Federated Learning in Edge Devices: Strengthening Data Privacy and Security

Kirill Smirnov
Eindhoven, Netherlands
Fontys University Of Applied Science
Masters Of Applied IT
k.smirnov@student.fontys.nl

January 2024

Abstract

This study investigates performance-based dynamic clustering in Federated Machine Learning (FML) for possible usage in autonomous vehicles, focusing on mitigating data poisoning in resource-constrained edge environments. By grouping clients based on performance metrics, the approach enhances model robustness under varying adversarial conditions, including poisoned clients and diverse data distributions. Using MobileNetv3, the results demonstrate improved accuracy and resilience, offering insights into securing FML systems.

Keywords

- Federated Machine Learning
- Edge Devices
- FML Security
- Dynamic Clustering
- Autonomous Vehicles
- Object Recognition

1. Background information

1.1 Introduction to Edge Devices

Edge devices encompass a diverse range of technologies such as IoT devices, smartphones, smartwatches and more. These devices are equipped with sensors and computational capabilities that enable them to collect, process, and transmit data directly at the source, rather than relying solely on centralized cloud systems [1]. This shift towards edge computing has improved data processing by reducing latency, minimizing bandwidth usage, and supporting real-time applications. Beyond these technical advantages, edge devices are indispensable in domains such as healthcare, smart cities, and autonomous vehicles, where timely and localized data processing directly impacts decision-making and outcomes [2].

1.2 Challenges in Edge Computing

Despite their advantages, the vast amounts of data generated by edge devices bring new challenges, including data privacy, communication costs, and hardware constraints. These challenges highlight the need for innovative solutions to ensure efficient and secure data management in resource-constrained environments.

1.3 Federated Machine Learning as a Solution

Federated Machine Learning (FML) offers an innovative solution to the challenges posed by edge computing. Introduced by Google in 2016, FML enables the collaborative training of machine learning models across distributed devices without the need to transfer raw data to a central server [3]. Unlike traditional centralized AI, which aggregates all data in a single location for processing, FML operates in a decentralized manner. This ensures that sensitive information remains local, addressing critical privacy concerns while still being able to align with the computational constraints of edge devices. FML is particularly practical for applications like autonomous driving systems, where data sensitivity and real-time responses are paramount [4].

1.4 Integration of FML with Edge Computing

By integrating FML with edge computing, the benefits extend beyond privacy. Edge-enabled FML preserves bandwidth, reduces reliance on cloud infrastructure, and enhances system scalability. For instance, in autonomous vehicles, FML allows vehicles to collaboratively learn traffic patterns, road conditions, and driving behaviors while keeping raw sensory data onboard. This localized learning minimizes data transfer while still enabling fleet-wide model improvements, ultimately enhancing safety and efficiency. Additionally, the decentralized nature of FML mitigates risks associated with central data breaches, making it a robust solution for other sensitive applications such as healthcare and finance [5].

2. Research Problem

2.1 Benefits and Requirements of Federated Machine Learning

Federated Machine Learning (FML) offers multiple advantages over traditional centralized AI systems, particularly in scenarios requiring enhanced privacy and the ability to manage diverse data distributions. However, its adoption in edge environments depends on specific conditions. For FML to outperform centralized AI, edge devices such as sensors, smart meters, or autonomous vehicles must have sufficient computational resources to train models locally. Furthermore, the network infrastructure must support frequent exchanges of model updates with a central server. These requirements present major challenges in resource-constrained environments, where limited processing power, memory and other hardware limitations are common [6].

2.2 Challenges in Edge Environments

Resource constraints are especially pronounced in applications like autonomous vehicles. Sensors in these systems must process real-time data from cameras, radar, and lidar to make immediate safety decisions, leaving little capacity for computationally intensive tasks such as local model training [7]. In addition, edge environments often deal with **Non-IID (Non-Independent and Identically Distributed)** data. In FML, **IID (Independent and Identically Distributed)** data assumes all clients share similar distributions, such as autonomous vehicles collecting equal traffic patterns and weather data. Non-IID data reflects real-world variability, like vehicles in urban, rural, or extreme weather conditions, creating challenges in model convergence and performance. These discrepancies complicate the learning process, impede model convergence, and degrade global model performance [8].

2.3 Security Vulnerabilities in FML

FML also faces security challenges similarly to normal AI, with data poisoning being one of the most common threats. In autonomous vehicles, adversaries can inject malicious updates into the global model by manipulating sensor data on compromised vehicles. These poisoned updates can severely degrade the global model’s ability to detect road signs, predict traffic conditions, or identify obstacles, potentially leading to hazardous outcomes [9]. For instance, attackers could alter data to misinterpret speed limits or traffic signals, causing accidents or chaos on the road. In high-stakes environments like autonomous driving, these threats jeopardize public safety and erode trust in autonomous technology and its adoption [10]. The decentralized nature of FML exacerbates these vulnerabilities, as malicious clients can exploit the lack of direct data visibility to inject corrupted updates undetected. While centralized AI systems rely on direct data auditing, FML’s privacy-preserving design limits such oversight, making robust detection and mitigation strategies essential for secure implementation in edge environments.

