SpamBase Dataset

Abbas Zal - Supervised by Professor Thomas Marchioro

Introduction

Federated Learning (FL) is a decentralized machine learning paradigm where multiple clients collaboratively train a global model without sharing their raw data. The real-world application of FL, however, introduces complexities such as data heterogeneity, varying client participation(good and bad client). In this study, i aim to explore and optimize the stability and efficiency of federated learning systems under realistic conditions using two distinct datasets and game-theoretic approaches.

The primary objective of this research is to simulate realistic scenarios in federated learning and investigate whether there exists an optimal strategy to achieve both a globally accurate model and a stable configuration for participating clients. To this end, i used concepts from game theory, including Nash equilibrium and Shapley values, to analyze client behavior and contributions within various federated learning combinations.

The datasets was used in this work:

- A spam-based dataset [1], manually partitioned among 10 clients.

In my simulation, federated learning was carried out under different configurations. i systematically evaluated the global model accuracy and client-specific accuracies to identify the optimal federated learning setup. All possible combinations of client interactions were analyzed, with a focus on achieving a balance between the accuracy of the global model and the stability of participating clients' contributions.

The results of this study provide valuable insights into how federated learning systems can reach a stable and efficient state, i aim to identify strategies that maximize the overall utility for both the system and the individual clients.

1 Spambase Dataset

The Spambase dataset [1], sourced from the UCI Machine Learning Repository, is widely utilized for spam email classification tasks. It consists of 4,601 email instances, with a binary target variable indicating whether an email is spam (1) or not spam (0). Approximately 39.4% of the dataset instances are labeled as spam, and 60.6% as non-spam.1

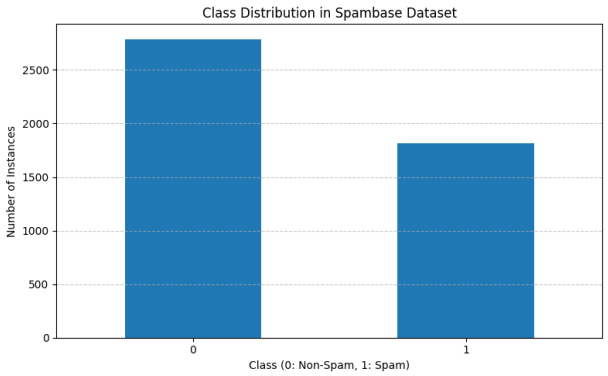

Figure 1:

The dataset includes 57 numerical features derived from analyzing email content and structure. These features can be categorized into three types:

- **Word Frequency Features:** 48 attributes represent the percentage frequency of specific words in the email text.
- **Character Frequency Features:** 6 attributes indicate the frequency of specific characters.
- **Capital Run Length Features:** 3 attributes measure the use of capital letters, including the average, longest sequence, and total count of uninterrupted capital letters.

1.1 General Pipeline Description

The process for preparing and analyzing the Spambase dataset was done in a structured manner as outlined below:

Data Preparation: The data was divided into training and testing sets, with 20% of the data reserved for testing. A standardization process was applied to the training data using `scaler.fit_transform`, which scaled the features to have a mean of 0 and a standard deviation of 1.

The same transformation was applied to the test data using `scaler.transform`. After these, now i had train set and test set as a global training set and test set.

1.2 Results of Centralized Learning on the Spambase Dataset

This section presents the result of the Centralized **Linear Logestic Regression** experiment on Spambase dataset, The results of the **Centralized Learning** experiment were evaluated based on the model's **accuracy**. In this approach, all data was aggregated to train a model.

Accuracy (Centralized Learning): *91.96%*

This comparison helps to measure the trade-offs in Decentralized or Federated Learning approaches, which aim to improve privacy and reduce communication costs at the potential expense of accuracy.

1.3 FL Pipeline Description

After the data preparation steps described earlier (see Section 1.3), the process for preparing and analyzing the Spambase dataset for Federated Learning (FL) is outlined below:

Partitioning Data for Federated Learning

The training data was partitioned equally among multiple clients to simulate a federated learning environment. An optional shuffling step was included to randomize the data before partitioning. Each client received an equal-sized subset of the training data.

