ABSTRACT

Federated Learning is a machine learning environment in which the aim is to train a high-quality centralized model using training data spread among multiple clients, each of which has unreliable and generally sluggish network connections. For this situation, we investigate learning methods in which each client individually computes an update to the current model based on its local data and sends this update to a central server, where the client-side changes are combined to compute a new global model. Mobile phones are the most common clients in this scenario; therefore, communication efficiency is critical.

Secure aggregation for federated learning enhances federated learning by guaranteeing that weights received from distant devices are encrypted before being forwarded to the central device to be aggregated or before being aggregated in another remote device. This adds an extra degree of security by guaranteeing that the weights stay anonymous, as the weights of a model might potentially jeopardize privacy.

1. INTRODUCTION

1.1 SCOPE

Federated learning is a method of training deep learning models where the training takes place across various remote devices before aggregating the results into a central model. This is done to protect the privacy of the data on the remote devices, as the central model does not have access to the personal data on the remote devices but only to the weights learned from the remote models.

To improve the security of this process, the weights learned from the remote devices are encrypted before being sent to the central model for aggregation. The encrypted weights are then decrypted after they have been aggregated.

In this project, remote devices are not used, but the process of training across remote models is simulated by dividing a dataset into smaller sub-datasets and creating separate models for each sub-dataset. A central model is created, but it has no access to the data. Instead, it only has access to the weights obtained from the training of the remote models in an anonymous manner.

Virtual workers are used to simulate independent, anonymous devices where the remote models are stored. The remote models are trained on their respective sub datasets, and the weights are returned to the central model for aggregation.

The weights from the remote models are encrypted using the Python Syft package and are then divided and aggregated among selected remote models in an encrypted form. Finally, they are returned to the central model to be decrypted.

This process ensures that the central model does not have direct access to the weights from the remote models but only to their aggregated, encrypted form. This adds an extra layer of privacy and protects the data on the remote devices from being compromised by the central model.

1.2 NOVELTY

The novelty of this project lies in the use of virtual workers to simulate remote devices for federated learning, where the training takes place across various independent and anonymous models on

sub-datasets. This simulates a distributed learning environment, where data privacy is maintained by encrypting the weights learned from the remote models before aggregating them into the central model. The use of the Python Syft package for encryption and decryption of the weights adds an extra layer of privacy and security to the federated learning process. Overall, this project explores the approach to maintaining the privacy and security of the training process.

1.3 LITERATURE SYRVEY

Title	Major Technologies Used	Results/ Outcomes	Major drawbacks
Gradient-based learning applied to document recognition.	Convolutional Neural Networks (CNNs), backpropagation algorithm, digitized images of documents for training and testing	The paper introduced a new approach for document recognition using CNNs, which achieved a significant improvement over the previous state-of-the-art methods.	The dataset used was relatively small compared to modern standards, and the paper did not address some of the computational challenges that come with training large CNNs on large datasets.
Biscotti: A Blockchain System for Private and Secure Federated Learning.	Blockchain, smart contracts, secure multi-party computation (MPC), federated learning	The paper introduced a new system for federated learning that leverages blockchain and smart contracts to ensure privacy, security, and fairness in a distributed environment. Experimental results showed that the proposed system outperformed existing federated learning methods in terms of privacy and security.	The proposed system involves a high computational overhead, which could limit its scalability and practicality in real-world scenarios.
A Hybrid Approach to Privacy-Preserving Federated Learning.	Homomorphic encryption, differential privacy, federated learning,	The paper proposed a new hybrid approach to federated learning that combines the benefits of homomorphic encryption and	The proposed method involves a significant computational overhead, which could limit its

	centralized learning	differential privacy to protect the privacy of user data while achieving high accuracy in the learning process. The proposed method achieved higher accuracy than existing methods with comparable levels of privacy protection.	scalability and practicality in real-world scenarios.
From distributed machine learning to federated learning: In the view of data privacy and security.	Federated learning, differential privacy, secure aggregation, decentralized optimization	The paper reviewed the evolution of distributed machine learning to federated learning, with a focus on the importance of data privacy and security. The authors highlighted the advantages and challenges of federated learning and proposed some solutions for ensuring privacy and security in the federated learning process.	The paper did not provide any empirical results or implementation details of the proposed solutions.
Federated Learning: Challenges, Methods, and Future Directions.	Federated learning, privacy-preserving techniques, distributed optimization	he paper provided an overview of federated learning, its challenges, and existing solutions for privacy-preserving federated learning.	
A review of applications in federated learning.	reviewed various applications of federated learning in industrial engineering. They highlighted the major	The review showed that federated learning has promising results in improving accuracy and privacy of machine learning models.	One drawback of federated learning is the potential for communication and synchronization overhead, which can limit the scalability of the system

	technologies used in federated learning, including differential privacy, homomorphic encryption, and secure multi-party computation.		
Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications.	conducted a survey on the enabling technologies, protocols, and applications of federated learning. They identified the major technologies used in federated learning, including differential privacy, secure aggregation, and secure multi-party computation.	The survey showed that federated learning has promising outcomes in terms of accuracy and privacy preservation.	One drawback of federated learning is the potential for malicious attacks from participating devices, which can compromise the integrity and privacy of the system.
A survey on security and privacy of federated learning.	They highlighted the major technologies used in federated learning, including differential privacy,	The survey showed that federated learning has promising outcomes in terms of privacy preservation, but security concerns remain a major challenge.	One drawback of federated learning is the lack of standardized security protocols and frameworks, which can make it difficult to ensure the security of the system.

	homomorphic encryption, and secure multi-party computation.		
The future of digital health with federated learning.	They highlighted the major technologies used in federated learning, including differential privacy, federated learning algorithms, and secure multi-party computation.	The review showed that federated learning has promising outcomes in terms of improving the accuracy and privacy of digital health applications.	One drawback of federated learning is the potential for communication overhead and privacy breaches, which can limit the scalability and effectiveness of the system
Efficient and Privacy-enhanced Federated Learning for Industrial Artificial Intelligence.	proposed an efficient and privacy-enhanced federated learning framework for industrial artificial intelligence. They used differential privacy and secure aggregation to improve privacy preservation in federated learning.	The results showed that their proposed framework can achieve high accuracy while preserving privacy.	One drawback of their framework is the potential for communication and computation overhead, which can limit the scalability of the system.
Security Data Collection and Data Analytics	They highlighted the major	The survey showed that secure data collection and analytics are critical	One drawback of federated learning is the potential for

in the Internet: A Survey.	technologies used in security data collection, including encryption, access control, and secure communication protocols.	for ensuring the security and privacy of federated learning systems.	security breaches from participating devices, which can compromise the integrity of the system.
Global Data Plane: A Federated Vision for Secure Data in Edge Computing.	proposed a federated vision for secure data in edge computing. They used a global data plane architecture to enable secure and efficient data sharing among edge devices.	The results showed that their proposed architecture can improve data privacy and availability in edge computing.	One drawback of their approach is the potential for communication overhead and synchronization issues, which can limit the scalability of the system.
Byzantine-Resilient Secure Federated Learning.	proposed a Byzantine-resilient secure federated learning framework to address the security challenges of federated learning. They used secret sharing and threshold cryptography to protect against malicious attacks from participating devices.	The results showed that their proposed framework can improve the security and privacy of federated learning systems.	One drawback of their approach is the potential for increased communication overhead and computation complexity, which can limit the scalability of the system.