2.4 Role of Clustering in FML

FML clustering involves grouping edge devices based on performance metrics, data characteristics, or contribution consistency to create subsets, or clusters, of clients that share similar behaviors. By tailoring the aggregation process to these clusters, FML can address the unique challenges of edge environments. Clustering is able to enhance FML's ability to manage Non-IID data by aligning similar data distributions within clusters[11], improving model convergence and reducing the negative impact of data discrepancies.

2.5 Clustering as a Security Measure

Clustering can enhance the security of FML systems by isolating and mitigating the effects of adversarial attacks such as data poisoning. Clustering minimizes the influence of these malicious updates by limiting their impact to specific clusters, reducing the likelihood of system-wide degradation. By grouping clients with similar contributions and isolating those exhibiting anomalous behavior, clustering provides an additional layer of defense against undetected attacks.

2.6 Objectives

This research aims to evaluate the effectiveness of performance-based dynamic clustering of clients, a method that groups clients based on model accuracy, as a mitigation strategy against data poisoning in FML (Federated Machine Learning) systems, using a simulated environment. By grouping clients based on performance metrics, such as model accuracy, the study seeks to identify and minimize the impact of malicious clients. The research investigates how this approach can enhance system robustness in diverse scenarios, including instances where over half of the clients are compromised, environments with Non-IID data distributions, and varying levels of poisoned data. Conducting the study in a simulated environment allows for controlled testing of these scenarios, facilitating a deeper understanding of the approach's potential to maintain model integrity and reliability in real-world edge computing applications.

3. Methodology

3.1 Clustering Strategies

Building on the identified challenges, this study investigates clustering-based approaches to address the vulnerabilities of FML on edge devices, particularly autonomous vehicles. Clustering, or dynamic grouping, provides a promising strategy for mitigating the impact of data poisoning by grouping clients based on performance and resource metrics, enabling the isolation of anomalous updates and reducing their effect on the global model.

Static FML clustering groups clients into fixed clusters at the outset of training and maintains these clusters throughout the process, regardless of changes in client behavior or performance. In contrast, Dynamic FML clustering continuously adjusts the grouping of clients based on evolving performance metrics or data patterns. This adaptability makes Dynamic FML clustering more flexible and robust in addressing challenges such as Non-IID data distributions, resource constraints, and adversarial attacks, whereas Static FML clustering is more rigid and may struggle to accommodate these dynamic changes effectively. This approach ensures that malicious updates are mitigated early, safeguarding the global model's accuracy and reliability [12].

The proposed methodology involves monitoring key metrics, including computation time, memory usage, and model update accuracy, to evaluate clustering's ability to reduce the impact of poisoned updates. By focusing on these metrics, the approach aims to maintain the integrity of the global model while ensuring computational efficiency on edge devices [13]. Additionally, the study will explore whether clustering can coexist with the high-priority processes of autonomous vehicles without significant model accuracy degradation.

Simulation-based experiments will provide insights into the viability of clustering on resource-constrained edge devices and its effectiveness in minimizing the impact of adversarial updates. By focusing on the mitigation of poisoned updates, this research seeks to contribute to the secure and reliable application of FML in autonomous vehicle networks and similar edge environments.

