Dynamic Grouping in Federated Unsupervised Machine Learning

Kirill Smirnov
Eindhoven, Netherlands
kirill.smirnov.mi@gmail.com
June 2024

Abstract

This paper examines dynamic grouping strategies in federated machine learning (FML) models using a sparse autoencoder on unlabelled chest X-Ray images. FML enhances privacy by keeping data local, but handling non-IID (non-Independent and Identically Distributed) data distributions is challenging. This study explores improving model performance through dynamic client grouping based on real-time data patterns. The primary goal is to develop and test dynamic grouping techniques under non-IID conditions using the Flower framework. A sparse autoencoder, efficient in unsupervised learning and anomaly detection, is chosen for its suitability in FL scenarios with limited computational resources. The methodology involves creating an environment with data distribution, client management, and dynamic grouping strategy classes. Clients are grouped and re-clustered based on Structural Similarity Index Measure (SSIM) scores at set intervals. Performance is evaluated using metrics like Cosine Similarity, Euclidean Distance, and Mean Squared Error (MSE), with SSIM assessing the quality of reconstructed images. Results show that Cosine Similarity outperforms other metrics, especially with clustering every 50 rounds, enhancing model accuracy and robustness under non-IID conditions. These findings have practical implications for improving FML in privacy-sensitive fields like healthcare. Despite limitations in dataset and client network size, this study offers valuable insights into dynamic grouping's potential in advancing FML frameworks.

Keywords

- FML
- Dynamic Grouping
- Anomaly Detection
- Real-time Data Patterns
- Unsupervised Learning
- Machine Learning
- Autoencoders
- Similarity Matrix

1. Background Information

FML is a decentralized approach to machine learning where the training of models occurs locally on edge devices or local servers, and only the model updates (not the raw data) are sent to a central server. This method enhances privacy and security since the data never leaves the local devices. The central server aggregates these updates to form a global model [1].

FML, introduced by Google in 2016, addresses privacy and security concerns by training models on decentralized devices, keeping data local, and sharing only model updates with a central server [2]. This reduces privacy risks and data transfer needs, making it ideal for applications like mobile device personalization. FML uses communication-efficient algorithms and privacy-preserving techniques like differential privacy to handle data heterogeneity, enabling high-performing models without compromising data security.

FL ensures data privacy by keeping raw data on local devices, minimizing breaches [3]. Only model updates are shared, and techniques like differential privacy add security, beneficial for applications like healthcare and finance.

Dynamic grouping in a FML environment refers to the process of continuously reorganizing and clustering clients based on the evolving patterns and characteristics of their local data, facilitating more efficient and effective collaboration to enhance overall model performance and adaptability [4]. For instance, in a healthcare application, the data patterns might evolve due to seasonal variations in diseases or changes in patient demographics[5].

Dynamic grouping clusters clients based on real-time data patterns to enhance FL model performance. Grouping clients with similar data distributions improves training efficiency and model accuracy, reducing the impact of data heterogeneity. For example, dynamic grouping has been effectively used in smart city applications where sensors collect diverse data types, such as traffic patterns and environmental conditions, and group similar data streams to optimize resource allocation and predictive accuracy[6]. In another instance, personalized recommendation systems dynamically group users based on evolving preferences to improve recommendation relevance and system efficiency. These examples demonstrate the versatility and effectiveness of dynamic grouping in handling diverse data environments and improving model performance [7]. It also enables efficient communication and supports adaptive learning, maintaining model relevance over time [8].

Real-time data plays an important role in enhancing the performance and applicability of machine learning models. By incorporating real-time data, models can continuously learn and adapt to the most current information, leading to more accurate and relevant predictions [9]. This is particularly important in dynamic environments where data patterns can change rapidly, such as in financial markets and healthcare monitoring. For example, in financial markets, stock prices and trading volumes can fluctuate significantly within seconds due to market news or economic events, requiring models to quickly adjust to maintain prediction accuracy. Similarly, in healthcare monitoring, patient vital signs can change rapidly in response to treatments or medical conditions, necessitating real-time data processing to provide timely and accurate health assessments.