A Survey of Incentive Mechanism Design for Federated Learning.	The major technologies used in this study are game theory, optimization, and machine learning.	The results of the study show that incentive mechanisms have significant impacts on the performance of federated learning.	One of the drawbacks of the paper is that it only focuses on incentive mechanisms and does not discuss other important aspects of federated learning.
Federated learning for drone authentication.	The major technologies used in this study are machine learning, blockchain, and cryptography.	The results of the study show that the proposed mechanism can achieve high accuracy and security for drone authentication.	one of the drawbacks of the paper is that the proposed mechanism requires a significant amount of computational resources, which may not be available on low-power devices.
A Performance Evaluation of Federated Learning Algorithms.	The major technologies used in this study are machine learning, optimization, and distributed computing.	The results of the study show that federated learning can achieve comparable performance to centralized learning under certain conditions, such as when the data is distributed evenly.	one of the drawbacks of the paper is that it only considers a limited set of federated learning algorithms and does not discuss their applicability in real-world scenarios.
Survey of Personalization Techniques for Federated Learning.	This paper surveys the personalization techniques for federated learning. The	The results of the study show that personalization can improve the performance of federated learning by	one of the drawbacks of the paper is that it only considers a limited set of personalization

	major technologies used in this study are machine learning, optimization, and privacy-preserving techniques.	leveraging user-specific data.	techniques and does not discuss their scalability and robustness in real-world scenarios.
TiFL: A Tier-based Federated Learning System.	This paper proposes a tier-based federated learning system called TiFL. The major technologies used in this study are distributed computing, machine learning, and optimization.	The results of the study show that TiFL can achieve high performance and scalability compared to other federated learning systems.	one of the drawbacks of the paper is that the proposed system requires a high degree of coordination and communication among the tiers, which may not be feasible in some scenarios.
Security and Privacy Threats to Federated Learning: Issues, Methods, and Challenges.	This paper discusses the security and privacy threats to federated learning. The major technologies used in this study are machine learning, cryptography, and security analysis.	The results of the study show that federated learning is vulnerable to various attacks, such as model poisoning and membership inference attacks. The paper also discusses some potential solutions to mitigate these threats.	one of the drawbacks of the paper is that it only discusses the threats and solutions at a high level and does not provide a comprehensive analysis of their effectiveness.
Privacy and Security in	Machine learning	Many of these papers propose new techniques	Some papers are highly technical

Federated Learning: A Survey.	Blockchain Privacy-preserving techniques (such as differential privacy) Distributed computing Encryption and secure computation Internet of Things (IoT)	or systems for improving federated learning in various aspects such as privacy, security, performance, and scalability. Some papers provide surveys or overviews of the state-of-the-art research in federated learning and related areas.	and require a strong background in machine learning, cryptography, or other specialized fields to fully understand. Some papers propose solutions that may not be practical to implement in real-world settings due to technical limitations, cost, or other constraints.
Federated Machine Learning: From a Software Engineering Perspective	Software engineering, federated machine learning	This paper provides a comprehensive overview of the software engineering aspects of federated machine learning. It explores the software engineering practices and challenges in developing federated machine learning systems, including system architecture, communication, security, and privacy.	This paper does not provide any empirical evaluation or case study.
A systematic review of federated learning applications for biomedical data	Biomedical data, federated learning	This paper presents a systematic review of the applications of federated learning in biomedical data. The authors analyzed 51 studies and found that federated learning has shown promising results in various biomedical	The studies analyzed in this paper used different federated learning approaches and evaluation metrics, making it difficult to compare and

		applications, such as medical image analysis, electronic health records, and genomics.	generalize the results. Additionally, the paper does not provide a meta-analysis or quantitative synthesis of the studies.

All the 30 papers which are based off federated learning have been reviewed thoroughly and are used for this project for comparison and better understanding.

On analysing, we conclude that federated learning has several limitations that can impact its effectiveness and efficiency.

Firstly, the approach heavily relies on the quality and quantity of data collected from devices, which can vary significantly in terms of data distribution and quality. Secondly, the communication and computation costs of federated learning can be high, especially when dealing with large datasets or many devices. This can lead to slow convergence and increased power consumption. Additionally, federated learning may suffer from issues related to privacy and security, as data is transmitted and processed on distributed devices, which increases the risk of data breaches or malicious attacks. Finally, federated learning may not be suitable for all applications, as some tasks may require centralized data processing or may not benefit from distributed learning.

2. ARCHITECTURE

Federated Learning

Main Principle: Train Locally – Average Globally

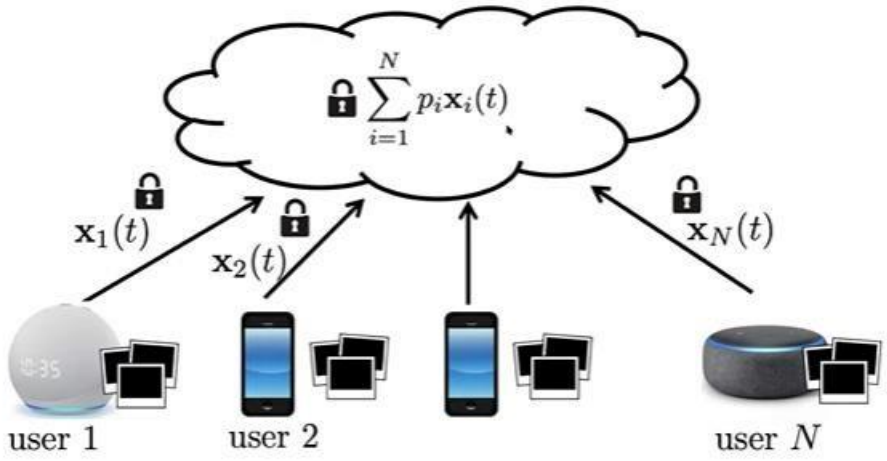

Fig 1: Federated Learning

Foundations

- Model Aggregation
- Data Heterogeneity
- No/Weak Labels – Unsupervised FL

Scalability

- Resource Constrained FL (Small edge models, large server model)
- Convergence: 1K users to 1M Users
- Federated neural architecture search.

