A Privacy-Preserving Data Valuation Visualization System

Yap Rong Yu

U1823960A

Supervisor:

A/P Han Yu

Examiner:

A/P Lin Guosheng

SCHOOL OF COMPUTER ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY
2021

Abstract

Data is increasingly being regulated by the governments, making it difficult to conduct collaborative machine learning without violating the regulation. This leads to the increased interest in federated learning as data is processed at the client-side. However, stakeholders are hesitant to participate in federated learning. This is due to federated learning producing a huge amount of data as output and thus it is difficult to interpret the results of federated learning.

This leads to a need to have a visualisation system to present data in a manner that the stakeholders can interpret. Current visualisation systems are unable to meet the needs of the stakeholders as they are not able to handle the large data output produced by federated learning.

In this report, Shapley value and its various estimation will be reviewed along with the previous studies of visualisation systems for federated learning. The design and results of the visualisation system will be discussed after the literature review.

Acknowledgement

I would like to thank Dr Liu Zelei has he has been guiding me throughout the Final Year Project. Dr Liu has been patient with me on getting the concept of federated learning and Shapley value right. Adding on, Dr Liu’s insight on designing user interface has been invaluable to me as I do not have much experience in this area.

Table of Content

Table of Content

Table of Figures

1. Introduction

1.1 Background

Data is being increasingly regulated by the government with policies such as General Data Protection Regulation (GDPR) in the European Union [1] [2] and Personal Data Protection Act (PDPA) in Singapore. In a typical collaborative machine learning setting, the participant's data is sent to the server for processing. With these data protection laws, it has become difficult to conduct collaborative machine learning without violating data protection laws.

Federated learning is a collaborative machine learning technique. It can be conducted with data being kept and processed at the participant's machine [1] [2]. Thus, there has been a growing interest in federated learning [3]. In the federated learning process, the participant downloads the model from the federated machine. Data will be processed to update the model in the participant's machine(s). Finally, the updated model is then uploaded back to the federated learning machine [3].

It is expected to have a large number of participants (between 10 to 20) with huge datasets. As a result, training a federated model will lead to huge data output that is difficult for stakeholders to interpret. Thus, there is a need for a visualisation system to make data presentable for the stakeholders.

1.2 Purpose of the Research

The purpose of the research is to explore ways to develop a visualisation system that is simple to use yet robust enough to display the relevant data to the stakeholders. Stakeholders without technical knowledge in federated learning should be able to use this system without issues.

1.3 Scope

This report presents the result of exploring ways to visualize data from federated learning. The importance of this project is to show that data from federated learning can be displayed into a simple to understand visualisation system. The study is done using Python, with libraries such as Matplotlib for data visualization and PyQt 5 for the user interface, respectively. Anomaly detection will not be implemented in this system. However, obvious anomaly in training rounds will be visualised in the graphs.

1.4 Organisation of the Report

First, Shapley value will be reviewed on how it can be used to evaluate the participants' contributions followed and why it is not feasible to use Shapley value for evaluation. Several methods of estimating Shapley value such as Truncated Monte-Carlo, Gradient Shapley and Guided Truncation Gradient Shapley will be discussed right after. The needs and research gap for a visualisation system for federated learning will be discussed at the end of the literature review.

In the System Design section, the chosen architecture design, design decision taken, and lo-fi prototype design will be discussed. Next, the programming language and libraries used will be discussed. Design decisions made to meet the needs of the stakeholders will be discussed in the follow-up. At the end of the System Design section, the lo-fi prototype will be shown.

In the Results section, the purpose and functionality of each view in the visualisation system will be discussed. At the end of the report, a conclusion will be made.

2. Literature Review

2.1 Shapley Value

Shapley Value is a solution to distribute rewards fairly for participants with different levels of contribution in game theory in the maths field. It distributes rewards fairly by distributing rewards for each participant based on their contribution. If a participant did not contribute, the participant should not receive any reward. If participants i and j contributed equally, they both should receive equal rewards [4].

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\;(n-|S|-1)!}{n!}(v(S \cup \{i\}) - v(S))$$

Figure 1.1 Formula of Shapley Value

The whole equation represents that the Shapley value of i (φi(v)) is the summation of the marginal gain of i multiplied by the combination of the coalition i is in for all the coalitions excluding participant i. φi(v) represents the Shapley value of the individual i. S represents the coalition of participants, N represents all the participants, n represent the number of participants. |S|!(|N|−|S|−1)! represents the number of combinations of coalition S before i joined the coalition [5]. Thus, |S|!(|N|−|S|−1)!|S|!(|N|−|S|−1)! / n! represents the combination of i can join the coalition [5]. v(S∪{i})−v(S)) represents the contribution of player i if player i joins the coalition after all players in S has joined [5].

2.2 Using Shapley Value in Federated Learning

Shapley value is used for the distribution of rewards in federated learning as it solves the problem of distributing rewards fairly in consideration of expected uneven contribution. In federated learning, the marginal gain would be the performance increase of the federated model [6]. The derived Shapley value will be used to evaluate the contributions of the participants.

Shapley value being able to evaluate the participants based on the marginal gain on the federated model by the participant's data is another reason to use Shapley value. Currently, participants are evaluated based on the quality and size of the data they provided. Due to the data regulations, accessing the participant's data by the federation owner for evaluation using current methods may not be possible without violating data regulations.

Using Shapely Value to determine the contribution of participants is costly. If Shapely Value is used, participants would have to train the federated model for every permutation of the participants. The resulting time complexity is O(2^n), growing exponentially based on the number of participants. This is an issue as at the business level, it is expected to have between 10 to 20 participants. As a result, training the federated models multiple times is costly and time consuming to the participants.

2.3 Estimation of Shapley Value

Shapley Value can be estimated by sampling the participant's contribution. By doing an estimation of Shapley Value, it significantly reduces required resources to evaluate the participants' dataset as the time complexity reduces from exponential to polynomial. The proposed solutions of estimating Shapley Value are Truncated Monte-Carlo, Gradient Shapely and Guided Truncation Gradient Shapley.

2.3.1 Truncated Monte-Carlo

Monte-Carlo is a method of estimating by sampling random data points. The application of Monte-Carlo in Shapley Value is to estimate the Shapley Value of a participant by sampling the dataset randomly. Marginal gain is calculated at the end of every round. The process is further improved by truncating the participant's marginal gain and set to 0 when it is below a given threshold in each round. That participant and the remaining participants will be excluded in the computation of Shapley Value [6]. Shapley Value will be calculated at the end by averaging the marginal gain of the participants. The truncation cuts down the time training the model when the marginal of the participant(s) becomes very low.

Truncated Monte-Carlo (TMC) reduces the number of training by the participants. However, participants are still required to train the model multiple times beyond the first. Randomly sampling the dataset may also cause unfairness by placing a certain participant in front or behind frequently in the training rounds. It is unfair to place a participant in front or behind frequently in the training rounds as the position in the training rounds affects the contribution value of the participant. Participants who are the first few to train the model generally have a higher contribution value than the last few.

2.3.2 Gradient Shapley

Gradient Shapley is another Shapley Value estimation method. When a participant trains the federated model, the participant's gradient update to the model in each round is stored. The gradients then can be used to construct every permutation of the sub-models for evaluation [2] [7]. Thus, no extra training of the model beyond the first is required and adding new participants would be relatively easy.

This concept is used for One Round Reconstruction based Algorithm (OR) and Multi-Rounds Reconstruction based Algorithm (MR). For OR, gradients are recorded only in the final round for model reconstruction. Sub-models are reconstructed using gradients from the last rounds. The participants will be evaluated based on the average contribution towards the sub-models. OR has the fastest execution speed compared MR and TMC [2]. It is due in OR, it only needs to record a single round of gradients to reconstruct models for evaluation. In the best-given scenario, OR will accurately predict Shapley Value [2]. However, OR is very susceptible to noise [2] as one round of gradient is used for reconstruction.

For MR, gradients are recorded every round during the training, MR requires more time compared to OR due to needing to record gradients in each round. However, it is still much faster than TMR as participants only need to train the model once. In the sub-model reconstruction process, the participant's contributions are evaluated every round in each sub-model reconstruction instead of using only the last round. As a result, MR will take a longer time to evaluate a participant's contribution as compared to OR. In conclusion, MR's execution speed is in between OR and TMR [2]. MR is more robust and less susceptible to noise compared to OR as it has multiple rounds of gradients to reconstruct the model.

