume 12, Number 18, Page 912

[12]. T. Huang, W. Lin, W. Wu, L. He, K. Li and A. Y. Zomaya, "An Efficiency-Boosting Client Selection Scheme for Federated Learning With Fairness Guarantee," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1552-1564, 1 July 2021.

[13]. C. Che, X. Li, C. Chen, X. He and Z. Zheng, "A Decentralized Federated Learning Framework via Committee Mechanism With Convergence Guarantee," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 12, pp. 4783-4800, 1 Dec. 2022.

[14]. S. Wang et al., "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1205-1221, June 2019.

[15]. FLRA: A Reference Architecture for Federated Learning Systems. Software Architecture, 2021, Volume 12857. ISBN : 978-3-030-86043-1. Sin Kit Lo, Qinghua Lu, Hye-Young Paik

[16]. R. Song et al., "Federated Learning via Decentralized Dataset Distillation in Resource-Constrained Edge Environments," 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-10.

[17]. R. Myrzashova, S. H. Alsamhi, A. V. Shvetsov, A. Hawbani and X. Wei, "Blockchain Meets Federated Learning in Healthcare: A Systematic Review With Challenges and Opportunities," in IEEE Internet of Things Journal, vol. 10, no. 16, pp. 14418-14437, 15 Aug.15, 2023.

[18]. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. Nuria Rodríguez-Barroso, Daniel Jiménez-López, M. Victoria Luzón, Francisco Herrera, Eugenio Martínez-Cámara