Trustworthiness

- Secure and resilient model aggregation
- Adversarial users (data/model poisoning)
- Leveraging trusted computing environments

Ensuring Privacy by avoiding Data Movement from the users

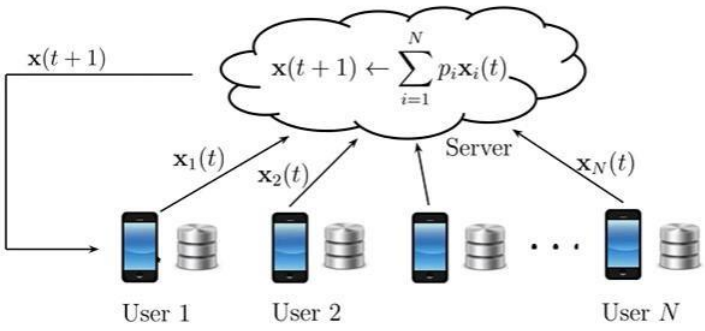

Fig 2: Ensuring Privacy by avoiding Data Movement from the users

Secure aggregation is a **multi-round secure MPC problem with user dropouts.**

Partial user participation leads to **privacy leakage.**

- Random selection may reveal all individual models.
- Exp o N=40 users o MNIST dataset with non iid distribution o K=8 users are selected at random at each round o The server estimates the individual gradients through least-squares.

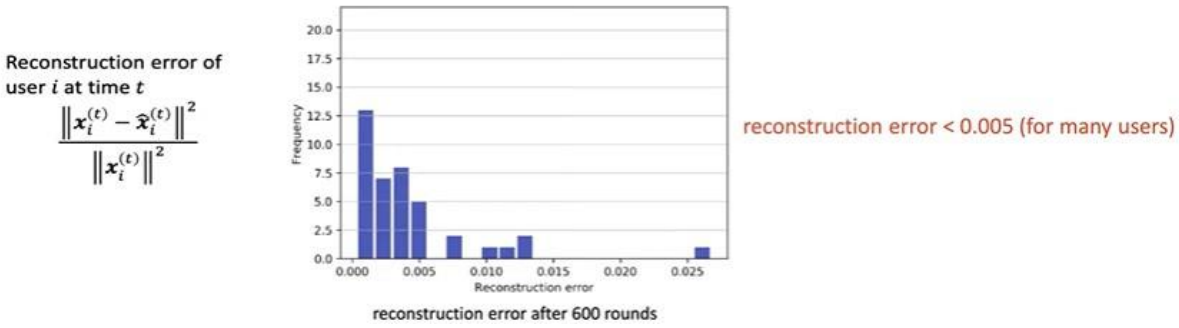

Fig 3: Result after random selection

Federated averaging with partial user participation

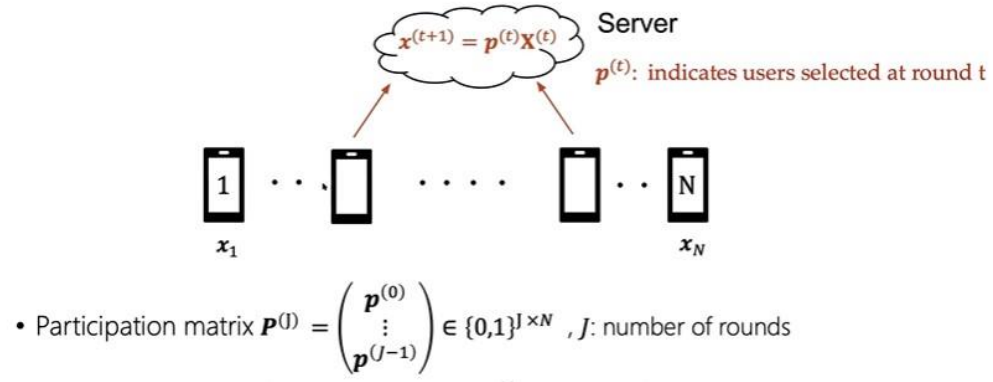

Fig 4: Federated averaging

Method 1: Multi-round Privacy (T)

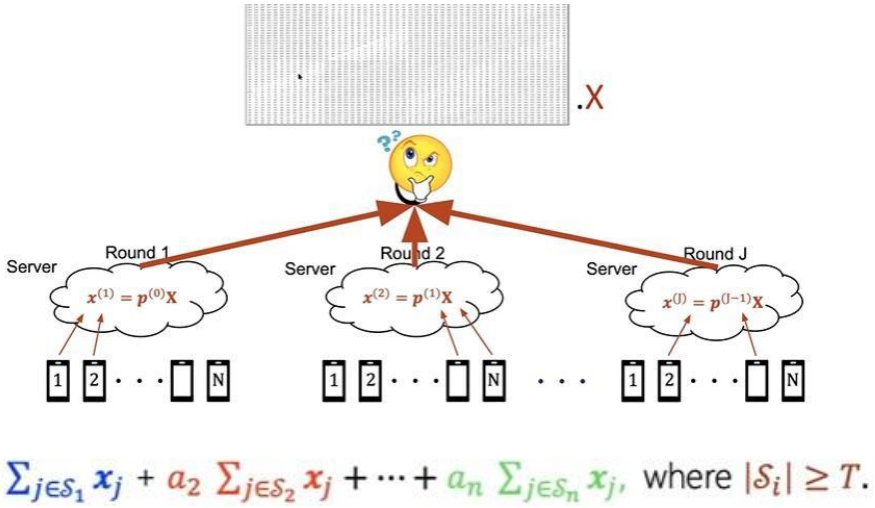

Fig 5: Multi-Round privacy

Baseline

1. User partitioning
   - Large multi-round privacy ***T=group size***
   - In many rounds, however, no groups are available.
2. Random Selection
   - Small multi-round privacy ***T=1***
   - Any subset of available users can be selected in any round.