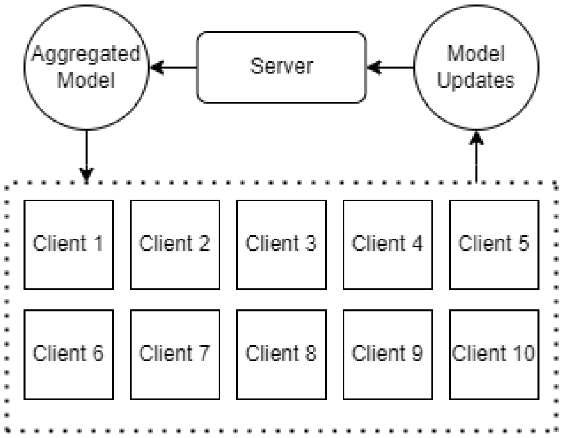

Fig. 1: Base FML simulation setup

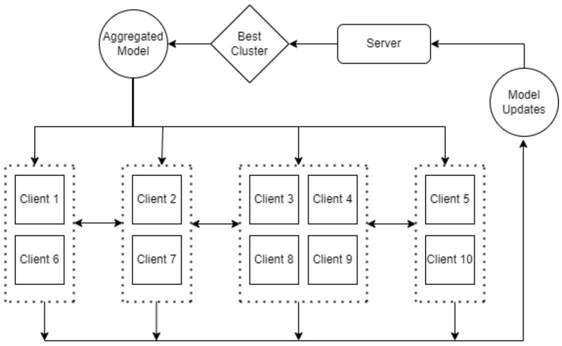

Fig. 2: Clustering FML simulation setup

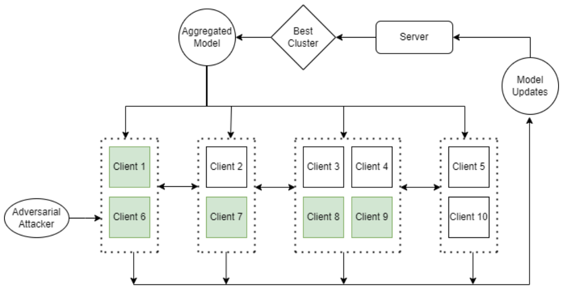

Fig. 3: Clustering FML simulation setup with data poisoning

3.2 Evaluation

The clustering and non-clustering FML models are compared by evaluating their performance under varying adversarial conditions, including the number of poisoned clients, the degree of data poisoning, and data distributions (IID and Non-IID). Clustering is validated as superior if it consistently achieves higher accuracy, F1 score, and lower log loss compared to the baseline model, demonstrating improved resilience against data poisoning attacks.

4. Experiment Setup

The simulation for running the tests utilized the Flower Framework[14], which was chosen for its flexibility and user-friendly environment. Flower supports a wide range of machine learning libraries, including TensorFlow and PyTorch, making it accessible to developers and simplifying the implementation of FML experiments. Flower's design is particularly advantageous as it reduces development overhead while enabling a focus on innovative methodologies.

4.1 Data

The simulation focused on the classification of objects in images captured by car sensors, mimicking real-world scenarios encountered by autonomous vehicles. The dataset used contained 5,000 images sized 480x300, categorized into five classes: 'car,' 'truck,' 'pedestrian,' 'bicyclist,' and 'light.'

4.2 Model

MobileNetv3[15] was selected as the model for this study due to its lightweight architecture and efficiency, making it highly suited for deployment on resource-constrained edge devices such as autonomous vehicle processors. Its design optimises performance while maintaining a minimal computational footprint while still being able to process complex data such as images. This makes MobileNetv3 a suitable choice for real-time image classification tasks, where edge devices require accurate and efficient models without significant computational costs.

4.3 Metrics

The performance of the system was evaluated using accuracy, which measures the proportion of correctly classified images, F1 score, which assesses the balance between precision and recall, and log loss, which evaluates the confidence of predictions while penalizing incorrect ones. Accuracy showcases the effectiveness of reducing the influence of data poisoning by indicating how well the model performs on correctly classifying data despite the presence of corrupted or mislabeled samples. A high accuracy suggests that the clustering mechanism successfully mitigates the impact of poisoned data, allowing the model to maintain robust and reliable predictions even in scenarios where some training data has been compromised. This reflects the system's ability to isolate or counteract the effects of data poisoning, preserving the integrity of the overall learning process.

4.4 Evaluation

To evaluate the effectiveness of dynamic clustering in minimizing the impact of data poisoning attacks in FML, the study compared the Baseline FML Model (without clustering) and the Dynamic Clustering Model. The evaluation considered varying numbers of poisoned clients (up to 9 out of 10), different poisoning percentages per client, and data distributions across clients.

4.5 Variables

The variables for evaluation included the **Number of Poisoned Clients (NPC)** ranged from 0 to 9 out of 10, simulating increasing adversarial influence. The **Data Poisoning Percentage per Client (DPP)** varied between 25% and 100%, reflecting different levels of corruption within client updates. In the simulation, the poisoning percentage determines how much of the data given to an affected client has its labels flipped. Since the simulation only allows setting a single value, at 25% DPP, each affected client will have 25% of its data mislabeled.

Data Distribution per Client (DDPC) was evaluated under two scenarios: **IID** (Independent and Identically Distributed), where all clients had uniform data, and **Non-IID** (Non-Independent and Identically Distributed), where clients had heterogeneous data distributions. These variables provided a detailed assessment of the system’s ability to handle adversarial conditions and diverse data environments.

4.6 Cluster Configuration

The cluster configuration for the Dynamic Clustering Model included a fixed number of four clusters. The best-performing model among the clusters was selected based on validation accuracy. The tests are 40 rounds each, every round consists of each client training for 3 epochs and clustering of clients occurring every 5 rounds.

4.7 Data Distribution

Under IID data distribution, each client received an equal portion of the total data, with a per-client distribution of 10%. In the real world, however, each client generates a varied quantity of data due to differences in activity or usage. To reflect this in the simulation, the Non-IID data distribution assigned varied amounts of data to clients, with per-client distributions ranging from 5% to 30%, introducing deviations from equal distribution.

4.8 Testing scenarios

To replicate various real-world scenarios, testing was divided into four parts: Baseline Experiments Without Poisoning, Effect of Number of Poisoned Clients (NPC) with Fixed DPP (100%), Effect of Data Poisoning Percentage (DPP) with Fixed NPC (5), and

Effect of Data Distribution (DDPC) with Fixed NPC (5) and DPP (100%) in a Non-IID environment. The first part establishes a baseline to ensure clustering doesn't degrade performance; the second tests up how many poisoned clients clustering is effective for; the third examines the required level of client data infection for clustering to be beneficial; and the fourth evaluates clustering performance under Non-IID data combined with data poisoning.