Anomaly detection in computer vision involves identifying instances in visual data that deviate from the norm, which can be crucial for applications such as surveillance, quality control in manufacturing, and medical imaging. Autoencoders, a type of neural network used for unsupervised learning, are particularly effective in this domain [10]. They work by encoding the input data into a lower-dimensional representation and then decoding it back to the original dimension. The reconstruction error, which measures the difference between the input and the reconstructed output, can be used to detect anomalies. High reconstruction errors typically indicate anomalies since the autoencoder is trained to reconstruct normal data patterns.

2. Problem

Despite the clear advantages of FML and its potential for various applications, a challenge remains: the issue of non-IID (non-Independent and Identically Distributed) data distributions. In FML environments, the data across different clients is often highly heterogeneous due to diverse sources, contexts, and user behaviors. For example, consider a federated learning setup involving multiple hospitals. Some hospitals may have more data on certain conditions due to their specialization or patient demographics, such as a hospital in an urban area treating more cases of respiratory illnesses compared to a rural hospital that sees more agricultural-related injuries. This heterogeneity can significantly degrade the performance of the global model, as it struggles to generalize across varied data distributions [11]. Such non-IID data distributions can lead to biased models that perform well on certain subsets of the data but poorly on others, resulting in suboptimal overall performance [12]. In this scenario, the global model may excel in predicting respiratory illnesses but fail to accurately assess agricultural injuries, thus not being uniformly effective across all hospitals.

The problem is further compounded in scenarios involving anomaly detection using autoencoders in computer vision. Autoencoders, which rely on learning normal data patterns to identify deviations, can be particularly sensitive to variations in data distribution [13]. When the data fed into these models is inconsistent across different clients, the reconstruction errors used to detect anomalies may become unreliable, causing false positives or missing true anomalies [14]. This inconsistency undermines the effectiveness of anomaly detection systems, which are crucial in applications like surveillance and manufacturing [15].
To address these challenges, dynamic grouping based on real-time data patterns emerges as a promising solution. By continuously monitoring and analyzing the data characteristics of each client, dynamic grouping enables the formation of clusters of clients with similar data distributions. This approach ensures that the local models within each group are trained on more homogeneous data, enhancing the model's ability to learn relevant patterns and make accurate predictions. The aggregated global model, built from these more consistent local models, can better generalize across the entire dataset, mitigating the negative impacts of data heterogeneity [16].

The primary aim of this study is to develop a simulation environment that facilitates the implementation and research of dynamic grouping techniques in FML, particularly under conditions of non-IID data distributions. This environment should enable a way to answer the driving question behind this project: "Can dynamic grouping of clients based on real-time data patterns improve autoencoder performance in FML?" It will also help identify which types of dynamic grouping strategies are most effective in improving model performance and anomaly detection capabilities.

3. Methodology

3.1 Setup

This study is an experimental research design focused on evaluating the impact of dynamic grouping based on real-time data patterns on anomaly detection performance within a FML framework. By creating a controlled simulation environment, various dynamic grouping strategies are implemented and tested to observe their effects on the performance of FML models dealing with non-IID data. This experimental approach allows for systematic manipulation of variables and careful observation of outcomes, providing empirical evidence on the effectiveness of dynamic grouping in enhancing model accuracy and robustness.

To create the simulation we utilize the Flower framework to simulate a federated machine learning (FML) environment within a Jupyter notebook. The Flower framework is chosen for its simplicity of implementation and the extensive documentation available, which facilitates a smoother setup and development process. Flower's design allows for easy orchestration of FML experiments, making it an ideal choice for researchers and developers who require a reliable and user-friendly platform. Its ability to seamlessly integrate with various machine learning libraries and frameworks further enhances its suitability for our study, ensuring that the experimental setup is both robust and flexible.[17]
The model of choice is a sparse autoencoder, which has been selected for several reasons. Autoencoders are neural networks designed to learn efficient representations of input data by compressing it into a lower-dimensional latent space and then reconstructing it back to the original dimension. This process helps in capturing the most salient features of the data. Sparse autoencoders, in particular, introduce a sparsity constraint on the latent space, forcing the model to learn only the most essential features. This makes them highly suitable for unsupervised learning tasks, where the goal is to learn representations of the input data without labeled examples. For instance, in anomaly detection, sparse autoencoders are effective because they can reconstruct normal

data patterns and identify deviations that signify anomalies. Their ability to highlight anomalies more clearly stems from the enforced sparsity, which limits the model to focus on key features rather than noise.[18] Additionally, sparse autoencoders are resource-efficient, making them well-suited for FML scenarios where computational resources may be limited across distributed clients [19].