Method 2: Average Aggregation Cardinality (C)

C = Average number of participating users over all rounds

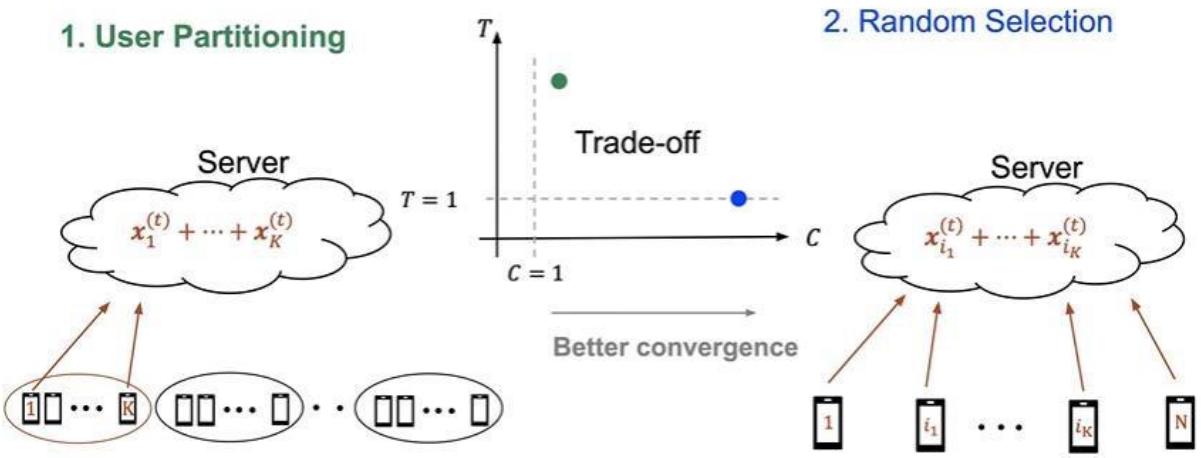

Fig 6: Average aggregation cardinality

Method 3: Aggregation Fairness Gap (F)

- Aggregation Fairness Gap F
- F = max. average participation frequency – min. average participation frequency

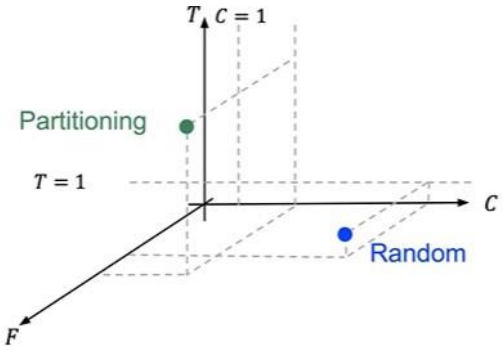

Fig 7: Aggregation fairness gap

Proposed Approach: Multi Round Sec Agg

1. **Batch Partitioning**
   - **Idea:** Partition users into T-user batches; allow selection of any K/T available batches
   - **Input**: N, K<= N, 1<= T <= K

- **Output**: A family of K User sets satisfying the multi-round privacy T ▪ This Family is represented by a matrix B.

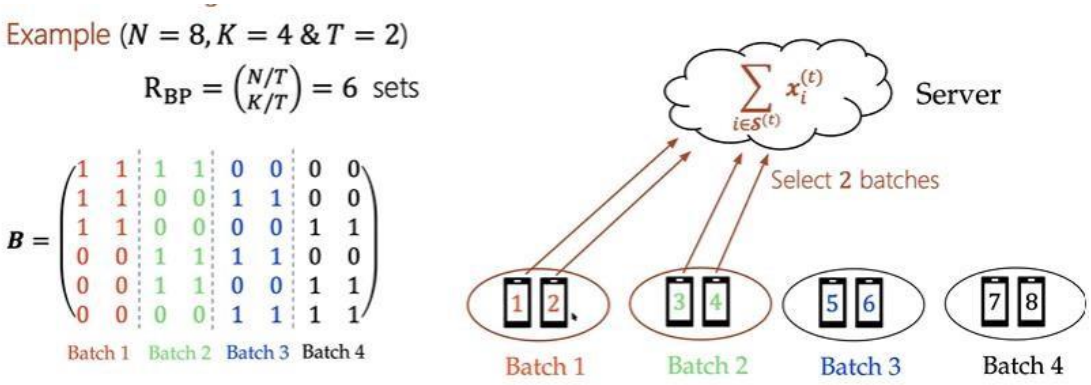

Fig 8: User sets converted to binary

2. **Available batch selection to guarantee fairness.**
   - **Idea:** Select based in the minimum frequency of participation
   - **Input:** Set of available users at round t and B
     ■ **Output:** Set of users that will participate at round t.