2.3.3 Guided Truncation Gradient Shapley (GTG – Shapley)

GTG – Shapley uses the MR approach for the recording of the gradients every round and then reconstruct models for evaluation [7]. GTG – Shapley defers from MR in the model reconstruction process as GTG – Shapley uses the guided truncation to determine the sub-models to be reconstructed. The permutation of the first few positions is fixed to allow every participant to have an equal chance to be at the front of the training process. The rest of the positions are randomly assigned. First, it is the between-round truncation is performed. If the marginal gain of the round is less than the defined threshold, the round is ignored and the contribution of every participant of that round is set to 0. The contributions of the rounds after the round below the defined threshold will not be calculated and is set to 0 too. Afterwards, the within-round truncation will be performed. When a participant's marginal gain is below a defined threshold, further marginal gain calculations for that round will stop, that participant's contribution and the remaining participant's contribution is set to 0. With the truncations, it reduces the overall resources needed for model reconstruction [7].

2.4 Visualisation System

The visualisation system designed will be taking in output from the selected Shapley algorithm. In this research, GTG – Shapley is the selected algorithm. Federated learning produces a large amount of data and visualising all these data causes clutter [3]. Thus, making it difficult for stakeholders to interpret. For the visualisation system to be user friendly for relevant stakeholders, the system must only display relevant data to the stakeholders. The visualisation system will also need to display obvious anomaly in the federated learning so to allow the stakeholders to investigate the cause of anomaly.

2.5 Research Gap

Currently, there are several visualisation systems to visualise data produced by federated learning [3]. However, these systems did not account for the distributed nature of federated learning. Thus, unable to handle a large number of clients [3].

3. System Design

3.1 System Architecture

The designed visualisation system uses layered architecture as a reference. The layers are the database layer, data access layer, logic layer and user interface (UI) layer. The database layer is simulated by a pickle file containing data frames to represent a database with tables. The data access layer is for accessing data from the database layer and passing the data to logic for processing. The UI layer will be taking in data from the logic layer and displaying it to the user.

The pickle file used to simulate a database is named dbs_v2.pkl. In it, output from GTG – Shapley formatted in tables: federation, fed_round_info, sv_info, participants, p_participate and fpid_to_pid. In the list below, it shows the information stored in the tables.

- federation table: Stores the federation id and description.
- fed_round_info: Stores the datetime, accuracy, loss and participants involved in the training rounds of the federated model.
- sv_info: Store the accuracy of all the sub-models for every training round of the federated model.
- participants: Stores the mapping of unique participant id to the federated model.
- p_participate: Stores the Shapley value of the participants for every training round of the federated model.
- fpid_to_pid: Stores the mapping of unique participant id to participant in the federation.

The data access layer consists of several data access objects (DAO), each for each view. The purpose of having a data access layer is to retrieve data from the database, then return data requested by the logic layer. By having a data access layer, it prevents the logic and UI layer from directly accessing the database. This is to prevent illegal calls to the database, preventing errors and illegal retrieval of data.

The logic layer consists of several controllers, each for each view except FederationView. The purpose of having a logic layer is to retrieve data from the data access layer, handle the data processing and returns the data to be displayed to the UI layer. With a logic layer, changes to logic will affect the data access layer and UI layer minimally.

The FederationView does not have a controller as too much data moves between the logic and UI section. The UI in FederationView is dynamic as it displays data based on user selection. When the user makes a selection, FederationView will request and process a large amount of data. UI is then generated dynamically to display the data. Thus, it is challenging to design a controller for FederationView and a decision is made to combine logic and UI for FederationView

Finally, the UI layer consists of views and UI classes. The views are the classes of the screens that are responsible for taking in data from the logic layer, displaying the data and providing an interactive UI. The UI classes are additional classes of input dialogue box and table, used by the views.