Federated Model Aggregation

Logistic regression models trained by individual clients were aggregated. The aggregation process involved averaging the model coefficients and intercepts across all client models to create a single federated logistic regression model.

Federated Learning Experiment

To evaluate the performance of the FL system, the experiment considered all possible configurations of client participation. With 10 clients available, every subset of clients was evaluated to form a federated learning group. This resulted in $2^{10} = 1024$ distinct configurations. However, the configuration where no clients participated was excluded, leaving $1024 - 1 = 1023$ configurations to analyze. These configurations ranged from individual clients training their models independently to all 10 clients collaborating in the federated learning setup.

Methodology for Exploring All Configurations

To systematically analyze the performance across all client combinations, the following approach was employed:

1. **Binary Representation of Configurations:** Each configuration was represented as a binary string of length n, where n is the number of clients. A binary digit of '1' indicated that the corresponding client participated in the configuration, while '0' indicated exclusion.
2. **Iterative Analysis:** Starting with the binary representation of '0000000001' (only one client participating- Client 1-) and iterating up to '1111111111' (all clients participating), all valid combinations were evaluated.
3. **Model Aggregation:** For each configuration, the models from the participating clients were aggregated using the `aggregate_lr_models` function(see Section 1.3).
4. **Accuracy Evaluation:** The aggregated model was evaluated on the test data, and its performance was recorded as Global Accuracy. Additionally, the individual accuracy of each participating client model was computed on the global test set for comparison as Client Accuracy.

Simulating Real-World Scenarios

To simulate real-world scenarios, a custom function was introduced during the training process. This function selects a specified number of clients and deliberately introduces noise into their datasets. The noise degrades the quality of their data, reflecting realistic conditions where clients may have unreliable or lower-quality data. This simulation helps evaluate the robustness of the federated learning system under such challenges.

Noise Injection Mechanism

The core mechanism for simulating data corruption is described as follows:

- The **corrupt_data** function introduces two types of noise:

1. **NaN Injection:** Some entries in the dataset are replaced with `NaN` values to simulate missing data.
2. **Additive Noise:** Random noise with a specified standard deviation (*noise_std*) is added to a subset of the dataset.

- The level of corruption is controlled by *corruption_prob*, which defines the proportion of data points affected.
- An additional parameter, *nan_prob*, determines the likelihood of introducing `NaN` values within the corrupted data.

Corrupting Client Data

A secondary function was implemented to corrupt the datasets of specific clients:

- A random selection of clients was chosen based on the number of clients to corrupt (*n_corrupt_clients*).
- For each selected client, their dataset was passed to the *corrupt_data* function, and the corrupted datasets replaced the original ones in the partitions.

This process ensures that the experimental setup mimics real-world scenarios where data quality varies across clients, allowing for an evaluation of the federated learning model's robustness in the face of such challenges.

1.4 Methodology of Analysis

To evaluate the performance and contribution of clients in the federated learning setup, several analytical methods were applied. These methods are detailed below:

Client Performance Evaluation

- The best and worst clients were identified based on their local accuracies on their own test set.
- The impact of individual clients on the global accuracy was assessed by calculating the mean, standard deviation, minimum, and maximum global accuracies, both with and without the inclusion of each client.

Federated Model Performance

- The best global accuracy achieved by each client was identified by analyzing configurations where the client participated. Similarly, the worst global accuracy was determined for each client by evaluating configurations that included them.
- The contribution of clients to the best and worst-performing federated configurations was quantified by counting their frequency in these configurations.

Robustness and Contribution Analysis

- Client contributions to the overall model were analyzed using Shapley values, a fairness metric that quantifies each client's marginal contribution to the global accuracy across all possible combinations.
- The Shapley values were normalized and compared with the different local accuracies of the clients on global test set and local test set to explore their alignment and disparities.

Game Theory Insights

- Nash equilibria were identified by analyzing configurations where no client had an incentive to deviate unilaterally. This helped evaluate the stability of federated learning configurations.

Impact of Client Inclusion and Exclusion

- The mean global accuracies were compared for each client, both when they were included in and excluded from the federated learning process. The differences between these accuracies were calculated and visualized to highlight the impact of each client.