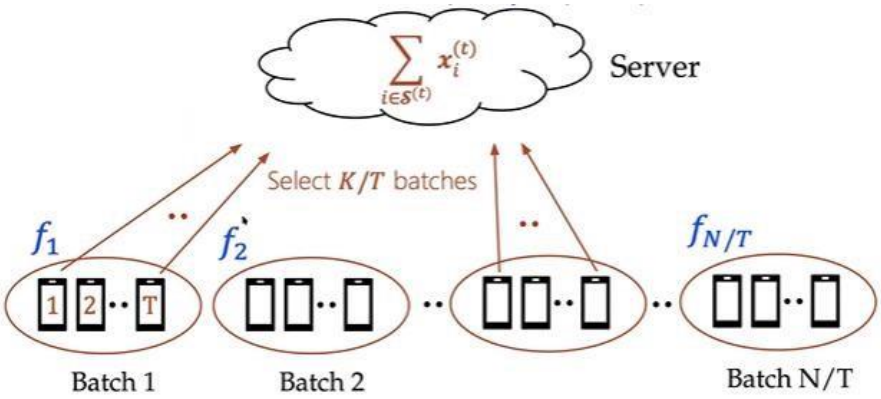

Fig 9: Batch Selection

3. **Multi-Seg SecAgg Theoretical Guarantees**

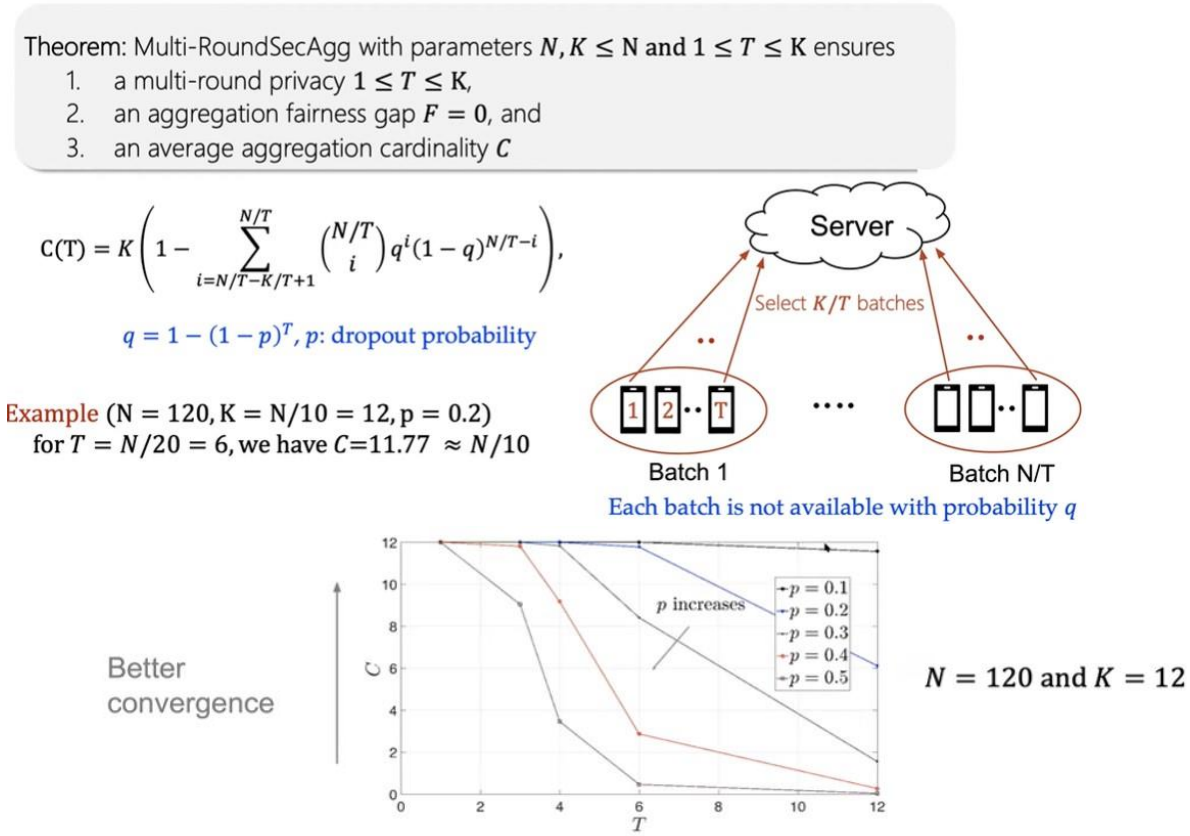

Fig 10: Multi-Seg SecAgg

Optimality of Multi-RoundSecAgg

- Any strategy satisfying the multi-round privacy guarantee must have a batch partitioning structure
- For given N,K<=N/2 and T, any strategy satisfying a multi-round privacy T can have at most Rmax, user sets

$$R_{\max} \leq \binom{N/T}{K/T} = R_{BP}$$ number of sets in BP

3. IMPLEMENTATION

File Formats of Dataset:

The data is stored in a very simple file format designed for storing vectors and multidimensional matrices. General info on this format is given at the end of this page, but you don't need to read that to use the data files.

All the integers in the files are stored in the MSB first (high endian) format used by most non-Intel processors. Users of Intel processors and other low-endian machines must flip the bytes of the header.

There are 4 files:

```
train-images-idx3-ubyte:      training      set      images
train-labels-idx1-ubyte:      training      set      labels
t10k-images-idx3-ubyte:         test       set       images
t10k-labels-idx1-ubyte:  test set labels
```

The training set contains 60000 examples, and the test set 10000 examples.

The first 5000 examples of the test set are taken from the original NIST training set. The last 5000 are taken from the original NIST test set. The first 5000 are cleaner and easier than the last 5000.

TRAINING SET LABEL FILE (train-labels-idx1-ubyte):

```
[offset] [type]          [value]
[description] 0000     32 bit integer
0x00000801(2049) magic number (MSB first)
0004     32 bit integer  60000            number of
items
0008     unsigned byte   ??               label
0009     unsigned byte   ??               label
........
xxxx     unsigned byte   ??               label

The labels values are 0 to 9.
```

TRAINING SET IMAGE FILE (train-images-idx3-ubyte):

```
[offset] [type]          [value]
[description]
0000     32 bit integer  0x00000803(2051) magic
number
0004     32 bit integer  60000            number of
images
0008     32 bit integer  28               number of
rows
```

```
0012     32 bit integer  28              number of
columns
0016     unsigned byte   ??              pixel
0017     unsigned byte   ??              pixel
........
xxxx     unsigned byte   ??              pixel
```

Pixels are organized row-wise. Pixel values are 0 to 255. 0 means background (white), 255 means foreground (black).

TEST SET LABEL FILE (t10k-labels-idx1-ubyte):

```
[offset] [type]          [value]
[description] 0000     32 bit integer
0x00000801(2049) magic number (MSB first)
0004     32 bit integer  10000           number of
items
0008     unsigned byte   ??              label
0009     unsigned byte   ??              label
........ xxxx     unsigned byte   ??
label
```

The labels values are 0 to 9.

TEST SET IMAGE FILE (t10k-images-idx3-ubyte):

```
[offset] [type]          [value]
[description]
0000     32 bit integer  0x00000803(2051) magic
number
0004     32 bit integer  10000           number of
images
0008     32 bit integer  28              number of
rows
0012     32 bit integer  28              number of
columns
0016     unsigned byte   ??              pixel
0017     unsigned byte   ??              pixel
........
xxxx     unsigned byte   ??              pixel
```