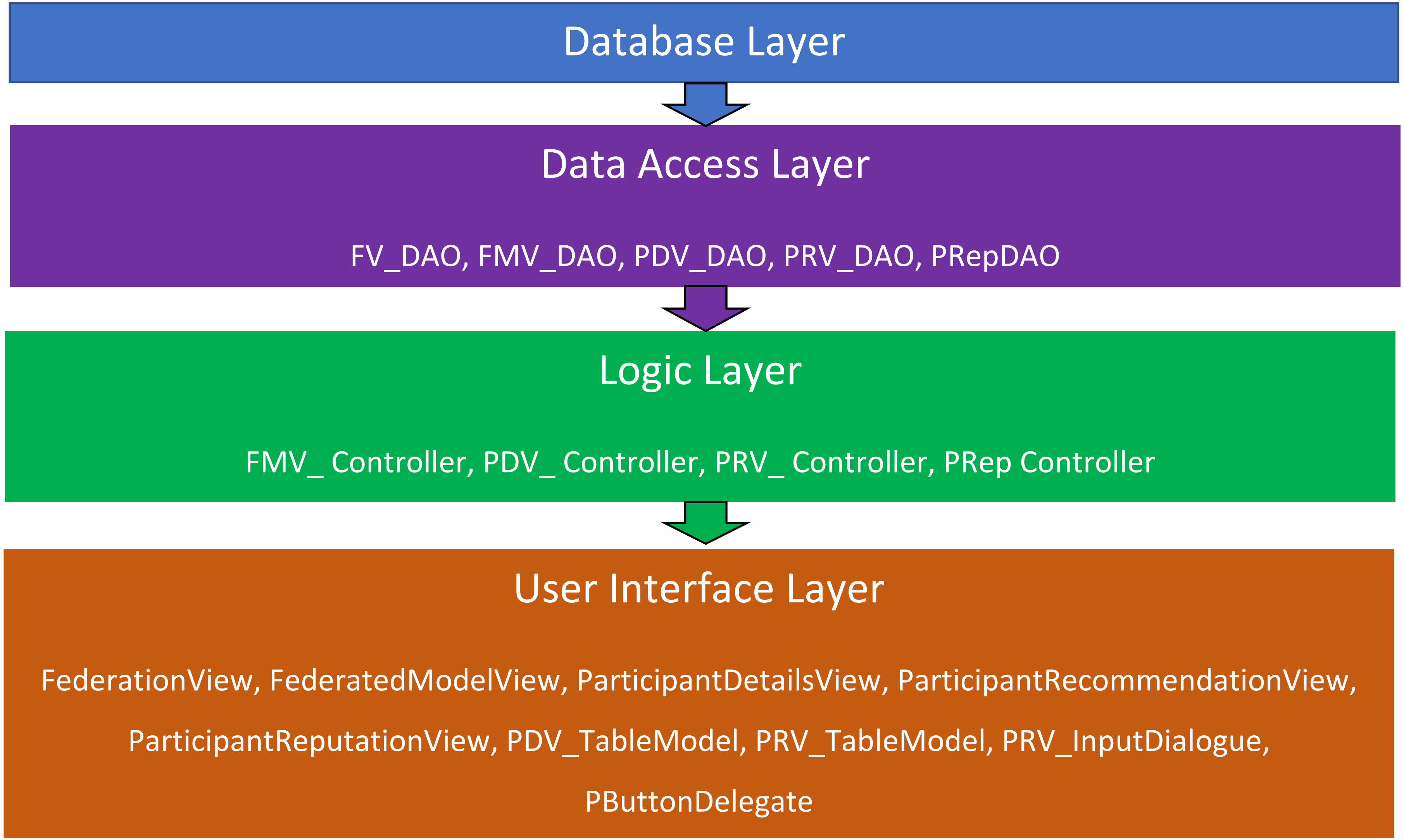


Figure 3.1 Architecture diagram

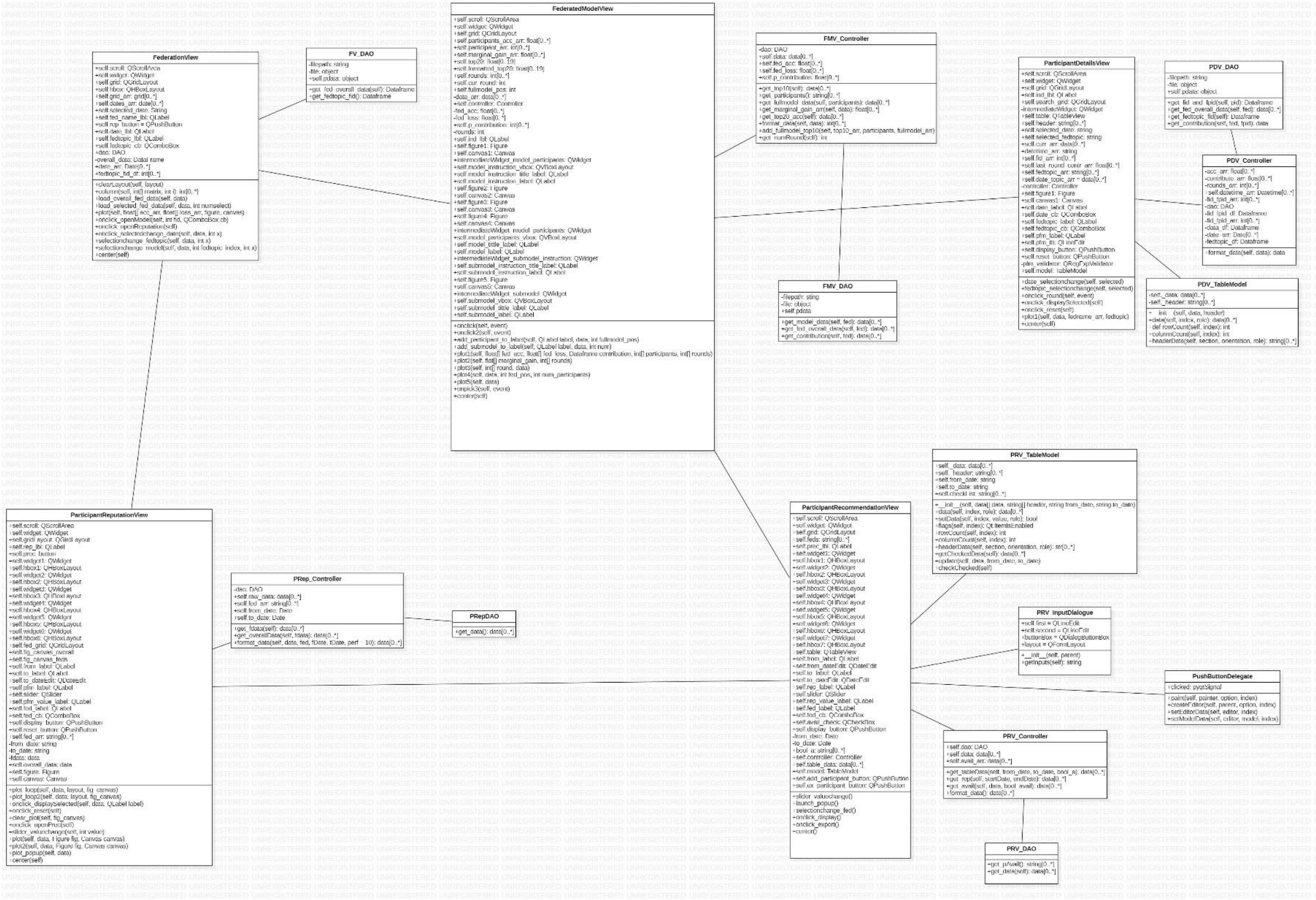


Figure 3.2 Class diagram

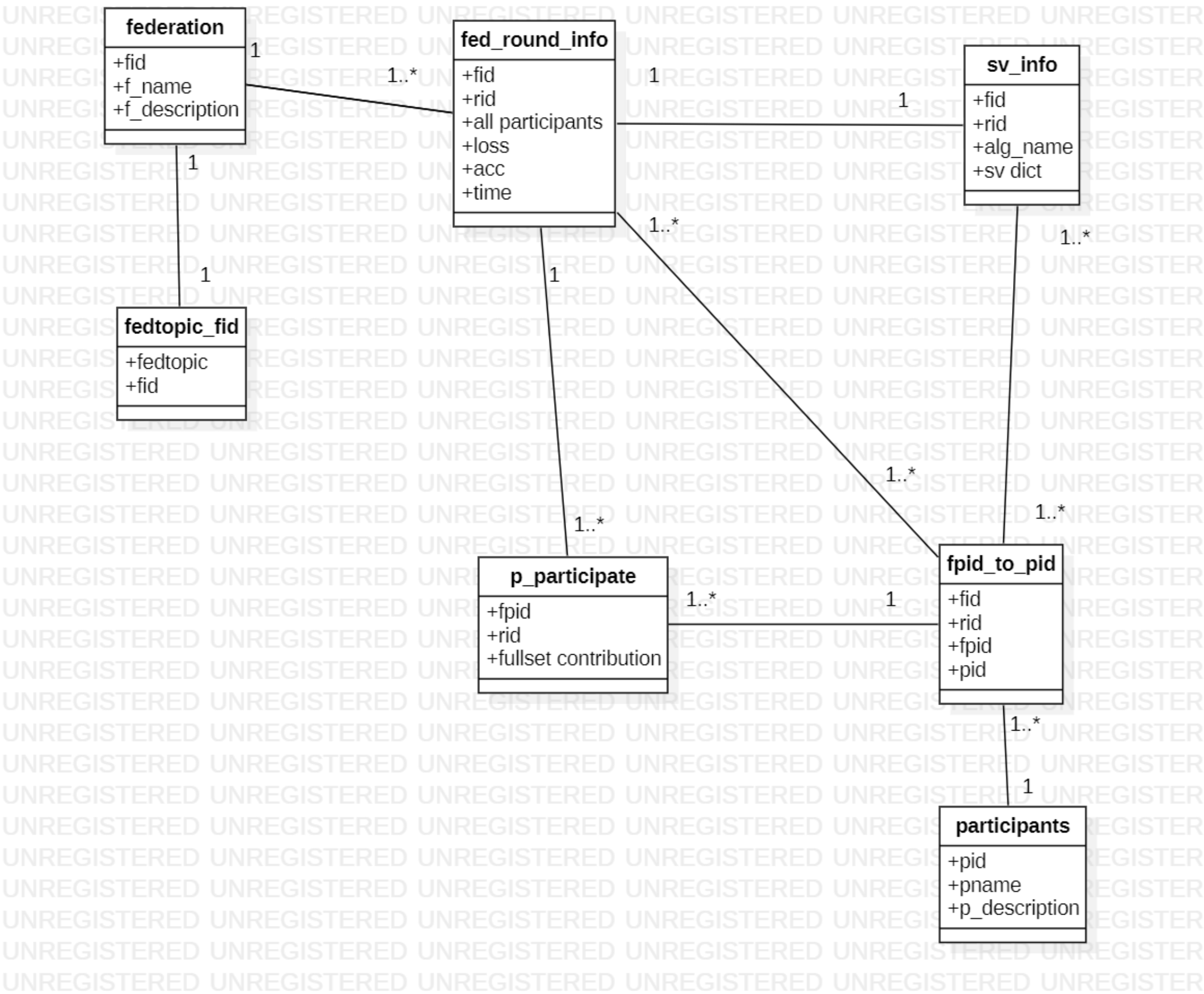


Figure 3.3 Entity relations diagram

3.2 Application Description

The visualisation system needs to be simple enough for the stakeholders to understand the output from federated learning and make decisions. Thus, relevant data is not to be excluded from the visualisation system to make it simpler. Relevant data is defined as the accuracy, loss and Shapley value from the output of the GTG – Shapley algorithm.

Narrowing down to the relevant data is still a lot of data due to the number of models and sub-models produced. Thus, the system is designed to allow the user to select the specific model, sub-model, round and other data through the use of UI such as drop-down lists, buttons, clickable graphs and search bar. This design is implemented in all the views.

Besides allowing the user to select specific data, overall data is displayed in Federated Model View and Participant Reputation View. This is to allow the user to compare the overall data set and the selected data set.

3.3 Lo-Fi Prototype

In this section, the lo-fi prototype design of the visualisation system will be shown and discussed.

3.3.1 Initial Design

The initial design of the visualisation system is to be a single view application. Its purpose is to display the performance, marginal gain, contribution of the participants, participant details and participant availability. This is found to be not a practical design as too much functionality is put into a single screen.

The initial design lacks the ability to select and display training round sub-model details, preventing the user from further examining the federated model. For the participant details, it lacks the required space to display all the participant's contributions towards a federation. Adding on, viewing the participant's contribution for a different model will require the system all the load data of the selected model.

Lastly, the participant availability lacks the function to allow the user to select participants and export a list. Another reason that the participant availability should be in a separate view is that viewing of the participant availability section should be independent from the viewing of the model details.

Thus, the initial design is broken up into several views: Federations View, Federated Model View, Participant Reputation View, Participant Details View Participant Recommendations View.

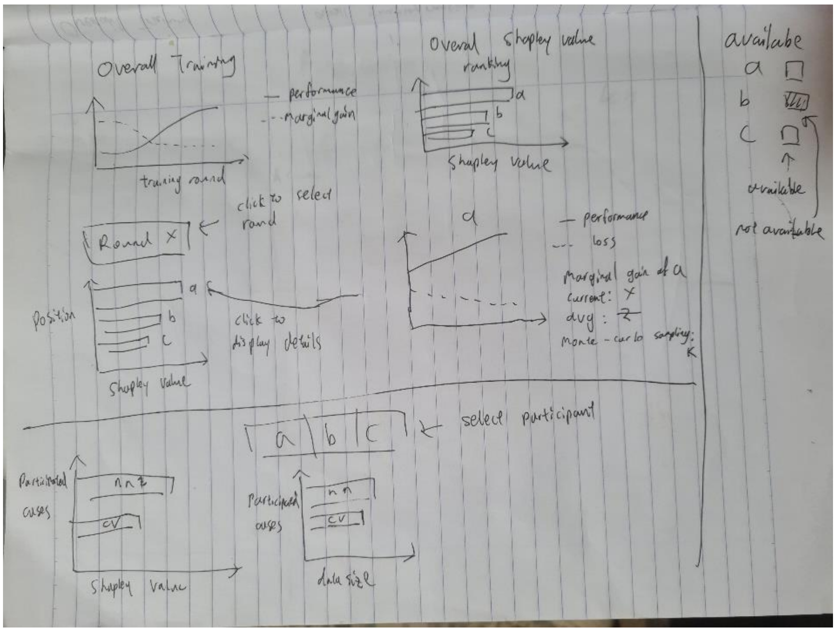


Figure 3.4 Initial prototype design

3.3.2 Federations View

The Federations View is designed to provide an overview of federated models. It shows the accuracy, loss and federated model details. The initial design did not take into account multiple federated models will be displayed in this view. Thus, the prototype only displays 1 model. Date picker and federation drop-down list are added in the implementation to facilitate the filtering of the model by the user. This is to allow the user to select a specific model to examine.