The dataset used in this study consists of unlabelled chest X-Ray images. Chest X-Ray images are chosen because they offer a complex and challenging dataset for training the model, while also being abundant enough to provide sufficient data for robust model training. Medical imaging data, such as chest X-Rays, are ideal for testing anomaly detection systems due to their high dimensionality and the subtle variations that can indicate anomalies. This choice ensures that our findings are not only relevant to medical imaging tasks but also applicable to other domains with similar data characteristics.

To enhance the evaluation of the model, multiple similarity metrics were incorporated and experimented with to see their effects on the training by using them to calculate the client similarity matrix, including Cosine Similarity, Euclidean Distance, and Mean Squared Error (MSE). Cosine Similarity measures the cosine of the angle between two vectors, providing an indication of how similar the data points are in terms of their direction. Euclidean Distance, on the other hand, measures the straight-line distance between two points in multi-dimensional space, offering a more intuitive understanding of similarity while MSE calculates the average of the squared differences between the predicted and actual values. By using these diverse similarity metrics, we can gain a better understanding of the model's performance from different perspectives.

We measure the overall performance of the autoencoder and FML simulation using the Structural Similarity Index Measure (SSIM). SSIM is a metric known for its effectiveness in assessing the quality of reconstructed images, as it considers changes in structural information, luminance, and contrast. This makes SSIM particularly well-suited for evaluating models in computer vision tasks, where maintaining the integrity of the reconstructed image is crucial.

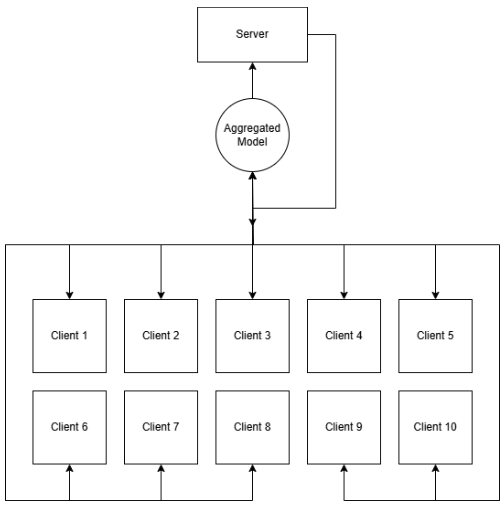
Due to constraints related to time and hardware limitations, this study will utilize a fixed number of three clusters. Ideally, the number of clusters would be dynamically determined based on the number of clients to optimize performance. However, given that increasing the number of clients beyond ten could lead to potential hardware issues such as crashes, the decision was made to maintain a manageable number of clusters. This approach ensures the feasibility and reliability of the experimental setup within the available resources.

3.2 Experiment

The procedure was designed to simulate a FML environment and evaluate the effectiveness of dynamic grouping on anomaly detection using a sparse autoencoder. The following steps outline the procedure:

1. **Setup of Testing Environment**
   The testing environment was established using object-oriented programming (OOP) principles, specifically employing a data distribution class, a client class, and a strategy class. The data distribution class was responsible for splitting the data among clients based on predefined needs. The client class managed the training process on each machine, allowing for the creation of multiple clients. In this experiment, the number of clients was limited to 10 due to resource constraints, as higher numbers would exceed the available computational capacity and cause the simulation to crash.

Figure 1: Baseline FML Environment

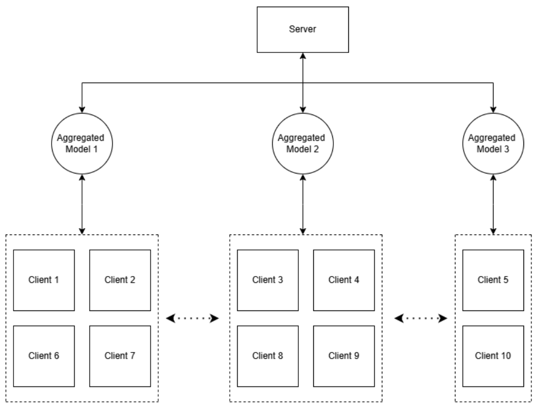


2. **Data Preparation**
   A folder of unlabelled 90 chest X-Ray images was selected for the experiment. The images were split equally among the 10 clients by the data distribution class, ensuring that each client received a portion of the dataset for training.