Pixels are organized row-wise. Pixel values are 0 to 255. 0 means background (white), 255 means foreground (black).

```
x_train = x_train.astype(np.float32)
y_train = y_train.astype(np.int32)
x_test = x_test.astype(np.float32).reshape(10000, 28, 28, 1)
y_test = y_test.astype(np.int32).reshape(10000, 1)
```

```
total_image_count = len(x_train)
image_per_set = int(np.floor(total_image_count/split))
```

```
client_train_dataset = collections.OrderedDict()
for i in range(1, split+1):
    client_name = "client_" + str(i)
    start = image_per_set * (i-1)
    end = image_per_set * i

    print(f"Adding data from {start} to {end} for client : {client_name}")
    data = collections.OrderedDict((('label', y_train[start:end]), ('pixels', x_train[start:end])))
    client_train_dataset[client_name] = data

train_dataset = tff.simulation.datasets.TestClientData(client_train_dataset)

sample_dataset = train_dataset.create_tf_dataset_for_client(train_dataset.client_ids[0])
sample_element = next(iter(sample_dataset))
```

```
def preprocess(dataset):

  def batch_format_fn(element):
    """Flatten a batch `pixels` and return the features as an `OrderedDict`."""

    return collections.OrderedDict(
        x=tf.reshape(element['pixels'], [-1, 784]),
        y=tf.reshape(element['label'], [-1, 1]))


  return dataset.repeat(NUM_EPOCHS).shuffle(SHUFFLE_BUFFER, seed=1).batch(
      BATCH_SIZE).map(batch_format_fn).prefetch(PREFETCH_BUFFER)


preprocessed_sample_dataset = preprocess(sample_dataset)
sample_batch = nest.map_structure(lambda x: x.numpy(), next(iter(preprocessed_sample_dataset)))
```

```
def make_federated_data(client_data, client_ids):
    return [preprocess(client_data.create_tf_dataset_for_client(x)) for x in client_ids]

federated_train_data = make_federated_data(train_dataset, train_dataset.client_ids)
```

```
logdir = "/tmp/logs/scalars/training/"
try:
  tf.io.gfile.rmtree(logdir)  # delete any previous results
except tf.errors.NotFoundError as e:
  pass # Ignore if the directory didn't previously exist.
summary_writer = tf.summary.create_file_writer(logdir)
train_state = training_process.initialize()
```

```
[ ] with summary_writer.as_default():
      for round_num in range(1, NUM_ROUNDS):
        result = training_process.next(train_state, federated_train_data)
        train_state = result.state
        train_metrics = result.metrics
        for name, value in train_metrics['client_work']['train'].items():
          tf.summary.scalar(name, value, step=round_num)
```

These are the few samples of the code training the MNIST Dataset with different techniques, loading with GPU interfaces, Logging the directories to save the outputs, Training the dataset, and rewriting the client work policies.

4. RESULTS AND DISCUSSION

Here, we have analysed normal sound samples and abnormal sound samples and compared them by putting one sound wave over the other. In this way, the faulty machines could be detected among the devices.

First, we have taken a normal sound sample from the mnist dataset. This is the sound sample of a washing machine.

```
plt.figure(figsize=(15,5))
data_normal,sr_normal = librosa.load(normal)
librosa.display.waveshow(data_normal,sr=sr_normal)
print(f'The signals have a {data_normal.shape} shape.')
```

```
The signals have a (220500,) shape.
```

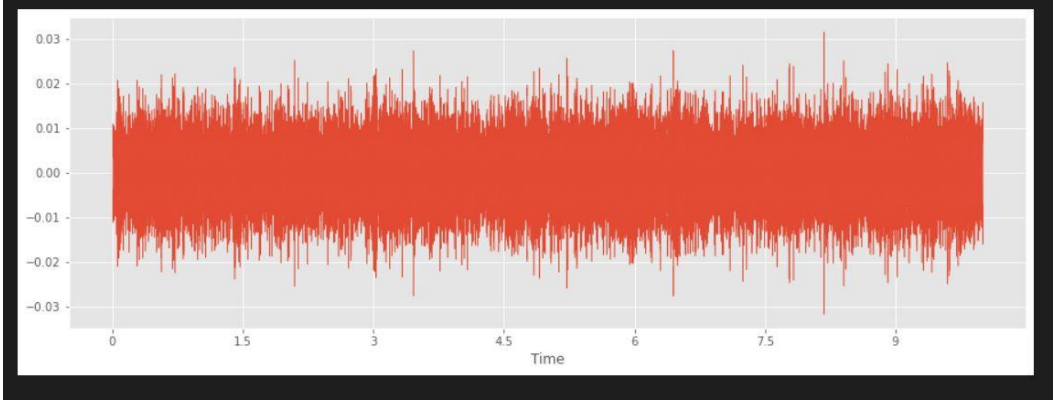

Fig 11: Normal sound sample of a washing machine

Next, we take the input from the user. We collect the sound sample of the abnormal machine.

```
plt.figure(figsize=(15,5))
data_abnormal,sr_abnormal = librosa.load(abnormal)
librosa.display.waveshow(data_abnormal,sr=sr_abnormal)
print(f'The signals have a {data_abnormal.shape} shape.')
```

```
The signals have a (220500,) shape.
```

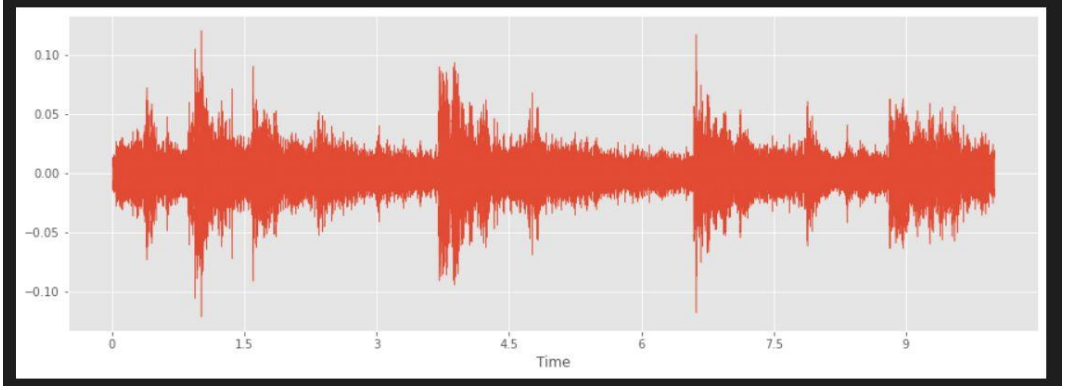

Fig 12: Sound sample of a faulty washing machine

Now we compare the normal and abnormal sound samples and find that there was indeed an error/ fault in the machine. This is where the detection part comes in.

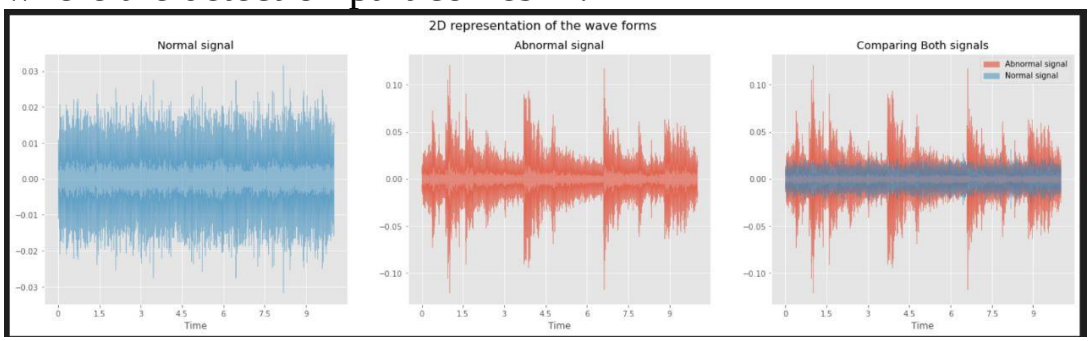

Fig 13: Comparing the normal and faulty outputs

In case of very complex sound samples, we use fourier transform to break the samples into parts and then analyse the defect.