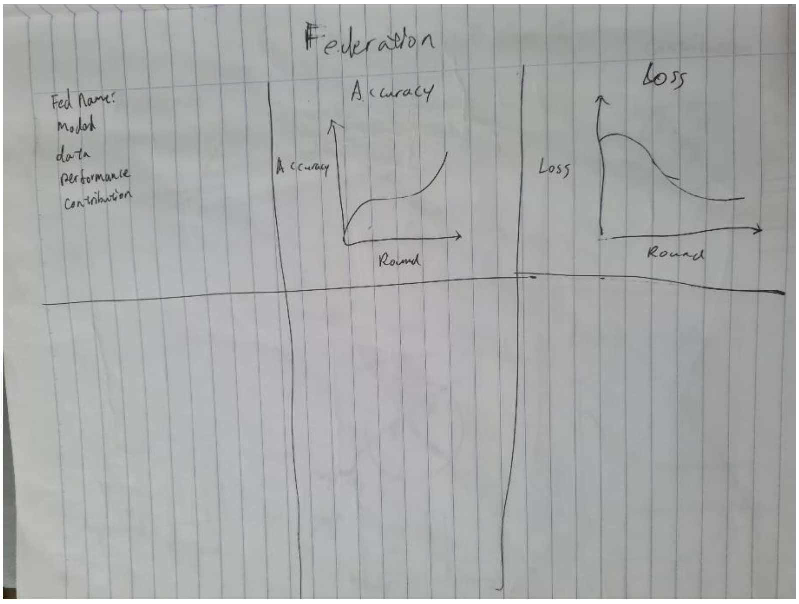


Figure 3.5 Federations View prototype

3.3.3 Federated Model View

Federated Model View is designed to provide an overview of the selected federated model and allow the user to view details of the training rounds and sub-models. The accuracy, loss, participant contribution and marginal gain is displayed as the overview of the federated model. The Marginal Gain graph is displayed in a line graph rather than a bar graph to better show the drop in marginal gain in each round. The Marginal Gain graph is made to be clickable to allow the user the examine the sub-models and participants in each round. The Marginal Gain in Round X graph is not implemented as it is redundant.

The Top 10 Model graph was designed to be displaying the top 10 sub-models of the last round. Furthermore, if the federated model is not in the top 10, it will not be displayed in this graph. The Top 10 Model is redesigned to show the top 10 sub-models based on the round clicked on by the user in the Marginal Gain graph. The redesigned Top 10 Model Graph will be displaying the federated model and its ranking regardless of it is in the top 10 or not.

On the right of the Top 10 Model graph, it is the Sub Models graph. It was supposed to show the sub model’s accuracy in a per round manner. Hovering above each square displays a list of participants and their contributions. Clicking on the square will display the accuracy, loss and participant contribution of the selected sub-model.

The Sub Models graph is redesigned to display the top 20 sub-models based on the round clicked on by the user in the Marginal Gain graph. The on-hover interaction was not implemented due to the difficulty in implementing it. Clicking on the sub-model displays the summarised sub-model details in a panel instead of 3 panels of accuracy, loss and participant contribution.

In the implementation, there will be a panel for the Marginal Gain graph and Sub Models graph each to inform the user that the 2 graphs are clickable.

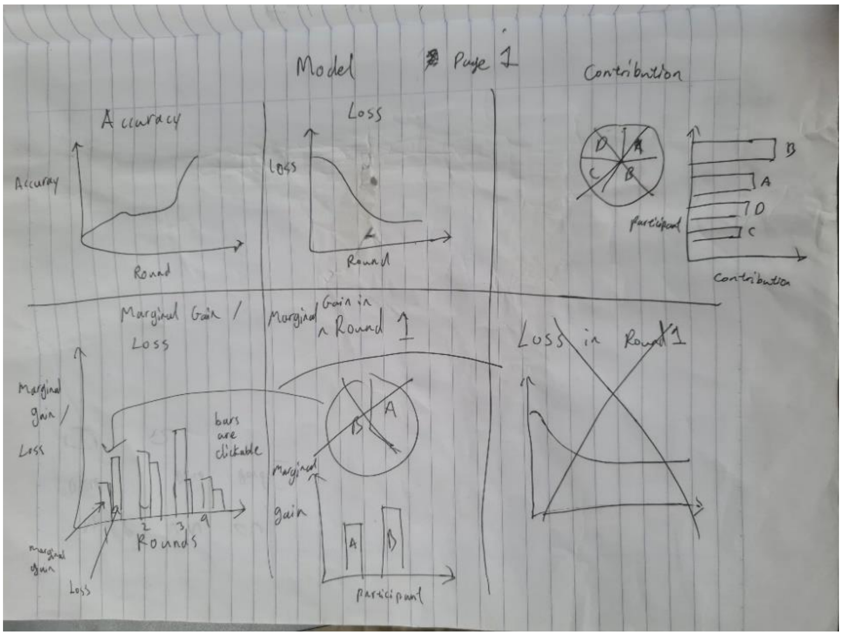


Figure 3.6 Federated Model View prototype part 1

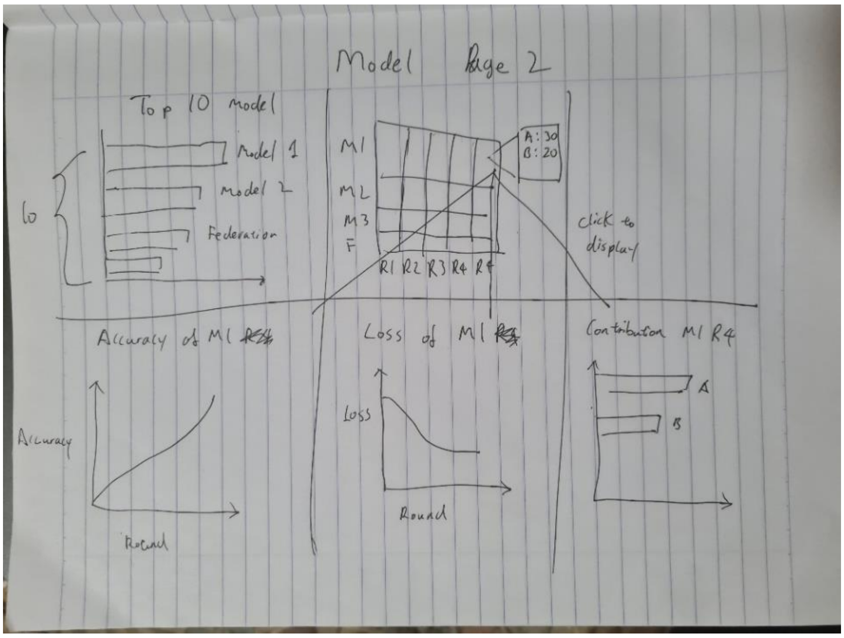


Figure 3.7 Federated Model View prototype part 2

3.3.4 Participant Reputation View

Participant Reputation View is designed to provide an overview of the participant contribution to federated learning. In the Overall Reputation graph, it shows the overall participant contribution towards the federated learning and federation topic. For the NLP and CV graphs, it shows the participant contribution towards the federation topic and the federated models.

The search function for Participant Reputation View on the right was designed to filter the data and displayed on Overall, NLP and CV graphs. In the implementation, the search function is implemented as a search bar in the second row. The displayed result is changed to be displayed in a results graph right below the search bar.