2 Results of Decentralized Learning (FL) on Spambase

This section presents the results of the federated **Linear Logestic Regression** experiment on Spambase dataset, where the robustness, client contributions, and overall system performance were evaluated under various configurations. For example for 1, 2,3,...,8,9 corrupted clients. Now i will discuss about some of them.
The **First experiment** involved 10 clients, with 1

of them being corrupted **(client 9)**. The following insights were derived:

Top Configurations

The top-performing configurations in terms of global model accuracy are summarized in Table 3. The highest global accuracy achieved was 0.935939, observed in configurations where specific combinations of clients (e.g., [1, 3, 6, 8, 10]) participated.

Table 1: Top 5 Configurations with Highest Global Accuracy

Combination	Clients	Acc
1010100101	[1, 3, 6, 8, 10]	0.935939
1010101001	[1, 4, 6, 8, 10]	0.934853
1000110110	[2, 3, 5, 6, 10]	0.934853
1000100111	[1, 2, 3, 6, 10]	0.933768
1010110001	[1, 5, 6, 8, 10]	0.933768

Best and Worst Clients

The best client was identified as Client 3, while the worst client was Client 9, based on their local accuracies on their own test set.

Client Contributions

Figures 5 and 6 show the frequency of each client in best and worst-performing configurations per each client, respectively. Clients 6 and 10 frequently contributed to the best-performing combinations, while Clients 9 as expected, (because this is the corrupted clients) were dominant contributors to the worst-performing configurations.

Shapley Values and Fairness Analysis

The Shapley values, which quantify the marginal contribution of each client to the global model, were computed and compared for local accuracy on the global test set. As shown in Figures 4, Client 6 had the highest Shapley value (0.1124), indicating a substantial contribution to the global model, **aligning well with its high local accuracy (0.9164) on the global test set**.

Nash Equilibria Analysis

The Nash equilibria were analyzed to identify stable federated learning configurations. Table 4 summarizes the results, showing configurations where no client had an incentive to unilaterally change their participation.

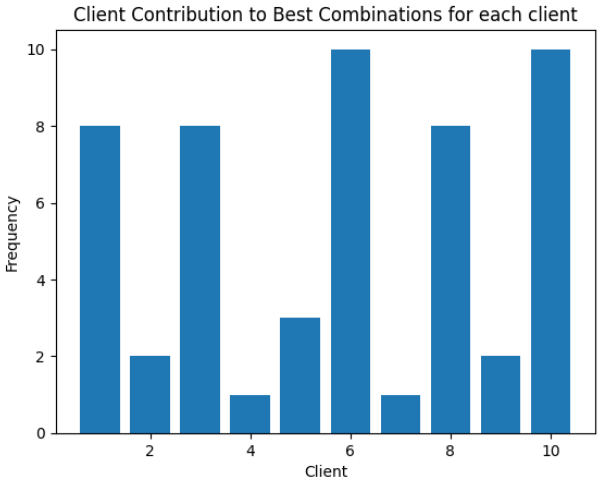

Figure 2: Client Contribution to Best Combinations

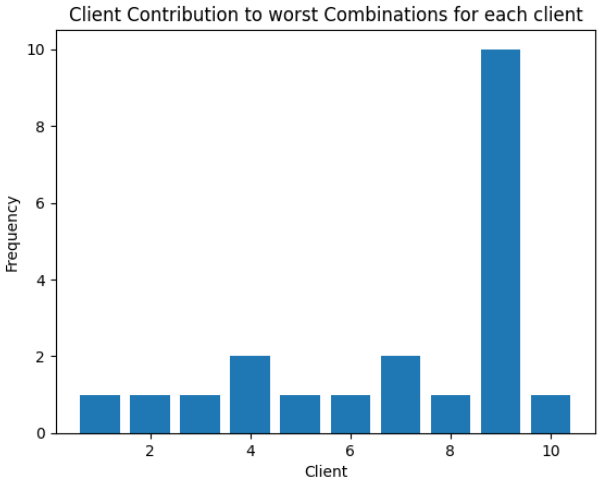

Figure 3: Client Contribution to Worst Combinations

Table 2: Nash Equilibria Configurations

Combination	Clients	Acc
0100000000	[9]	0.669924
1111111111	[1, ..., 10]	0.926181

Now, i tried to test my result with more corrupted clients, The **Second experiment** involved 10 clients, with 5 of them being corrupted **(Clients 6, 3, 5, 10, 2)**. The following insights were derived:

Top Configurations

The top-performing configurations in terms of global model accuracy are summarized in Table 3. The

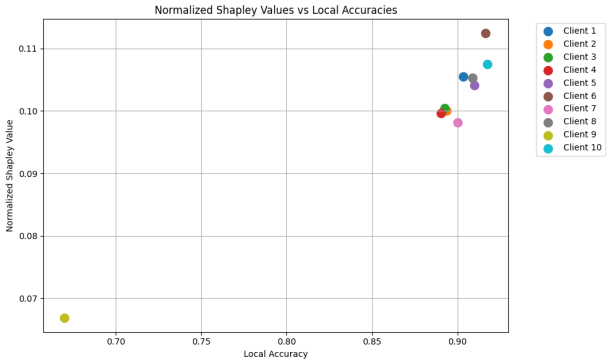

Figure 4: Normalized Shapley Values vs Local Accuracies on global Test set

highest global accuracy achieved was 0.927253, observed in configurations where specific combinations of clients (e.g., [1, 4, 8]) participated.

Table 3: Top 5 Configurations with Highest Global Accuracy

Combination	Clients	Acc
0010001001	[1, 4, 8]	0.927253
0000001001	[1, 4]	0.921824
0011000011	[1, 2, 7, 8]	0.920738
0011011011	[1, 2, 4, 5, 7, 8]	0.919653
0011001001	[1, 4, 7, 8]	0.919653

Best and Worst Clients

The best client was identified as Client 8, while the worst client was Client 6, based on their local accuracies on their own test set.

- With Client 8 included, the global accuracy had a mean of 0.9008 (standard deviation: 0.0089).
- In contrast, when Client 6 was included, the global accuracy mean dropped slightly to 0.8868 (standard deviation: 0.0319).

Client Contributions

Figures 5 and 6 show the frequency of each client in best and worst-performing configurations per each client, respectively. Clients 1 and 8 frequently contributed to the best-performing combinations, while Clients 6, 3, 2, 10 and 5, as expected, (because they are the corrupted clients) were dominant contributors to the worst-performing configurations.

Figure 5: Client Contribution to Best Combinations

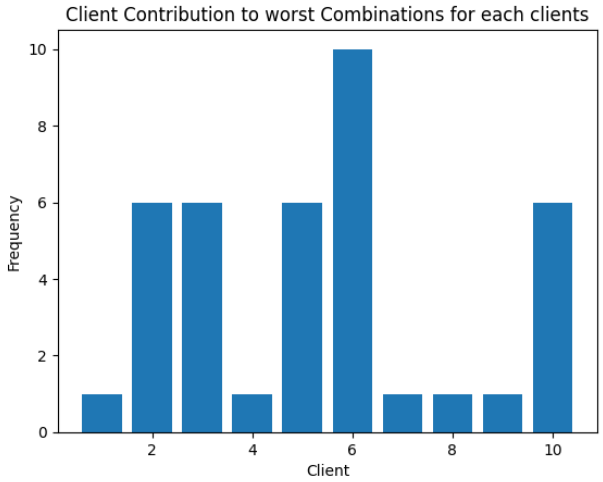

Figure 6: Client Contribution to Worst Combinations

Shapley Values and Fairness Analysis

The Shapley values, which quantify the marginal contribution of each client to the global model, were computed and compared for local accuracy on the global test set. As shown in Figures 7, Client 1 had the highest Shapley value (0.1288), indicating a substantial contribution to the global model, **aligning well with its high local accuracy (0.9034) on the global test set**.

Nash Equilibria Analysis

The Nash equilibria were analyzed to identify stable federated learning configurations. Table 4 sum-

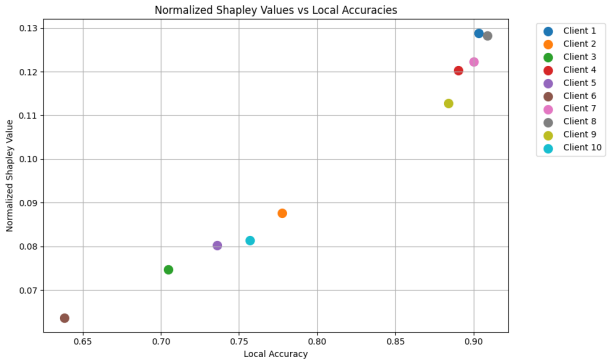

Figure 7: Normalized Shapley Values vs Local Accuracies on global Test set

marizes the results, showing configurations where no client had an incentive to unilaterally change their participation.