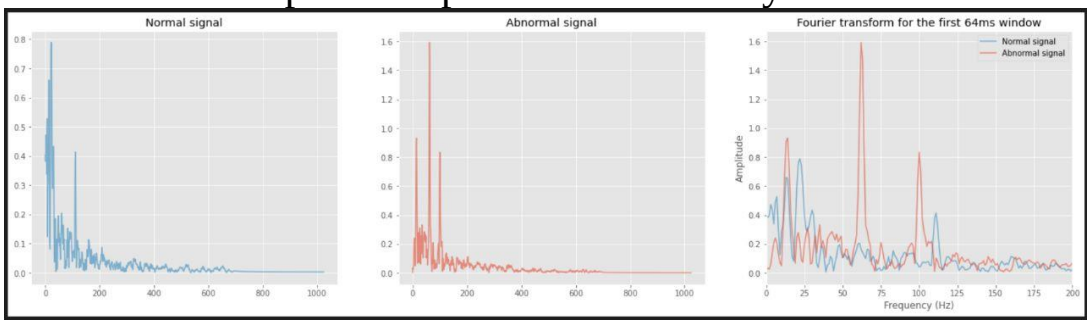

Fig 14: Using fourier transform to understand the difference better

Below is the spectrogram model of the sound sample. In this too, we find that there's a difference between normal and abnormal sounds.

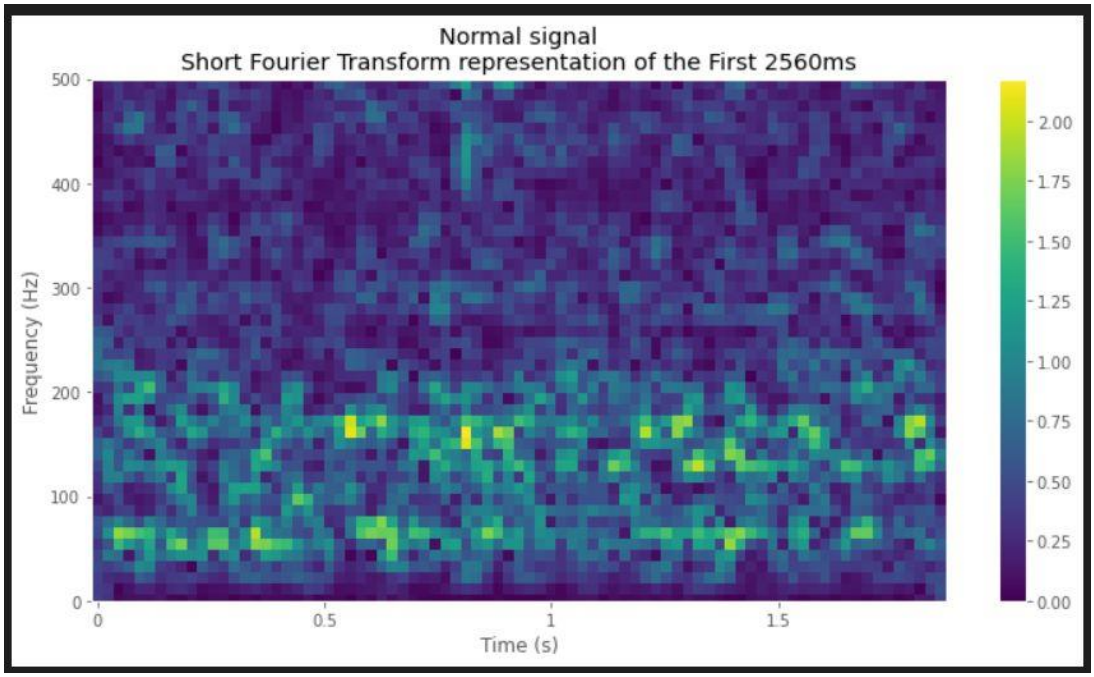

Fig 15: Spectrogram sample of short fourier transform of a normal working device

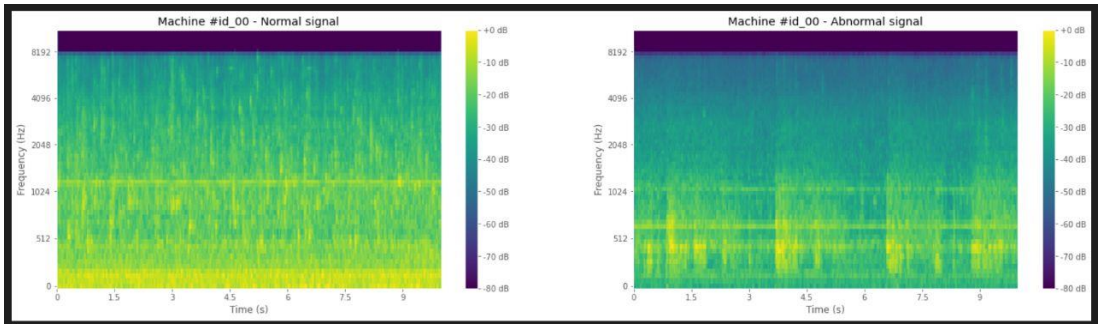

Fig 16: Comparison with an abnormal signal

Here, we can compare the different histogram plot graphs with frequency and time as axes, for both normal and abnormal sounds tracked. Our project will be continued in determining the statements for the problem detection and will be communicated with the clous without any disturbances and data loss.

We are using a very simple neural network with 100 epochs due to limited computational resources and we got an overall score of 76.9% accuracy.

```
Epoch 100/100
13/13 [==============================] - 1s 61ms/step - loss: 0.5880 - accuracy: 0.7253 - val_loss: 0.5451 -
val_accuracy: 0.7692
4/4 [==============================] - 0s 16ms/step - loss: 0.5451 - accuracy: 0.7692
Test accuracy: 0.7692307829856873
```

5. CONCLUSION AND FUTURE WORK

In conclusion, the use of Federated Learning is to train a centralized machine learning model using data distributed among multiple clients with unreliable and sluggish network connections. To improve the security of this method, Secure Aggregation is introduced, which ensures that the weights received from remote devices are encrypted before being forwarded to the central device or aggregated on another remote device. This helps to protect the privacy of the data and the anonymity of the clients. The critical importance of communication efficiency is highlighted, particularly with mobile phones being the most common clients.