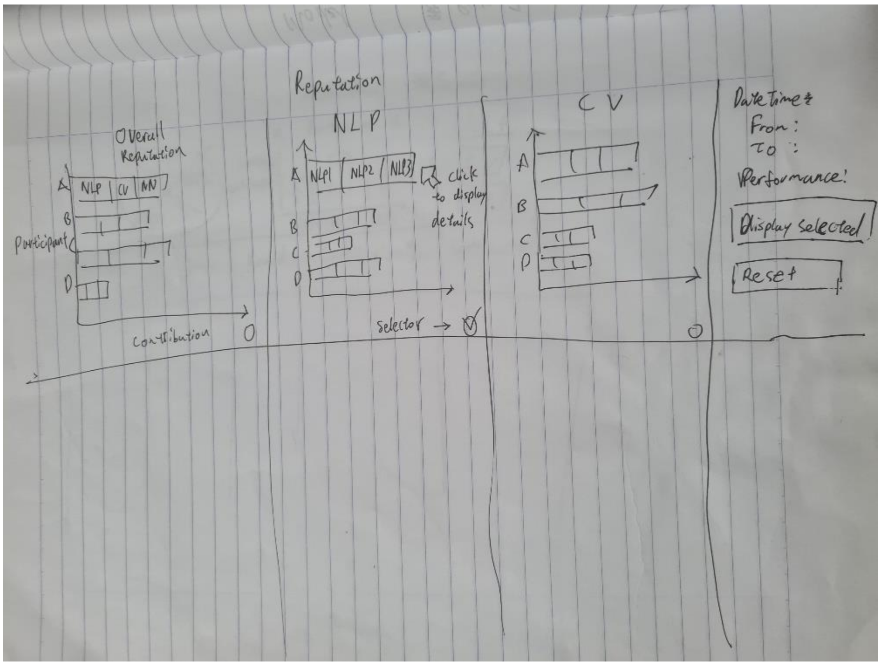


Figure 3.8 Participant Reputation View prototype

3.3.5 Participant Details View

Participant Details View is designed to display the participant’s contribution and contribution percentage in a table. The graph of the participant’s contribution on the top left towards the models. The displayed data can be changed by clicking on one of the buttons on the right of the graph.

In the implementation, the buttons are replaced with a drop-down list for the user to choose a federation topic. A Date drop-down list and a Performance textbox are added to give the user more choice on filtering the displayed data. The graph is made to be clickable to bring the user to Federated Model View.

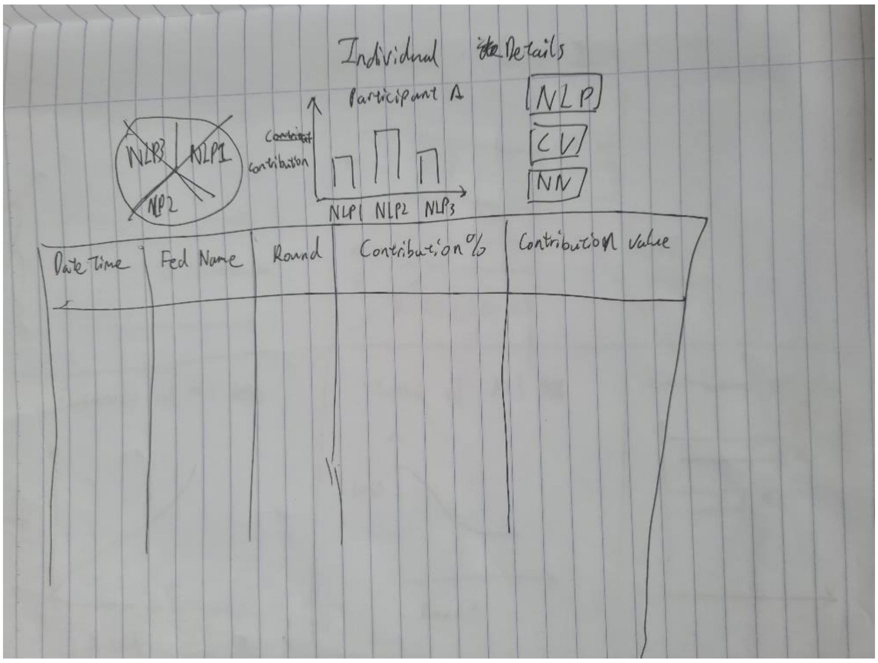


Figure 3.9 Participant Details View prototype

3.3.6 Participant Recommendations View

A lo-fi prototype was not drawn for Participant Recommendations View. However, designs from implementing Participant Reputation View and Participant Details View are applied to Participant Recommendations View.

The main functionality of Participant Recommendations View is to display an overview of the participant reputation in a table. The table displays the participant name, reputation score, federation topic and availability. Above the table there is a search bar for the user to filter out the participant list. Buttons are available for the user to export the participant list and add participants.

3.4 Programming Language

Python is the chosen programming language as GTG – Shapley is developed using Python too, allowing the visualisation system to take in output from GTG – Shapley with ease. With

GTG – Shapley producing several dataframes as output, using Python will allow direct manipulation of the dataframes or the data in it.

Secondly, Python has easy to use UI and graphing libraries used for the development of the visualization system.

3.5 Libraries

3.5.1 Pandas

Using the Pandas library is required to read the dataframes stored in dbs_v2.pkl. Pandas is further used for manipulation of the dataframes such as joining 2 dataframes together or selecting specific data in the dataframe.

3.5.2 PyQt 5

Initially, Tkinter was the library used for the development of the UI. In the middle of the development process, it is discovered that Tkinter have performance issues with displaying multiple graphs. After discovering the performance issue, PyQt 5 is found to be more suitable for displaying multiple graphs.

3.5.3 Matplotlib and NumPy

After testing several graphing libraries, Matplotlib is chosen due to its simplicity in embedding into the UI. NumPy is used in the plotting of the bar graphs by calculating the y coordinates input data.

3.6 Designed for Stakeholders

The application is designed for stakeholders such as the participants, federation initiator and federation owner to use. The main focus of the design will be on allowing the participants and federation initiators to view data from federated learning. For the federation owner, the federation owner will be able to oversee all the federated learning that has taken place and recommend participants for the federation initiator.

3.6.1 Participants

A participant is a stakeholder who is compensated for training the federated model. Participants would be interested to know how well their data performed in the model they participated in and against other participants. Participants will be able to view how well their data has formed in the federations they have participated in in Participant Details View. In the

Participant Details View, it allows the participant to select the date and federation topic to view details of the federations for the selected date and federation topic.

The participant can go into Federated Model View by clicking on the federated model on the graph. In Federated Model View**,** the participant will be able to compare contributions against other participants and see the ranking for the selected federation.

3.6.2 Federated Learning Initiators

A federation initiator is a stakeholder who starts the federated learning process, invites and compensates participants for training the federated model. When initiating a federated learning process, the federation initiator will need to evaluate a list of participants and then invite them to participate. This list of participants will be available in Participant Recommendations View. The federation initiator will be able to filter the list of participants based on a date range, reputation score, federation topic and participant availability.

After the federation initiator has initiated the federated learning, the federation initiator will be able to see an overview of the federated learning in Federations View. In the Federations View, the federation initiator will be able to view the accuracy and loss of the federated model. If the federation initiator has initiated more than 1 federated learning, the results viewed can be filtered by date and federation topic.

From the Federations View, the federation initiator view details of the federated model by clicking on the respective View Model button. By clicking on the View Model button, it brings the federation initiator to Federated Model View where the federation initiator can see the contribution of each participant, marginal gain per round, top 10 sub-model details and top 20 sub-model details.

3.6.3 Federation Owners

A federation owner is a stakeholder that owns the federated learning system and oversees all the federated learning processes. The federation owner will be able to view all the federations and monitor them. The participant list will be managed by the federation owner too.

4. Results

4.1 Overview

In this section, each of the implemented views: Federations View, Federated Model View, Participant Reputation View, Participant Recommendations View and Participant Details View will be discussed in detail. The discussion includes the user interaction and functionality.

4.2 Federations View

The Federations View provides an overview of federated models. It displays federated models grouped by federation topic in each section (1). In each section, federation topic, federation name, date, model name, performance, contributors, accuracy and loss of the selected federated model is displayed. The user can select a date from the date drop-down list at (2) to view models on a particular date and further filter by federation topic using the federation topic drop-down list at (3).

In each section, the user can view the overview of the specific model of the federation topic by selecting from the model drop-down list at (4). To view the details of the select model, the user can access Federated Model View by clicking on the View Model button at (5).

To view the participant reputation, the user can access Participant Reputation View by clicking on the View Reputation button at (6).