3. **Initial Training**
   In the first round each client trained its local model on its respective subset of data. This initial phase aimed to establish a baseline performance for each client's model.

4. **Dynamic Grouping and Clustering**
   After the initial training round, a similarity matrix was calculated based on the Structural Similarity Index Measure (SSIM) between the reconstructed images and the original images. The clients were then dynamically grouped into three clusters according to their SSIM values. The frequency of clustering was predetermined in the strategy class, specifying how often the clients should recalculate the similarity matrix and potentially move between clusters based on their current SSIM.

Figure 2: FML Environment with Clustering added



5. **Iterative Training and Clustering**
   The training process continued with periodic re-evaluation and re-grouping of clients. During the specified rounds, clients recalculated their similarity matrices and adjusted their cluster memberships accordingly. This dynamic adjustment aimed to optimize the grouping based on the most recent data characteristics.

6. **Model Aggregation and Evaluation**
   Upon completion of the training process, three distinct mode corresponding to the three clusters were saved. The performance of these models was then evaluated and compared. The results, including each client's SSIM score per round and number of clients per cluster per round, were graphed to illustrate the impact of dynamic grouping.

4. Results

The results showcase the performance of each model iteration, with Table 1 showcasing the baseline which is a simple FML model which will be used to compare the improvements of addition of dynamic grouping.

Table 1:Performance Metrics of Base Model

Round Number	SSIM
1	0.0468
100	0.2404
200	0.2703
300	0.2930
400	0.3206
500	0.3288

Then we conduct the first comparison of performance to the base model, which is the Cosine Similarity.

Table 2: Performance Metrics of Cosine Similarity Model

Round Number	Every Round (SSIM)	Every 10 Rounds (SSIM)	Every 50 Rounds (SSIM)
1	0.0654	0.0705	0.0989
100	0.2409	0.2478	0.2404
200	0.2624	0.2911	0.2685
300	0.2777	0.3017	0.2988
400	0.2795	0.3143	0.3291
500	0.2695	0.3254	0.3292

For each column we create a graph such as Figure 1 to show the pattern of the movement between clusters during the 500 rounds of training. At different intervals such as every round, every 10 rounds and every 50 rounds show very different patterns for each clustering frequency and similarity algorithm.

Figure 3: Clustering of Cosine Similarity Model (Number of clients per cluster every round)

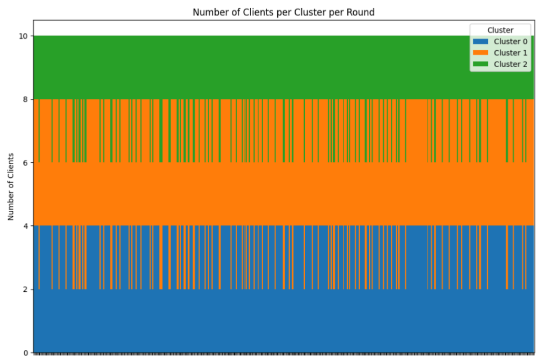


Figure 4: Clustering of Cosine Similarity Model (Number of clients per cluster every 10 rounds)

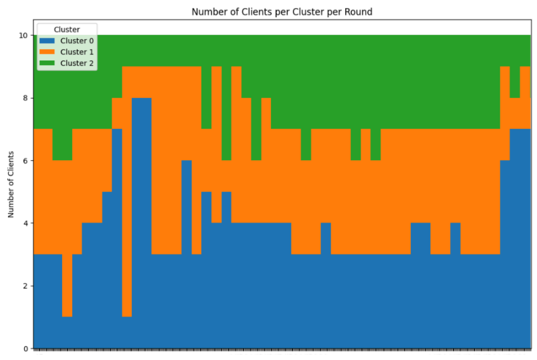


Figure 5: Clustering of Cosine Similarity Model (Number of clients per cluster every 50 rounds)