Future work in this area could focus on developing more efficient and secure methods for federated learning and secure aggregation, particularly for scenarios where network connections are particularly slow or unreliable. Additionally, research could explore ways to improve the accuracy of the machine learning models produced through Federated Learning, as well as how to incorporate additional privacy and security measures to further protect sensitive data. Finally, there may be opportunities to apply Federated Learning to new domains, such as healthcare or finance, where privacy and security concerns are particularly high.

6. REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998. [on-line version]

[2] Shayan, M., Fung, C., Yoon, C. J. M., & Beschastnikh, I. (2021). "Biscotti: A Blockchain System for Private and Secure Federated Learning". *IEEE Transactions on Parallel and Distributed Systems*, 32(7), 1513–1525.

[3] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). "A Hybrid Approach to Privacy-Preserving Federated Learning". *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security* - AISec'19.

[4] Shen, S., Zhu, T., Wu, D., Wang, W., & Zhou, W. (2020). "From distributed machine learning to federated learning: In the view of data privacy and security" *Concurrency and Computation: Practice and Experience.*

[5] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). "Federated Learning: Challenges, Methods, and Future Directions." *IEEE Signal Processing Magazine*, 37(3), 50–60.

[6] Li, L., Fan, Y., Tse, M., & Lin, K.-Y. (2020). "A review of applications in federated learning." *Computers & Industrial Engineering*, 106854.

[7] Aledhari, M., Razzak, R., Parizi, R. M., & Saeed, F. (2020). "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications." *IEEE Access,* 1–1.

[8] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2020). "A survey on security and privacy of federated learning." *Future Generation Computer Systems.*

[9] Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., … Cardoso, M. J. (2020). "The future of digital health with federated learning." *Npj Digital Medicine*, 3(1).

[10] Hao, M., Li, H., Luo, X., Xu, G., Yang, H., & Liu, S. (2019). "Efficient and Privacy-enhanced Federated Learning for Industrial Artificial Intelligence." *IEEE Transactions on Industrial Informatics*, 1–1.

[11] Jing, X., Yan, Z., & Pedrycz, W. (2018). "Security Data Collection and Data Analytics in the Internet: A Survey" *IEEE Communications Surveys & Tutorials,* 1–1.

[12] Mor, N., Pratt, R., Allman, E., Lutz, K., & Kubiatowicz, J. (2019). "Global Data Plane: A Federated Vision for Secure Data in Edge Computing." 2019 *IEEE 39th International Conference on Distributed Computing Systems* (ICDCS).

[13] So, J., Guler, B., & Avestimehr, A. S. (2021). "Byzantine-Resilient Secure Federated Learning." *IEEE Journal on Selected Areas in Communications*, 39(7), 2168–2181.

[14] Zhan, Y., Zhang, J., Hong, Z., Wu, L., Li, P., & Guo, S. (2021). "A Survey of Incentive Mechanism Design for Federated Learning." *IEEE Transactions on Emerging Topics in Computing*, 1–1.

[15] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., & Karimipour, H. (2021). "Federated learning for drone authentication." *Ad Hoc Networks*, 120, 102574.

[16] Nilsson, A., Smith, S., Ulm, G., Gustavsson, E., & Jirstrand, M. (2018). "A Performance Evaluation of Federated Learning Algorithms." *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning* - DIDL '18.

[17] Kulkarni, V., Kulkarni, M., & Pant, A. (2020). "Survey of Personalization Techniques for Federated Learning." *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability* (WorldS4).

[18] Chai, Z., Ali, A., Zawad, S., Truex, S., Anwar, A., Baracaldo, N., … Cheng, Y. (2020). "TiFL: A Tier-based Federated Learning System." *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*.

[19] "Security and Privacy Threats to Federated Learning: Issues, Methods, and Challenges." Junpeng Zhang, Hui Zhu , Fengwei Wang, Jiaqi Zhao, Qi Xu, and Hui Li. *Volume 2022 | Article ID 2886795 | doi.org/10.1155/2022/2886795*

[20] "Privacy and Security in Federated Learning: A Survey." *Rémi Gosselin, Loïc Vieu, Faiza Loukil, and Alexandre Benoit.* 2022, 12(19), 9901; doi.org/10.3390/app12199901

[21] "Federated Machine Learning: From a Software Engineering Perspective" - Sin Kit Lo , Qinghua Lu , Chen Wang , Hye-

Young Paik , Liming Zhu - June 2022 Article No.: 95pp 1–39https://doi.org/10.1145/3450288:

[22] "A systematic review of federated learning applications for biomedical data" Matthew G. Crowson ,Dana Moukheiber,Aldo Robles Arévalo,Barbara D. Lam,Sreekar Mantena,Aakanksha Rana,Deborah Goss,David W. Bates,Leo Anthony Celi - May 19, 2022 https://doi.org/10.1371/journal.pdig.0000033

[23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. "Communication-efficient learning of deep networks from decentralized data." *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, PMLR 54: 1273-1282, 2017.

[24] M. Li, H. Li, Y. Li, and Y. Liu. "Federated machine learning: concept and applications." *ACM Transactions on Intelligent Systems and Technology*, 10(2):12, 2019.

[25] S. Wang, Y. Tu, C. Zhang, Y. Liu, Y. Liu, T. Zhang, and L. Wang. "Federated Learning on Non-IID Private Data: A Novel Meta-Algorithm and Its Applications." *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI-19, pages 2510-2516, 2019.

[26] M. Kim, Y. Jang, J. Yoo, and J. Paik. "Federated learning-based personalized healthcare recommendation with blockchain." *IEEE Access*, 9: 56961-56970, 2021.

[27] S. S. Mohamed and F. Al-Turjman. "A comprehensive review of blockchain for secure and privacy-preserving federated learning in healthcare." *Journal of Medical Systems*, 45(4): 1-15, 2021.

[28] R. M. Parizi, M. Aledhari, A. Dehghantanha, K. K. R. Choo, and G. Srivastava. "Federated learning in the dark: A survey on approaches for privacy and security." *Future Generation Computer Systems*, 122: 672-689, 2022.

[29] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik. "Federated optimization: Distributed machine learning for on-device intelligence." *ACM Transactions on Intelligent Systems and Technology*, 10(2): 1-19, 2019.

[30] M. Aledhari, A. Dehghantanha, R. M. Parizi, and S. S. Mohamed. "Federated learning for computer vision: Challenges and opportunities." *Computer Vision and Image Understanding*, 204: 113127, 2021.