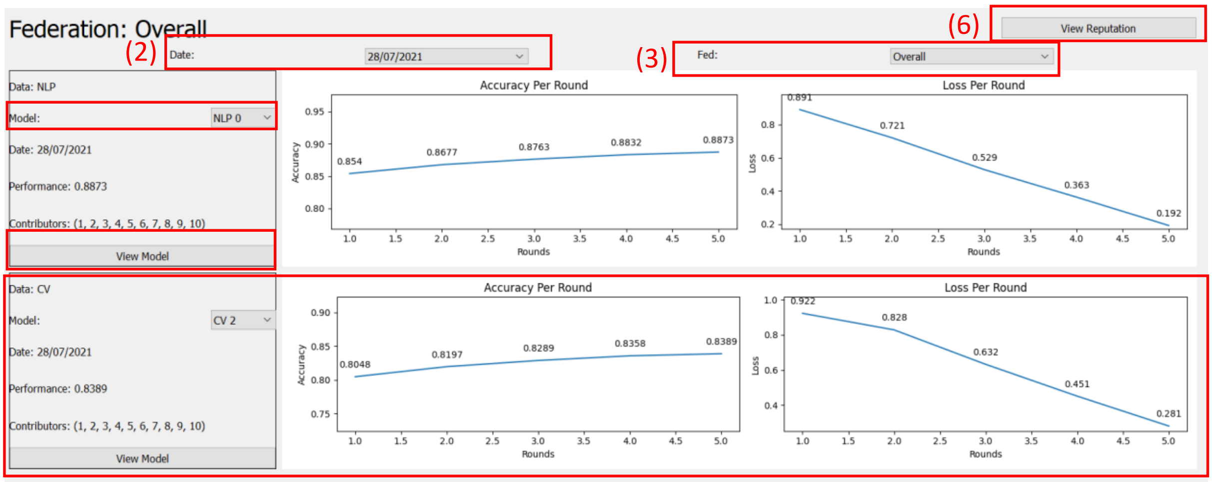


Figure 4.1 Federations View implementation

4.3 Federated Model View

In the Federated Model View provides an overview of the federated model and allows the user to view details of the selected round of the federated model or sub-model details. The first 2 rows in Federated Model View, display accuracy, loss and marginal gain in per round manner in line graphs to allow the user to evaluate the federated model.

The participant contributions displayed in the first row at (7) is calculated based on the Shapley value assigned to each participant by GTG – Shapley algorithm. As the Shapley value assigned to the participant is a long floating number, participant contribution is displayed in percentage instead. Firstly, all the positive and negative contribution is being summed up. The absolute value of the bigger sum will be used as the denominator. Contribution percentage is the participant’s contribution divided by the denominator then multiply by 100.

Having a positive contribution percentage means that training of the participant’s data on the federated model improved the accuracy and reduced loss of the federated model. On the other hand, having a negative contribution means that training of the participant’s data on the federated model harmed the accuracy and increased loss of the federated model.

Moving on, the user is able to view participants and accuracy of the top 10 sub-models of each round by clicking on the round marks in Marginal Gain per Round graph at (8). On the left of the Marginal Gain per Round graph, it informs the user that the round marks can be clicked to view the sub-model details. By having this feature, it allows the user to compare the performance of the federated model and sub-models.

The feature is implemented as it is not likely that the federated model performs better than the top few sub-model in a real-world setting. It might be due to data collected from different domains, noise, quality of data and so on. Thus, it is important to review the sub-models and compare it to the federated model in order to select the best model for use.

With respect to the UI, clicking on one of the markers at (8) will display the submodel and federated model performance and its participants of the selected round in the third row. By clicking on the marker, participants’ contribution percentage at the selected round will be displayed too. This allows the user to monitor the participants’ contributions for all the rounds.

Each square at (10) represents a sub-model. The number represents the sub model's performance multiplied by 100. The performance is multiplied by 100 and rounded up to the nearest whole number as the grid is unable to display float numbers. Clicking on the square the sub-model details will be displayed in the Sub Model Details section on the right of (10). Doing so allows the user to review the sub-model details and compare between sub-models.

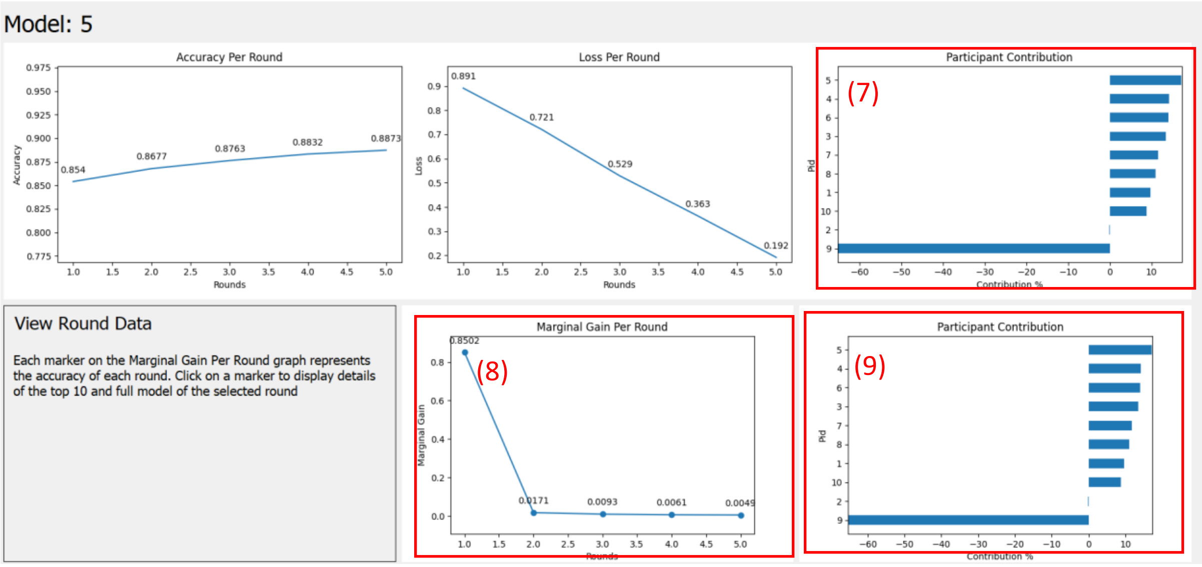


Figure 4.2 Federated Model View implementation part 1

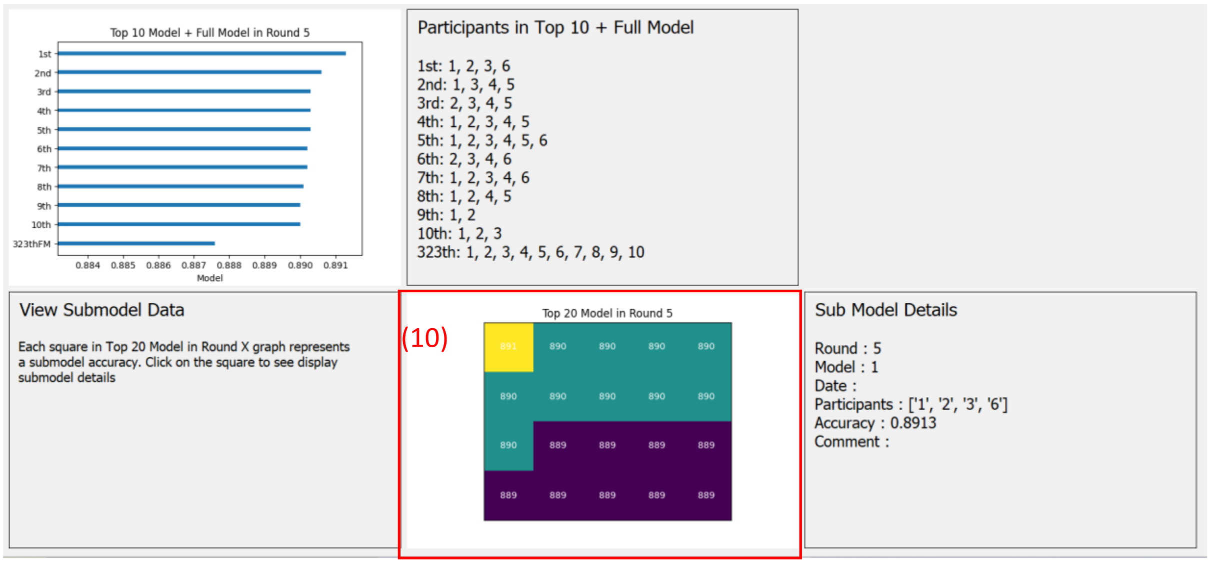


Figure 4.3 Federated Model View implementation part 2

4.4 Participant Reputation View

The Participant Reputation View provides an overview of participant contribution across all the federations. The overall graph in the first row shows the participant's total contribution towards all the federations that the participant has joined. The contribution highlighted in different colours represents the contribution towards the respective federation topic shown in the legend.

The rest of the graphs in the first row shows the participant's contribution towards the federation topic. Similar to the overall graph, the contribution is highlighted. The highlighted represents the contribution towards the respective federation shown in the legends.

The second row of Participant Reputation View is a search bar. UI from (12) to (16) can be manipulated by the user to display the desired selected data in (17). The date pickers at (12) allow the user to select a date range. The slider at (13) allows the user to select the minimum model performance value between 10 and 100. At (14) the user can select a federation topic from the drop-down list. Clicking on the Display Selected button at (15) will display the filtered data on the graph at (17). Similar to the overall graph in the first row, the contribution is highlighted to show contribution towards the federation topics. Clicking on the Reset Button at (16) resets the state of the search bar UI from (12) to (14) and the graph at (17).