4.9 Tests conducted

Each test was done as a pair, comparing the FML model with and without clustering to see the effects.

Test №	NPC	DPP	DDPC
1 & 2	0	0	IID
3 & 4	1	100	IID
5 & 6	3	100	IID
7 & 8	5	100	IID
9 & 10	7	100	IID
11 & 12	9	100	IID
13 & 14	5	25	IID
15 & 16	5	50	IID
17 & 18	5	75	IID
19 & 20	3	100	Non-IID
21 & 22	5	100	Non-IID
23 & 24	7	100	Non-IID

Table 1: Conducted Simulation Tests Variables

5. Results

The following line graphs showcase the accuracy measurements between the non-clustering (Blue) and clustering (Orange) models. On the y-axis the accuracy is shown (0 to 1 range) and on the x-axis the rounds (up to 40)

5.1 Accuracy graphing

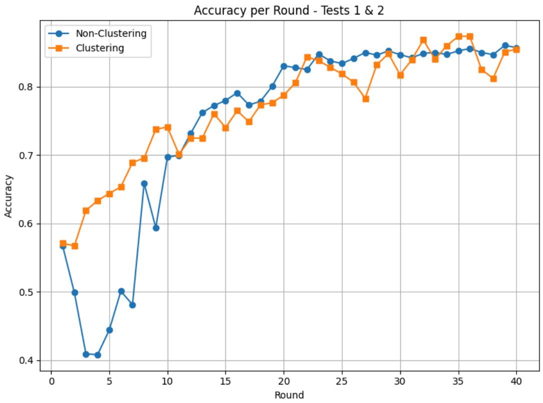

Fig. 4: NPC 0, DPP 0, DDPC IID

As seen in **Figure 4** The first 2 tests compare the base performance of the 2 models, one having clustering and the other one not, when no poisoned clients are present and the data is equally distributed, there are no large differences present.

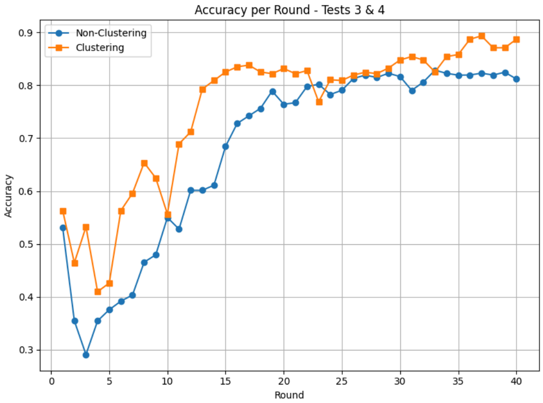

Fig. 5: NPC 1, DPP 100%, DDPC IID

In tests 3 and 4 as seen in **Figure 5** a single poisoned client was introduced which greatly affected the early training stages of the baseline model, after round 20 the 2 models had a similar performance but the clustering had slightly higher accuracy until the end point.

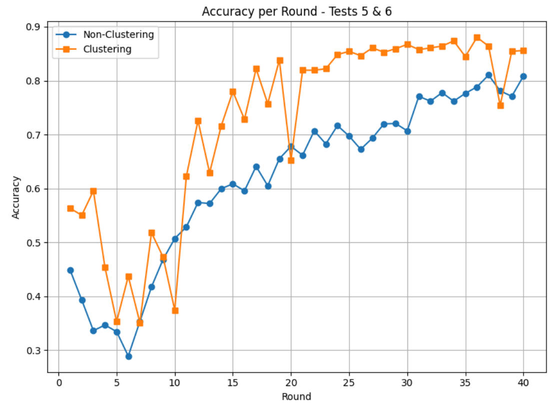

Fig. 6: NPC 3, DPP 100%, DDPC IID

In **Figure 6** the number of poisoned clients was increased to 3 and that's the accuracy improvement for dynamic clustering was clearer.

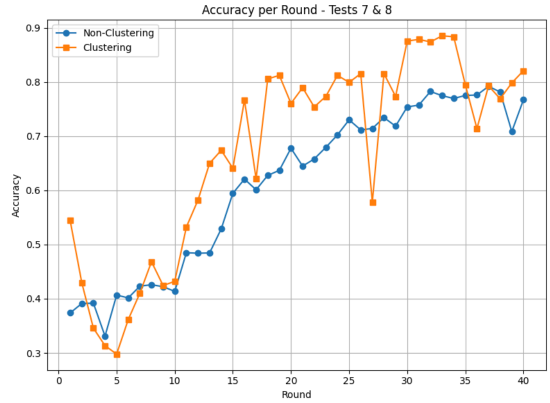

Fig. 7: NPC 5, DPP 100%, DDPC IID

In **Figure 7** the number of poisoned clients was set to 5 and the clustering model seemed to struggle in the beginning but at around round 10 it had more rapid improvements and had a higher accuracy then the baseline model.