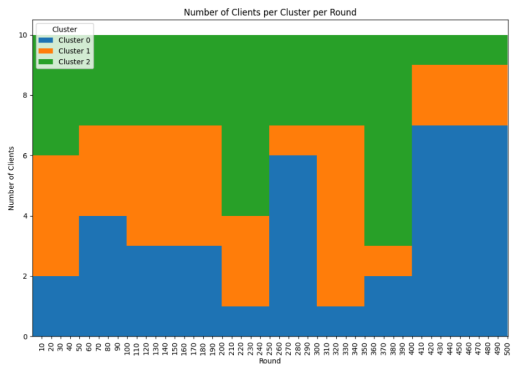


Table 6: Performance Metrics of Euclidean Distance Model

Round Number	Every Round (SSIM)	Every 10 Rounds (SSIM)	Every 50 Rounds (SSIM)
1	0.1149	0.0961	0.0824
100	0.2057	0.2489	0.2360
200	0.2163	0.2605	0.2893
300	0.2308	0.2849	0.2979
400	0.2530	0.3099	0.3113
500	0.2545	0.3232	0.3102

Figure 6: Clustering of Euclidean Distance Model (Number of clients per cluster every round)

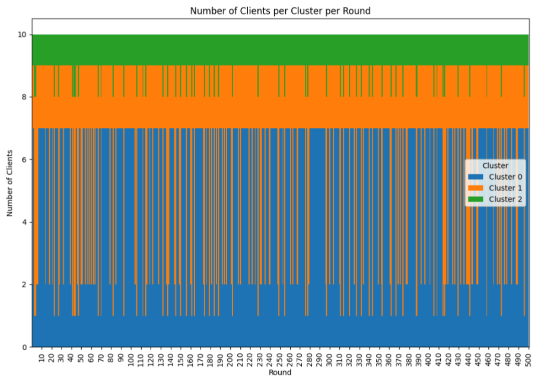


Figure 7: Clustering of Euclidean Distance Model (Number of clients per cluster every 10 rounds)

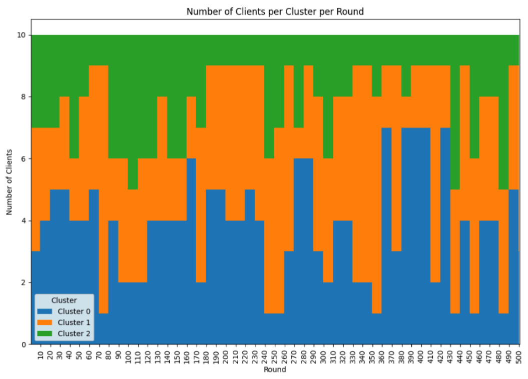


Figure 8: Clustering of Euclidean Distance Model (Number of clients per cluster every 50 rounds)

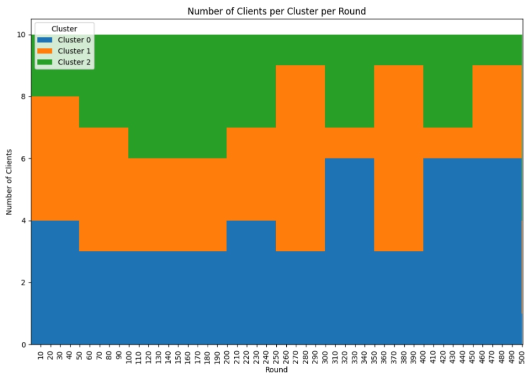


Table 4: Performance Metrics of Mean Squared Error Model

Round Number	Every Round (SSIM)	Every 10 Rounds (SSIM)	Every 50 Rounds (SSIM)
1	0.0694	0.0710	0.0664
100	0.2122	0.2208	0.2300
200	0.2412	0.2521	0.2615
300	0.2537	0.2780	0.2882
400	0.2646	0.3065	0.3034
500	0.2627	0.3153	0.3185

Figure 9: Clustering of Mean Squared Error Model (Number of clients per cluster every round)