To access the Participant Recommendations View, the user can click on the View Participant Recommendation button at (11).

Figure 4.4 Participant Reputation View

4.5 Participant Recommendations View

The Participant Recommendations View provides an overview of the participants, their reputation score, participated federations and availability. The reputation score is assigned to participants based on their contributions to federated learning. A newcomer would receive a score of 50. The score will increase or decrease based on the participant's contribution.

The Participant Recommendation View is divided into 2 parts. The search bar on top and the table of participants below. The search bar consists of UI from (18) to (22). The date pickers at (18) allow the user to select a date range. The slider at (19) allows the user to select the minimum reputation score of the participant having a value between 10 and 100. At (20) the user can select a federation topic from the drop-down list. Ticking the checkbox at (21) allows the user to only display available participants. Clicking on the Display Selected button at (22) will display the filtered list of participants in the table below.

The user can click on the participant's name at (23) to access Participant Details View. To export the list of participants into a CSV file, the user can tick the checkbox next to the participant's name and click on the Export Participant button at (25). When the export is successful, a success message will be displayed (Figure 4.6). However, if the CSV file is opened, exporting to the CSV file will not be available and an error message will be shown (Figure 4.8). If the user has selected 1 or more unavailable participant(s), a prompt will inform the user about the unavailable user(s) and ask to continue the export or not (Figure 4.9). The export operation will be aborted if the user chooses to. When the user tried to export without selecting any participant, an error message prompts the user that no participant has been selected (Figure 4.11).

The user can add participants by clicking on the Add Participant button at (24). A popup will appear to allow entering of the participant's name and initial reputation score. The default reputation score is 50 and can be adjusted by the user.

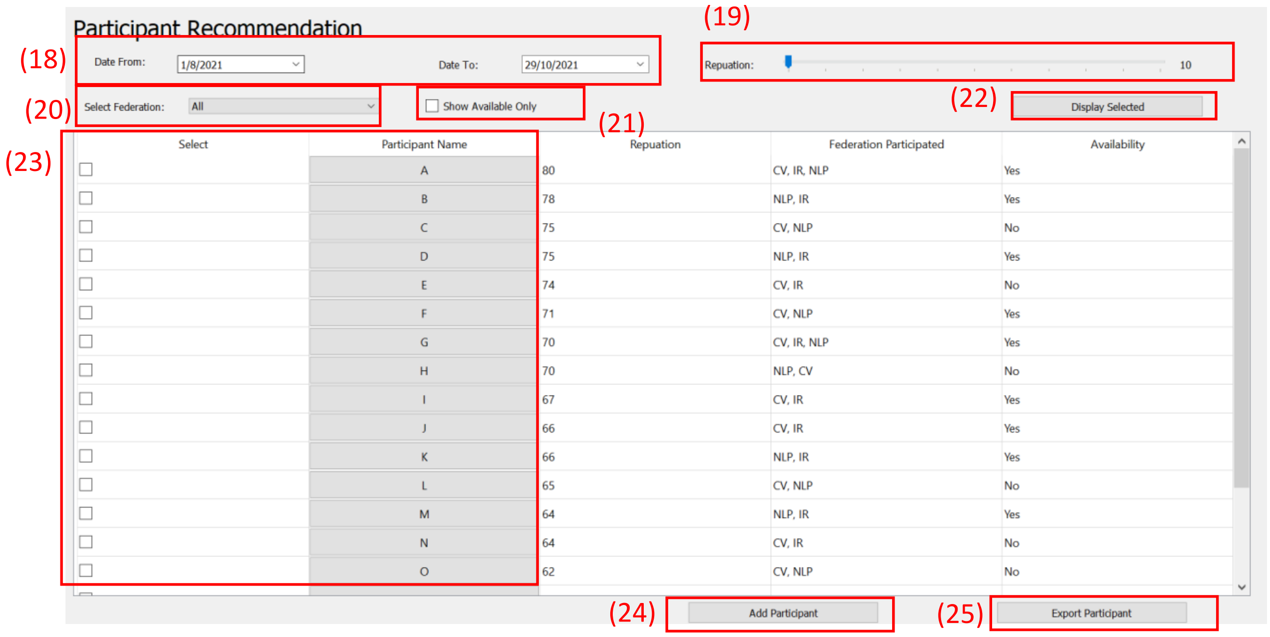


Figure 4.5 Participant Recommendations View

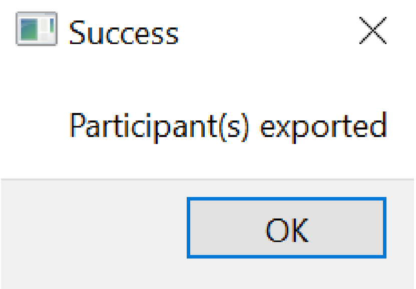
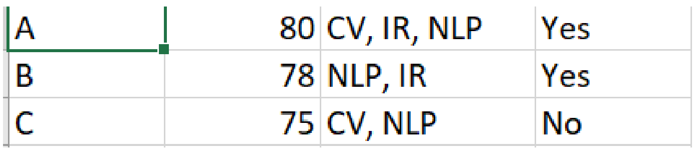


Figure 4.6 Successful export



A	80	CV, IR, NLP	Yes
B	78	NLP, IR	Yes
C	75	CV, NLP	No

Figure 4.7 Export Result

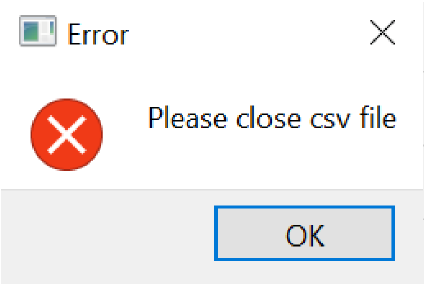


Figure 4.8 Opened csv file error

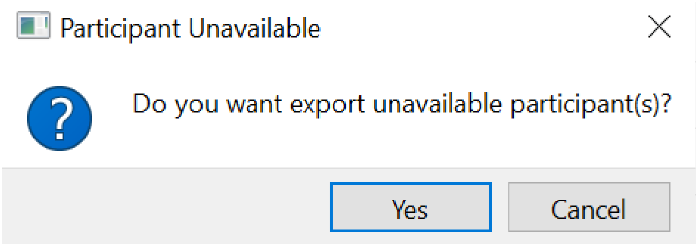


Figure 4.9 Unavailable participant prompt

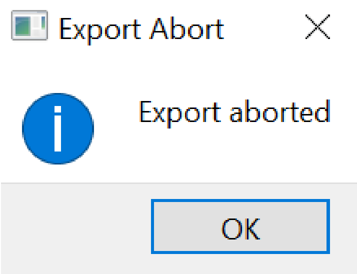


Figure 4.10 Export abort

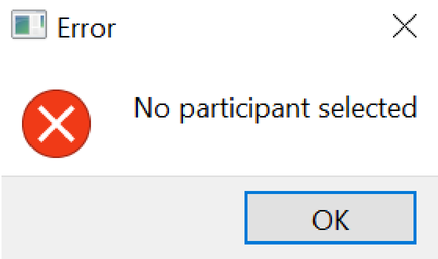


Figure 4.11 No participant selected

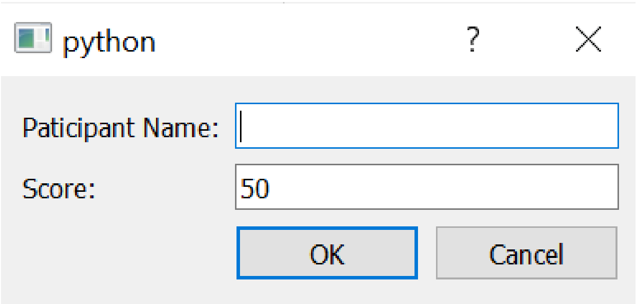


Figure 4.12 Add participant

4.6 Participant Details View

The Participant Details View allows the user to view the model performance of the federated model(s) that the participant had participated, the participant's contribution and contribution percentage. The data displayed is in a per round manner, filtered by date and federation topic.

The user can select the date from the drop-down list at (26) and a federation topic from the drop-down list at (27). A desired minimum performance value can be entered into the performance textbox at (28) and click on the Display Selected button at (29) to filter data by the minimum performance value. The reset button at (30) can be clicked on to reset data displayed in the table.

The graph at (31) can be clicked on the access the selected model. There will be a pop up when clicked, asking if the user wants to view the selected model. If yes, it will bring the user to the Federated Model View. Otherwise, it returns the user back to the Participant Details View. The table will also be updated to display data of the selected model.