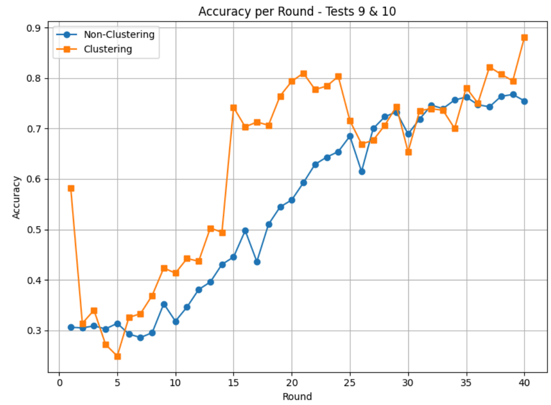

Fig. 8: NPC 7, DPP 100%, DDPC IID

In **Figure 8** once the number of poisoned clients reached 7 the performance difference of the clustering seemed to still have a visible improvement.

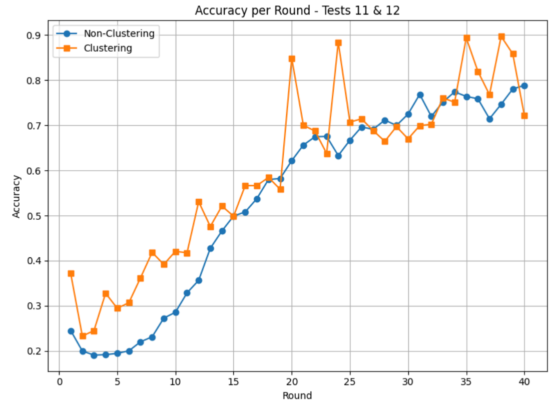

Fig. 9: NPC 9, DPP 100%, DDPC IID

The final test on the number of poisoned clients being at 9 as seens in **Figure 9** showed that the clustering still outperformed the non-clustering version but to a smaller degree.

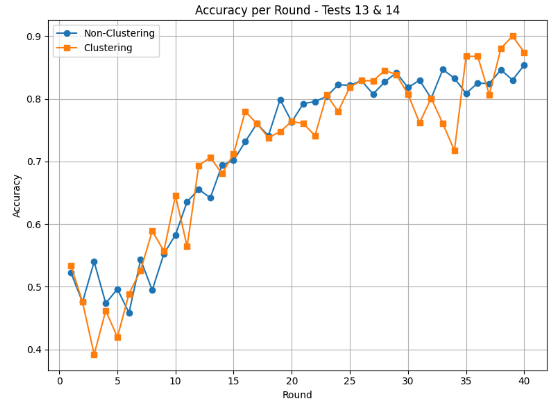

Fig. 10: NPC 5, DPP 25%, DDPC IID

In tests 13 and 14 as seen in **Figure 10** the experiments were switched to check the effects of DPP with fixed NPC(5), at 25% both models seemed to perform roughly the same.

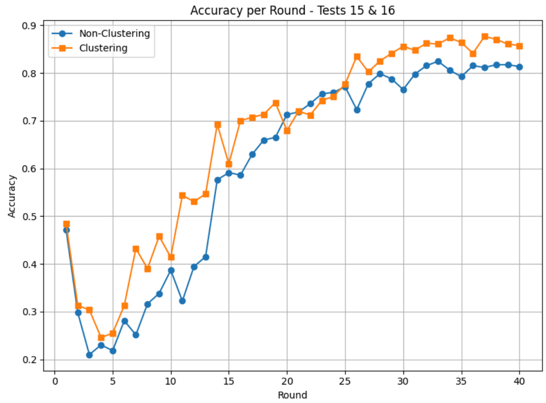

Fig. 11: NPC 5, DPP 50%, DDPC IID

In **Figure 11** at 50% data poisoning the models performed very similarly but clustering still had a minor advantage over the non-clustering model but it has increased when compared to 25%.

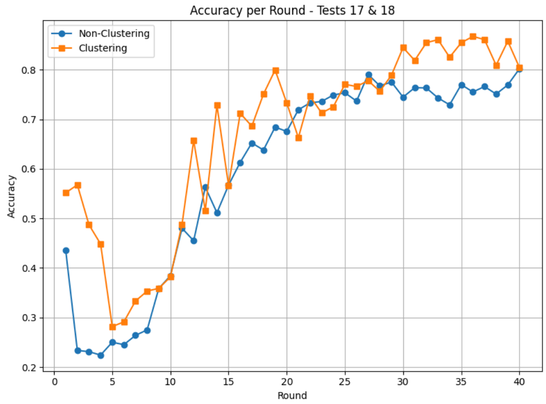

Fig. 12: NPC 5, DPP 75%, DDPC IID

In **Figure 12** throughout the training the performance of both models was similar except jumps of the clustering model between rounds 10 and 20 but at around round 29 the clustering model started to outperform the base model more consistently.