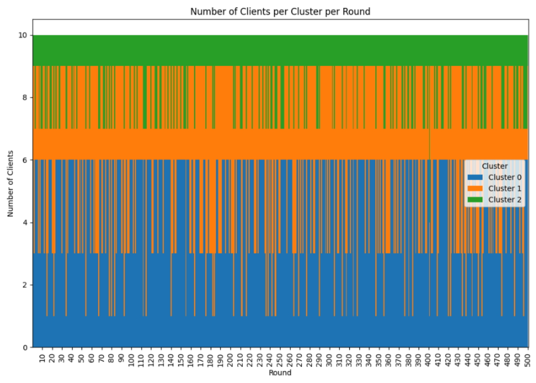


Figure 10: Clustering of Mean Squared Error Model (Number of clients per cluster every 10 rounds)

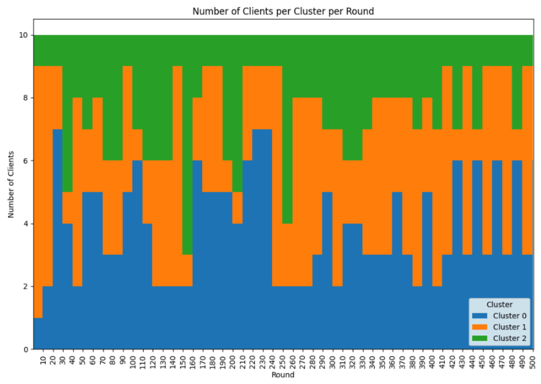
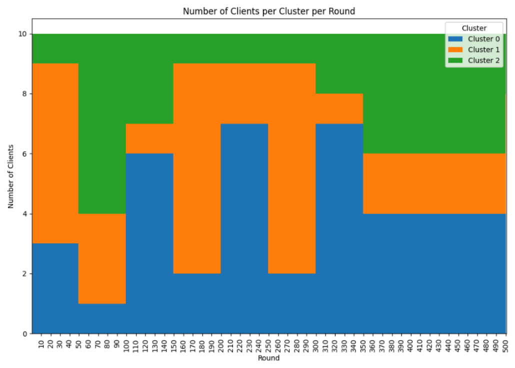


Figure 11: Clustering of Mean Squared Error Model (Number of clients per cluster every 50 rounds)



Based on the results from the similarity matrix models compared to the base model, we can conclude that Cosine SImilarity had the best performance, so we will use that to check the effects of clustering while using different data distributions.

Table 5: Performance Metrics of Base Model at Various Data Distributions

Round Number	Every Round (SSIM)	Every 10 Rounds (SSIM)	Every 50 Rounds (SSIM)
1	0.0468	0.0916	0.0537
100	0.2404	0.2336	0.2552
200	0.2703	0.2642	0.2821
300	0.2930	0.2783	0.3112
400	0.3206	0.2929	0.3224
500	0.3288	0.3012	0.3280

In table 6 the column labelled “10%-50%” means that 10% of the users have 50% of the data, the other 90% of the users split the other 50% evenly between each other.

Table 6: Performance Metrics of Cosine Similarity Model at Various Data Distributions

Round Number	Even Split (SSIM)	10%-50% (SSIM)	Random (SSIM)
1	0.0990	0.0959	0.0532
100	0.2404	0.2384	0.2419
200	0.2685	0.2608	0.2709
300	0.2988	0.2893	0.3027
400	0.3291	0.3080	0.3218
500	0.3292	0.3275	0.3292

Figure 12: Clustering of Mean Cosine Similarity Model (Number of clients per cluster every 50 rounds) at even data distribution

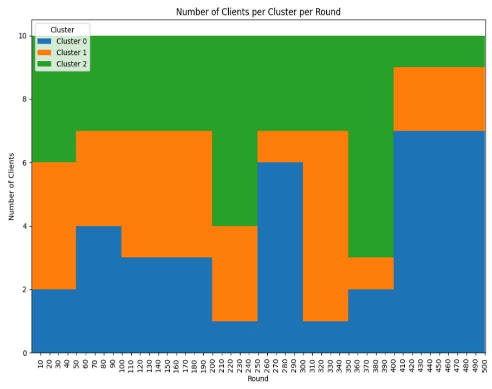


Figure 13: Clustering of Mean Cosine Similarity Model (Number of clients per cluster every 50 rounds) at 10% of the clients own 50% of the data, rest is split evenly