Table 4: Nash Equilibria Configurations

Combination	Clients	Acc
0000100000	[6]	0.638436
0000110100	[3, 5, 6]	0.738328

Now, i tried to test my result with 9 corrupted clients,The **third experiment** involved testing with 10 clients, where 9 of them were corrupted **(Clients 2, 3, 5, 6, 7, 8, 9, 10, and 4)**. The analysis highlights the performance and stability of configurations under this challenging setup.

Top Configurations

The top-performing configurations based on global accuracy are summarized in Table 5. The highest global accuracy achieved was **0.903366**, observed when only **Client 1** participated.

Table 5: Top 5 Configurations with Highest Global Accuracy

Combination	Clients	Accuracy
0000000001	[1]	0.903366
0000000011	[1, 2]	0.903366
0010000001	[1, 8]	0.899023
1000000011	[1, 2, 10]	0.896851
0000000111	[1, 2, 3]	0.895765

Best and Worst Clients

The best client was identified as **Client 1**, while the worst client was **Client 9**, based on their local accuracies and contributions to global accuracy:

- Including **Client 1** resulted in a mean global accuracy of 0.8694, with a standard deviation of 0.0114.
- Including **Client 9** resulted in a mean global accuracy of 0.8146, with a higher standard deviation of 0.0516.

Client Contributions

Figures 8 and 9 illustrate the frequency of clients in the best- and worst-performing configurations, respectively. **Client 1** was the most frequent contributor to the best-performing combinations.

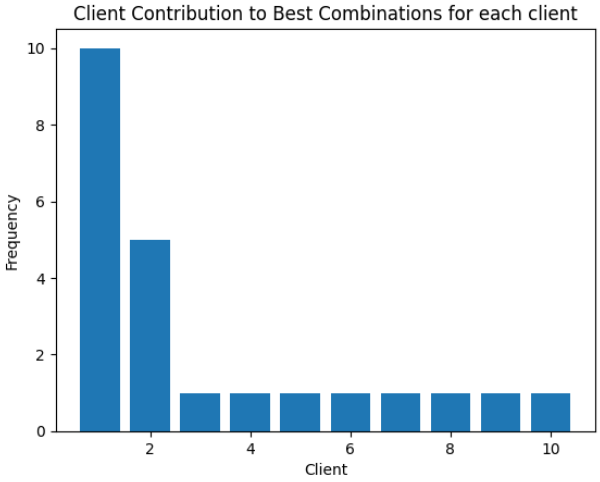

Figure 8: Client Contribution to Best Combinations

Shapley Values and Fairness Analysis

Shapley values were computed as in previous experiments. Figure 10 illustrates that **Client 1** had the highest Shapley value (0.2214), aligning with its strong local accuracy (0.9034) on the global test set. In contrast, **Client 5** had the lowest Shapley value (0.0681), reflecting its weak contribution.

Nash Equilibria Analysis

The Nash equilibria configurations are presented in Table 6.