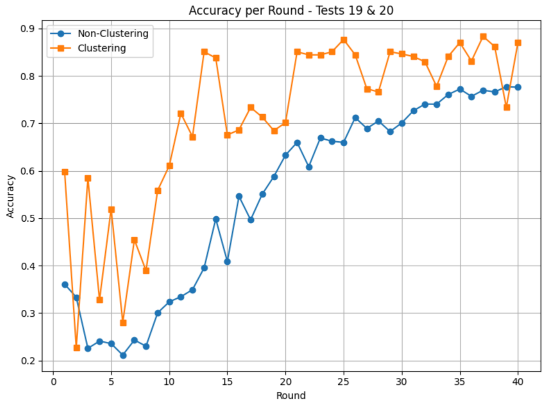

Fig. 13: NPC 5, DPP 100%, DDPC Non-IID

In **Figure 13** the accuracy of the clustering model was better nearly throughout the whole training process, if compared to **Figure 7** which had a similar scenario but with evenly split data, showing that clustering did help when the data is Non-IID.

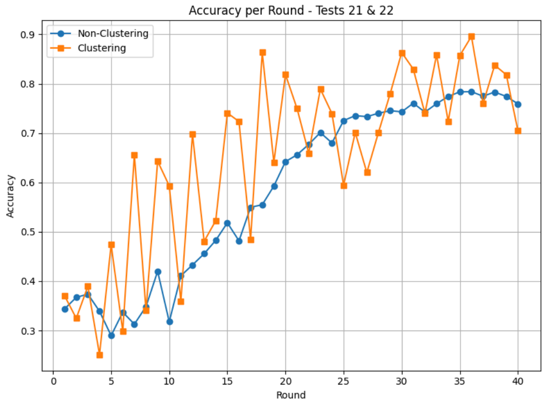

Fig. 14: NPC 7, DPP 100%, DDPC Non-IID

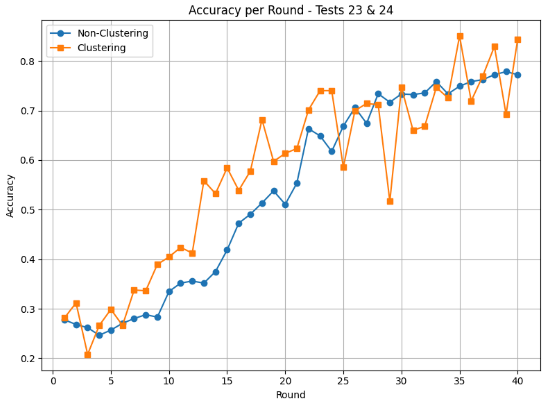

Fig. 15: NPC 9, DPP 100%, DDPC Non-IID

In **Figure 14** and **Figure 15** the accuracy of the clustering model does reach higher peaks but due to the poisoning and Non-IID data the stability was not as good as the non-clustering model.