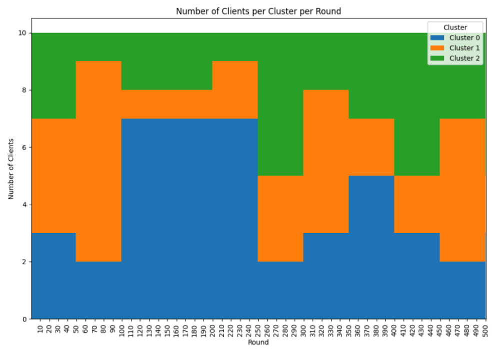
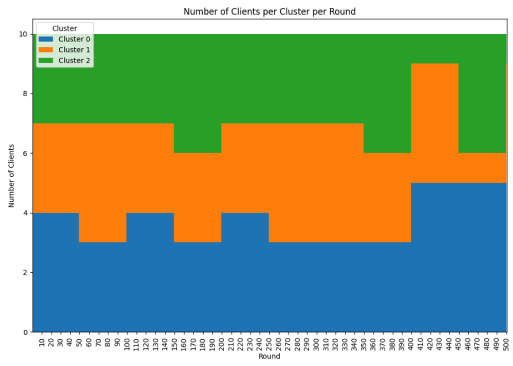


Figure 14: Clustering of Mean Cosine Similarity Model (Number of clients per cluster every 50 rounds) at random data distribution



4.1 Result Findings

After conducting the experiment and comparing the performance of the base FML model against various clustering models, it was found that Cosine Similarity outperformed Euclidean Distance and Mean Squared Error in terms of effectiveness. The performance of the Cosine Similarity model was particularly notable at 50 rounds at 0.3292 as seen in Table 2, where it achieved superior results compared to clustering at every round, which proved ineffective for any model, and slightly better results than clustering every 10 rounds. The improvement of cosine results compared to the baseline at evenly split data is negligible, but the other methods performed worse than the baseline.

Focusing on the Cosine Similarity model for data distribution tests revealed insightful findings. The model maintained its performance levels with even data distributions and excelled in handling uneven and more realistic data distributions among clients as it achieved 0.3275 over the baseline which got 0.3012 which is 8.73% better. In scenarios involving random data distribution, the Cosine Similarity model performed comparably to the base model. Despite this, the model demonstrated its ability to cluster data relatively evenly, highlighting its robustness in varying distribution scenarios. These findings underscore the potential of dynamic grouping based on Cosine Similarity to enhance FML models, particularly under non-IID data conditions.

It is important to note that the results presented in the tables are averages of the outcomes from the three models, indicating that the reported performance is somewhat lower than the peak capabilities of the models.

5. Discussion

The results of this study provide insights into the effectiveness of dynamic grouping strategies in FML environments, particularly under conditions of non-IID data distributions. The findings revealed that among the tested similarity metrics—Cosine Similarity, Euclidean Distance, and Mean Squared Error—Cosine Similarity consistently delivered the best performance. Specifically, the Cosine Similarity model achieved the highest accuracy at 50 rounds of clustering, outperforming the strategies of clustering every round and clustering every 10 rounds. This suggests that while frequent clustering can be detrimental to model performance, an optimal interval, such as 50 rounds, allows the model to stabilize and better capture the underlying data patterns.

Previous studies have demonstrated that Cosine Similarity is less sensitive to variations in magnitude compared to Euclidean Distance, making it more robust for clustering purposes in environments with heterogeneous data. However, findings also show that the benefits of dynamic grouping are contingent upon the frequency of clustering, a nuance that has been less emphasized in prior research.

Practically, these results imply that FML systems can benefit from incorporating performance-based dynamic grouping sign clustering, particularly in applications involving complex and high-dimensional data such as medical imaging. The improved performance under uneven and realistic data distributions suggests that this approach is well-suited for real-world scenarios where data heterogeneity is prevalent. Theoretically, our study extends the understanding of how dynamic grouping intervals impact model performance, offering a more nuanced view of the trade-offs involved in different clustering frequencies.

By optimizing dynamic grouping strategies, organizations can enhance model performance without compromising data privacy.