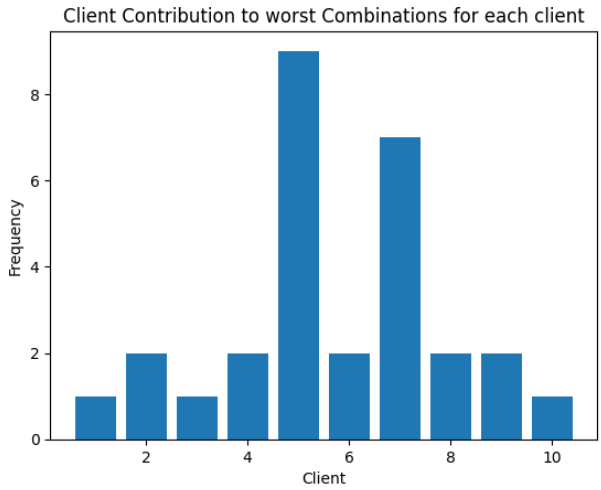

Figure 9: Client Contribution to Worst Combinations

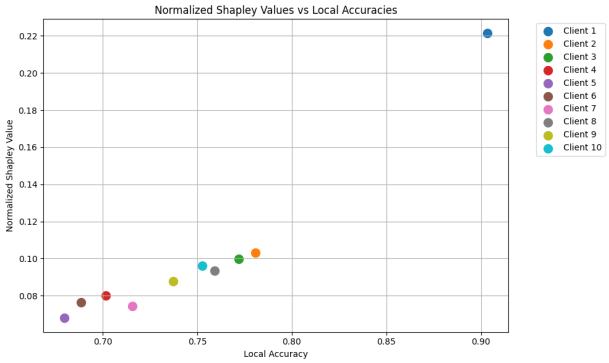

Figure 10: Normalized Shapley Values vs. Local Accuracies on the Global Test Set

Table 6: Nash Equilibria Configurations

Combination	Clients	Accuracy
0101111000	[4, 5, 6, 7, 9]	0.744843

3 Conclusion

In this section, I aim to highlight a key insight based on the results obtained from two different datasets analyzed using three different models. It is worth noting that while the findings from the Spambase dataset are aligned with my overall conclusion, the results may not be fully authoritative. This is because the Spambase dataset is linearly separable, which complicates deriving robust conclusions. By the way, I have observed nearly identical patterns and confirmed conclusions with this dataset as well.

Appearance in the Best-Performing Models

The number of appearances for the best clients was higher than that of other clients. Specifically, clients with high local accuracies on their own test sets appeared more frequently in the best-performing combinations for other clients.

Appearance in the Worst-Performing Models

Similarly, the number of appearances of the worst clients was higher than that of other clients in the worst-performing combinations. Clients with low local accuracies on their own test sets appeared more often in the worst-performing combinations for other clients.

Shapley Values

Shapley values help measure how much each client contributes to the performance of the entire federated system. Think of it like figuring out how much each team member helped achieve a group goal.

Low-Accuracy Clients Contribute Less

- Clients that perform poorly on the global test data consistently have the lowest Shapley values.
- This means that if a client has poor-quality data or a weaker model, it doesn't help the overall system much.

The Best-Performing Client Might Not Be the Most Valuable

- The client with the highest accuracy on its own data doesn't always have the highest Shapley value.
- **Why?** Because its performance might be very specific to its own data and not generalizable.

Who Consistently Has the Highest Shapley Value?

- The client that does the best on the global test data usually contributes the most to the overall system, as its updates are useful across the board.

Why Doesn't High Local Accuracy Guarantee High Shapley Value?

Specialized Data Doesn't Always Help

- A client's high accuracy on its own test data shows it's good for its specific data.
- But if this data is too unique, its updates might not work well for others.

System Goals Matter More

- The federated system's goal is to work well for everyone.
- A client's value depends on how its contributions align with this objective, not just its own performance.

Nash eq

The analysis of Nash equilibria across varying numbers of corrupted clients provides insights into the dynamics of client behavior:

- **Fewer than half of the clients are corrupted:** When the number of corrupted clients is low , the equilibria exhibit less consistency. The outcomes vary between full participation and partial participation involving some corrupted clients. For example:
  - With **1 corrupted client**, equilibria configurations often involve full participation, such as `1111111111`, or specific configurations like `[9]` (`0100000000`).
  - With **3 corrupted clients**, the equilibria are more diverse, including full participation (`1111111111`) and partial participation configurations like `[5, 9]` (`0100010000`), reflecting variability.
- **More than half of the clients are corrupted:** When corrupted clients exceed half the total, Nash equilibria become more predictable and often involve participation primarily from corrupted clients. For example:
  - With **6 corrupted clients**, he equilibria are dominated by different configurations with corrupted clients participating, such as `[9]` (`0100000000`), `[7, 9]` (`0101000000`), and `[2, 3, 4, 7, 9]` (`0101001110`).
  - With **9 corrupted clients**, the equilibria are dominated by configurations with corrupted clients participating, as seen in `[4, 5, 6, 7, 9]` (`0101111000`).

References

[1] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase [dataset]," 1999.