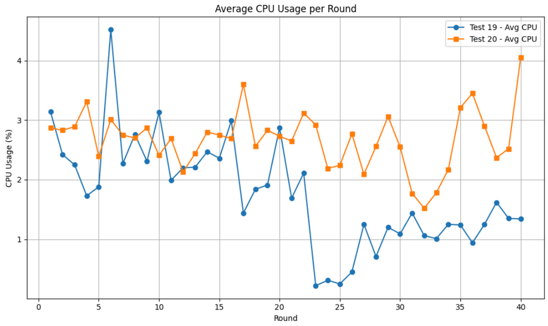

Fig. 16: Average CPU usage for tests 1 & 2

Maximum average usage for test 1: 4.52%

Maximum average usage for test 2: 4.05%

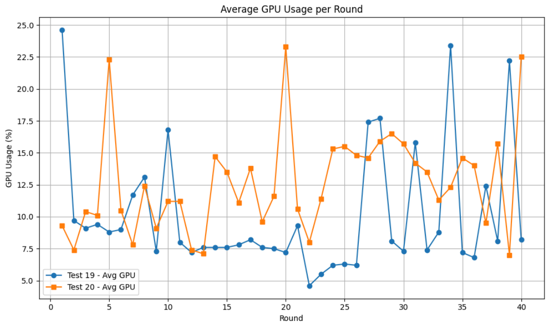

Fig. 17: Average GPU usage for tests 1 & 2

Maximum average usage for test 1: 24.60%

Maximum average usage for test 2: 23.30%

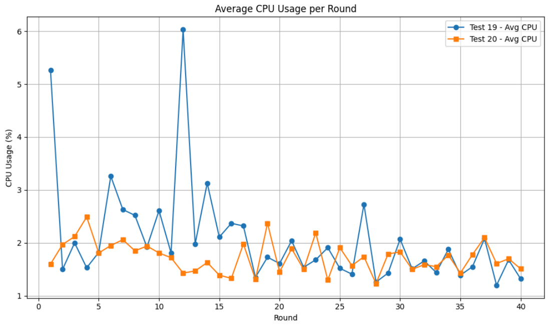

Fig. 18: Average CPU usage for tests 23 & 24

Maximum average usage for test 23: 6.03%

Maximum average usage for test 24: 2.49%

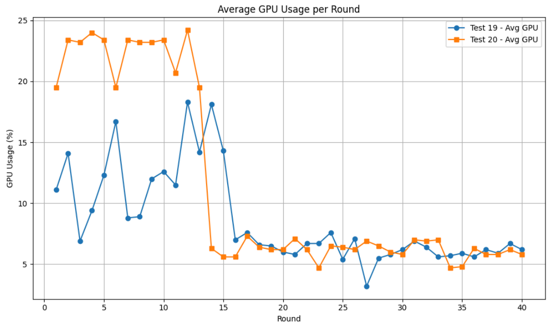

Fig. 19: Average GPU usage for tests 23 & 24

Maximum average usage for test 23: 18.30%

Maximum average usage for test 24: 24.20%

Based on **Figures 16-19** which showed the simplest scenario and the most complex we measured the highest average for the respective training to see how much processing power it would take, for CPU the highest being 6.03% and GPU 24.60%. During the testing the CPU used was AMD Ryzen 7 7800X3D so the result is **2.026 GHz** for CPU and GPU is RTX 4070 Ti which resulted in **9.85 TFLOPS**.

Test	Accuracy	F1 Score	Log Loss
1	0.8606	0.9228	0.0345
2	0.8738	0.9316	0.0094
3	0.8285	0.8929	0.1693
4	0.8935	0.9408	0.0157
5	0.8104	0.8857	0.1505
6	0.8803	0.9372	0.0081
7	0.792	0.8717	0.1392
8	0.8853	0.9384	0.0162
9	0.7674	0.8604	0.1388
10	0.8806	0.93	0.0905
11	0.7884	0.8721	0.0833
12	0.8968	0.9489	0.0164
13	0.8538	0.9219	0.0295
14	0.9	0.948	0.0047
15	0.8247	0.8933	0.0878
16	0.877	0.9363	0.0177

17	0.8017	0.8781	0.1304
18	0.8673	0.9247	0.0291
19	0.7771	0.8623	0.1856
20	0.8831	0.9315	0.0329
21	0.7839	0.8688	0.1369
22	0.8961	0.9456	0.0173
23	0.7787	0.8666	0.0925
24	0.8505	0.9118	0.0569

Table 2: Numerical test results

FML Type	Average Accuracy	Average F1 Score	Average Log Loss
Base	0.8056	0.88305	0.11486
Clustering	0.88202	0.93548	0.0262

Table 3: Average numerical test results

5.2 Analysis

From the conducted tests analyzing the effects of varying the number of poisoned clients (NPC) under a fixed data poisoning percentage (DPP), clustering demonstrated noticeably better performance compared to the baseline model until reaching seven clients. At this point, the accuracy became comparable to the baseline, and at nine clients, clustering showed a slight decline in performance.

In tests examining the effects of varying DPP with a fixed NPC (set at 5), the results indicated no significant impact at 25% poisoning. At 50% poisoning, the difference was minor, while at 75% poisoning, clustering demonstrated slightly better performance compared to the baseline.

For Non-IID data tests conducted at NPC values of 5, 7, and 9, results were consistent with previous observations. Clustering outperformed the baseline in all poisoned cases but the improvements were most noticeable at 3, 5 and 7 clients while 1 and 9 while showing improvements were not as drastic.

Overall, clustering exhibited less stability compared to the baseline model across multiple tests, highlighting an area for potential improvement in future iterations.

For performance testing newer Autonomous vehicles are equipped with high-performance computing systems optimized for real-time decision-making and AI-driven tasks. The CPU in such vehicles, like NVIDIA's DRIVE Orin, offers approximately **200 TOPS** (Trillions of Operations Per Second) with multi-core configurations operating around **2.0–2.2 GHz**. Similarly, the GPU, such as the NVIDIA DRIVE Pegasus, provides around **100–160 TFLOPS** for deep learning inference and advanced perception tasks. When comparing these benchmarks to the calculated CPU usage of **2.026 GHz** and GPU usage of **9.85 TFLOPS**, these values would consume roughly **10–20%** of the CPU's available resources and **6–10%** of the GPU's processing capacity. This demonstrates that the computational demands of the given workload are well within the capabilities of modern autonomous vehicle systems.

6. Discussion

6.1 Interpretation of Results

The findings from this study demonstrate that performance-based dynamic clustering can effectively decrease the impact of data poisoning in FML (FML) systems. The results showed that clustering approaches maintained higher model accuracy compared to baseline models in scenarios with moderate levels of poisoned data and even Non-IID distributions. However, as the number of poisoned clients increased beyond a certain threshold, the benefits of clustering diminished, indicating a limit to its effectiveness with a set number of clusters. The observed instability of clustering in some scenarios highlights the need for further refinement of the grouping mechanism to enhance stability and consistency. Overall, these results support the research hypothesis that clustering can reduce the impact of data poisoning while offering actionable insights into its limitations in specific edge computing scenarios.

6.2 Implications

This research advances the field by presenting a practical framework to mitigate data poisoning in FML systems operating on resource-constrained edge devices. By employing dynamic clustering grounded in performance metrics, the study attempts to offer an approach that combines computational efficiency with robust defenses against adversarial attacks. The adoption of MobileNetv3 highlights the practicality of deploying efficient machine learning models in real-time applications, particularly autonomous vehicles, where managing resource limitations is paramount. Furthermore, the research provides critical insights into the dynamics of FML systems under various adversarial scenarios, laying the groundwork for future advancements in securing and enhancing the reliability of distributed learning systems.