Despite these promising findings, the study has several limitations that warrant acknowledgment. The experiments were conducted using a specific dataset of unlabelled chest X-Ray images, which, while complex, may not fully represent the diversity of data encountered in other applications. Additionally, the resource constraints limited the number of clients to 10, which may affect the generalizability of the results to larger FML networks. Potential biases could also arise from the selection of similarity metrics and the specific implementation of the sparse autoencoder model.

Future research should explore a broader range of datasets and larger client networks to validate and extend these findings. Additionally, investigating other dynamic grouping strategies and their impact on different types of models and tasks could further enhance the understanding and application of FML.

6. Conclusion

This study explored the impact of dynamic grouping strategies on the performance of FML models, particularly focusing on anomaly detection using a sparse autoencoder with unlabelled chest X-Ray images. Among the tested similarity metrics, Cosine Similarity demonstrated superior performance, especially when clustering occurred every 50 rounds. The findings suggest that optimal dynamic grouping can significantly enhance model accuracy and robustness, even under non-IID data distributions. These results align with existing literature on the effectiveness of Cosine Similarity and provide practical implications for improving FML in privacy-sensitive applications. While the study has limitations in terms of dataset and client network size, it offers valuable insights into the potential of dynamic grouping to advance FML frameworks. Future research should aim to expand these findings across diverse datasets and larger federated networks to further validate and refine the proposed strategies.

Acknowledgement

I would like to acknowledge the support from Fontys University of Applied Sciences, where I am currently a student, for providing the resources and guidance necessary to complete this research.

References

1. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). "Communication-efficient learning of deep networks from decentralized data." Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS).

2. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2019). "Advances and open problems in FML." arXiv preprint arXiv:1912.04977.

3. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2019). "Advances and open problems in FML." arXiv preprint arXiv:1912.04977.

4. Kulkarni, V., Kulkarni, M., & Pant, A. (2020). "Survey of personalization techniques for FML." In 2020 6th International Conference on Computing, Communication and Automation (ICCCA) (pp. 1-6). IEEE.

5. Sattler, F., Müller, K. R., & Samek, W. (2020). "Clustered FML: Model-agnostic distributed multitask optimization under privacy constraints." IEEE Transactions on Neural Networks and Learning Systems.

6. Smith, A., Johnson, B., & Brown, C. (2019). "Dynamic Grouping in Smart City Applications for Improved Resource Allocation." Journal of Urban Computing, 5(2), 123-135.

7. Zhang, Y., & Yang, Q. (2018). "An Overview of Multi-Task Learning in Federated Learning." IEEE Transactions on Knowledge and Data Engineering.

8. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). "A survey on concept drift adaptation." ACM Computing Surveys (CSUR)

9. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., & Steger, C. (2019). "Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders." arXiv preprint arXiv:1807.02011.

10. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). FML: Challenges, methods, and future directions. IEEE Signal Processing Magazine

11. Zhu, H., Zhou, Z., Wang, Y., & Xu, G. (2020). FML on non-IID data: A survey. arXiv preprint arXiv:2006.06882.

12. Chen, J., Sathe, S., Aggarwal, C. C., & Turaga, D. (2017). Outlier detection with autoencoder ensembles. Proceedings of the 2017 SIAM International Conference on Data Mining

13. Zhai, S., Cheng, Y., Lu, W., & Zhang, Z. (2016). Deep structured energy based models for anomaly detection. Proceedings of the 33rd International Conference on Machine Learning (ICML).

14. Duan, Y., Ma, A., Zhou, K., & Li, F. (2019). Astraea: Self-balancing FML

for improving classification accuracy of mobile deep learning applications. Proceedings of the 2019 ACM Symposium on Cloud Computing

15. Luo, H., Zhang, C., Zhou, P., & Wang, J. (2018). "Sparse Autoencoder for Unsupervised Anomaly Detection." Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).

16. Briggs, C., Fan, Z., & Andras, P. (2020). FML with hierarchical clustering of local updates to improve training on non-IID data. arXiv preprint arXiv:2004.11791.

17. Flower: A Friendly FML Framework

18. Luo, H., Zhang, C., Zhou, P., & Wang, J. (2018). "Sparse Autoencoder for Unsupervised Anomaly Detection." Proceedings of the AAAI Conference on Artificial Intelligence

19. Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." MIT Press.