6.3 Limitations

Several challenges and limitations were encountered during this research. First, the experiments were conducted in a simulated environment, which, while useful for controlled testing, does not fully replicate the complexities of real-world edge deployments. Second, the scope of the study was limited to image classification tasks, which may not generalize to other machine learning applications. Third, the clustering approach, while effective in many scenarios, exhibited instability as the number of poisoned clients increased, indicating the need for additional refinement in the clustering algorithm. Lastly, the reduced dataset size used for testing, though practical for simulation purposes, may not fully capture the scalability challenges that arise with larger datasets in real-world deployments.

7. Conclusion

7.1 Summary

This study explored the application of dynamic clustering as a mitigation strategy against data poisoning in FML (FML) systems deployed on edge devices, with a focus on autonomous vehicles. The tests conducted revealed that clustering based on performance metrics can reduce the impact of poisoned updates in multiple scenarios by 9.05% on average or 13.74% in best case scenario. The use of MobileNetv3 as the selected model demonstrated the feasibility of deploying lightweight and resource-efficient machine learning solutions for real-time environments. Computational analysis confirmed that the proposed approach operates well within the capabilities of modern automotive-grade CPUs and GPUs, reinforcing its practicality for real-world applications. These results underscore the potential of dynamic clustering to enhance the security and robustness of FML systems.

7.2 Recommendations

Future research should focus on optimizing clustering methods to enhance their adaptability and efficiency in diverse and dynamic environments. Specifically, the selection of models from the clusters should be refined to ensure that the most representative and reliable updates are incorporated into the global model. Additionally, extending the clustering approach to go beyond a fixed number of three clusters, enabling it to dynamically adapt to the number of participating clients, can further improve performance and scalability. These advancements would allow clustering to better handle varying client behaviors and adversarial conditions, ensuring consistent global model reliability.

7.3 Closing Remarks

This study demonstrates the importance of securing FML systems against adversarial threats while maintaining computational efficiency. By addressing vulnerabilities in edge-based learning, this research lays the groundwork for more robust, scalable, and adaptable distributed learning systems. The proposed methodologies mark a step toward achieving secure and efficient machine learning applications in autonomous vehicles and other edge environments, emphasizing the need for continued innovation in this evolving field.

8. References

1. Saylam, B., & İncel, Ö. D. Federated Learning on Edge Sensing Devices: A Review. arXiv, 2023, Page 3.
2. Abreha, H. G., Hayajneh, M., & Serhani, M. A. Federated Learning in Edge Computing: A Systematic Survey. Sensors, 2022, Page 3, Lines 10–14.
3. Brecko, A., Kajati, E., Koziorek, J., & Zolotova, I. Federated Learning for Edge Computing: A Survey. Applied Sciences, 2022, Page 5, lines 23-28.
4. Hu, K., Gong, S., Zhang, Q., et al. An Overview of Implementing Security and Privacy in Federated Learning. Artificial Intelligence Review, 2024, Page 12, lines 5-14.
5. Li, H., Ge, L., & Tian, L. Survey: Federated Learning Data Security and Privacy-Preserving in Edge-Internet of Things. Artificial Intelligence Review, 2024, Page 7, lines 12-24.
6. Brecko, A., Kajati, E., Koziorek, J., & Zolotova, I. Federated Learning for Edge Computing: A Survey. Applied Sciences, 2022, Page 21, lines 1-8.
7. Islam, M. S., Javaherian, S., & Xu, F. FedClust: Tackling Data Heterogeneity in Federated Learning through Weight-Driven Client Clustering. ICPP, 2024.
8. Ghosh, A., Chung, J., Yin, D., & Ramchandran, K. An Efficient Framework for Clustered Federated Learning. NeurIPS, 2020, Page 2, lines 10-24.
9. Sattler, F., Müller, K. R., & Samek, W. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints. arXiv, 2019, Page 6, lines 10-28.
10. Hu, K., Gong, S., Zhang, Q., et al. An Overview of Implementing Security and Privacy in Federated Learning. Artificial Intelligence Review, 2024, Page 28, lines 12-28.
11. Kirill Smirnov, Dynamic Grouping in Federated Unsupervised Machine Learning, 2024
12. Abreha, H. G., Hayajneh, M., & Serhani, M. A. Federated Learning in Edge Computing: A Systematic Survey. Sensors, 2022, Page 15, lines 20-35.
13. Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. Proceedings of the IEEE International Conference on Computer Vision (ICCV), Page 5, lines 15-30.
14. Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., de Gusmão, P. P. B., Thakkar, O., & Lane, N. D. (2020). Flower: A Friendly Federated Learning Framework. arXiv.org. https://arxiv.org/abs/2007.14390
15. Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q., V., & Adam, H. (2019, May 6). Searching for MobileNetV3. arXiv.org. https://arxiv.org/abs/1905.02244