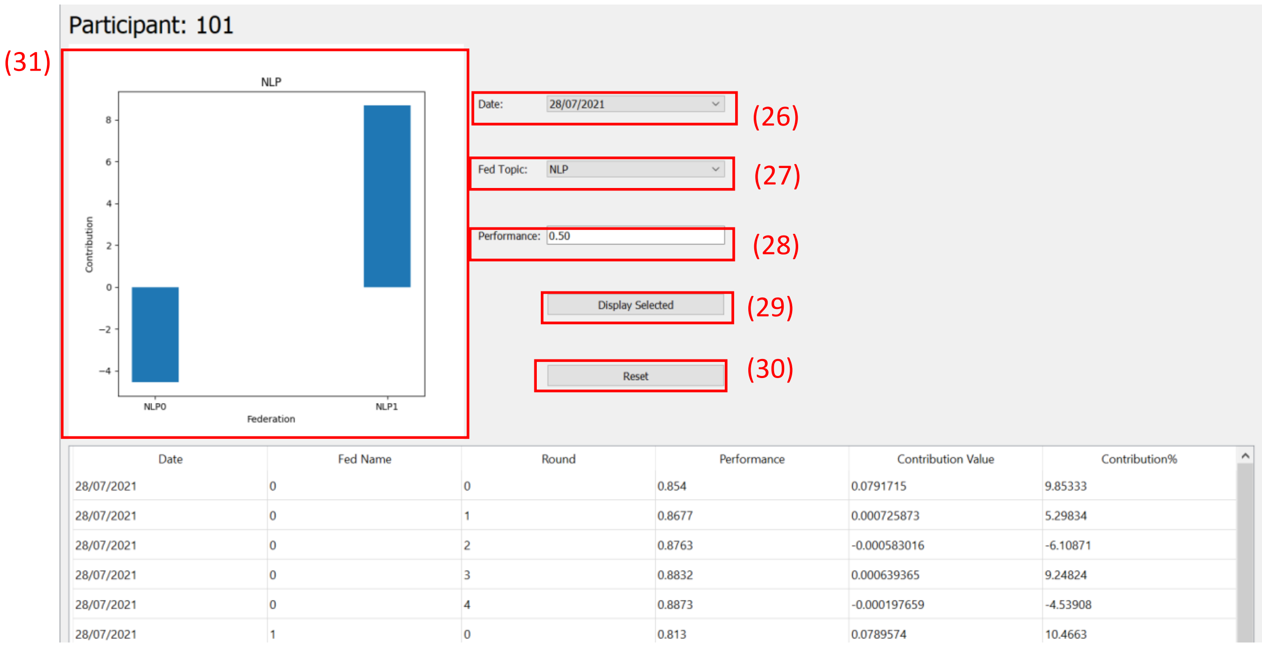


Figure 4.13 Participant Details View implementation

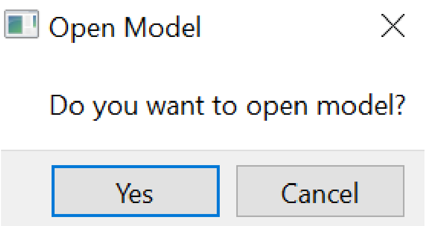


Figure 4.14 Open model prompt

4.7 Conclusion

The goal of the research is to develop a visualisation system that can present the output from federated learning in a manner that stakeholders can easily interpret. A visualisation system has been developed to take in the output from GTG – Shapley and present the data in easy to interpret graphs and lists. This allows stakeholders to examine the selected model, its training round and sub-models details with ease.

The limitation of the developed visualisation system is the lack of anomaly detection. Without anomaly detection, it would be difficult to detect attacks from a sophisticated malicious participant. Thus, anomaly detection could be considered for future research.

References

[1] X. Wei, Q. Li, Y. Liu, H. Yu, T. Chen and Q. Yang, "Multi-Agent Visualization for Explaining Federated Learning," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence Demos*, Macao, 10-16 August 2019, Pages 6572-6574, doi: doi.org/10.24963/ijcai.2019/960.

[2] T. Song, Y. Tong and S. Wei, "Profit Allocation for Federated Learning," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 2577-2586, doi: 10.1109/BigData47090.2019.9006327.

[3] L. MENG, Y. WEI, R. PAN, S. ZHOU, J. ZHANG, and W. CHEN, "VADAF: Visualization for Abnormal Client Detection and Analysis in Federated Learning", *ACM Transactions on Interactive Intelligent Systems*, Vol. 11, 3-4, Article 26, August 2021, doi: https://doi.org/10.1145/3426866.

[4] Matt Jackson. "GTO-7-03: The Shapley Value," YouTube, Nov 22, 2013, site: https://www.youtube.com/watch?v=qcLZMYPdpH4 [Accessed: Sep 19, 2021].

[5] Y. Narahari, "Game Theory Lecture Notes by Chapter 32. the Shapley Value," October 2012, site: https://www.semanticscholar.org/paper/Game-Theory-Lecture-Notes-by-Chapter-32.-the-Value-Narahari/f8f78a0f1849f58fb24b712c0ce56ac1902c62b0#related-papers

[6] A. Ghorbani, J. Zou, "Data Shapley: Equitable Valuation of Data for Machine Learning", arXiv:1904.02868 [stat.ML], 10 Jun 2019.

[7] Z. Liu, Y. Chen, H. Yu, Y. Liu, L. Cui, "GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning", arXiv:2109.02053 [cs.AI], 5 Sep 2021.

Appendix

Sample data in dbs_v2.pkl

federation table

	fid	f_name	f_description
0	0	MNIST_case1_MR	fed=0 data=OurMNIST sv_alg=MR case=1
1	1	MNIST_case2_MR	fed=1 data=OurMNIST sv_alg=MR case=2
2	2	MNIST_case3_MR	fed=2 data=OurMNIST sv_alg=MR case=3
3	3	MNIST_case4_MR	fed=3 data=OurMNIST sv_alg=MR case=4
4	4	MNIST_case5_MR	fed=4 data=OurMNIST sv_alg=MR case=5
0	5	MNIST_case5_MR	fed=5 data=OurMNIST sv_alg=MR case=6
1	6	MNIST_case5_MR	fed=6 data=OurMNIST sv_alg=MR case=7

fed_round_info table

	fid	rid	all participants	loss	acc	time
0	0	0	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.891	0.8540	2021-07-28 10:35:59
1	0	1	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.721	0.8677	2021-07-28 11:00:30
2	0	2	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.529	0.8763	2021-07-28 13:25:18
3	0	3	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.363	0.8832	2021-07-28 15:07:39
4	0	4	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.192	0.8873	2021-07-28 17:28:47
5	1	0	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.836	0.8130	2021-07-28 09:20:28
6	1	1	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.748	0.8492	2021-07-28 10:40:12
7	1	2	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.663	0.8597	2021-07-28 12:19:57
8	1	3	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.483	0.8647	2021-07-28 14:37:32
9	1	4	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.295	0.8712	2021-07-28 16:14:47
10	2	0	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.922	0.8048	2021-07-28 09:40:28
11	2	1	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.828	0.8197	2021-07-28 10:20:12
12	2	2	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.632	0.8289	2021-07-28 12:49:57
13	2	3	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.451	0.8358	2021-07-28 14:57:32
14	2	4	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	0.281	0.8389	2021-07-28 16:24:47

sv_info table

	fid	rid	alg_name	sv dict
0	0	0	MR	{(): '0.0505', (1,): '0.8464', (2,): '0.8492',...
1	0	1	MR	{(): '0.8540', (1,): '0.8617', (2,): '0.8625',...
2	0	2	MR	{(): '0.8677', (1,): '0.8673', (2,): '0.8748',...
3	0	3	MR	{(): '0.8763', (1,): '0.8784', (2,): '0.8778',...
4	0	4	MR	{(): '0.8832', (1,): '0.8807', (2,): '0.8830',...
5	1	0	MR	{(): '0.0586', (1,): '0.7316', (2,): '0.7283',...
6	1	1	MR	{(): '0.8130', (1,): '0.7978', (2,): '0.7945',...
7	1	2	MR	{(): '0.8492', (1,): '0.8183', (2,): '0.8261',...
8	1	3	MR	{(): '0.8597', (1,): '0.8207', (2,): '0.8219',...
9	1	4	MR	{(): '0.8647', (1,): '0.8271', (2,): '0.8324',...
10	2	0	MR	{(): '-0.0415', (1,): '0.7754', (2,): '0.7717'...
11	2	1	MR	{(): '0.8048', (1,): '0.8042', (2,): '0.8043',...
12	2	2	MR	{(): '0.8197', (1,): '0.8197', (2,): '0.8173',...
13	2	3	MR	{(): '0.8289', (1,): '0.8247', (2,): '0.8260',...
14	2	4	MR	{(): '0.8358', (1,): '0.8368', (2,): '0.8318',...

participants table

	pid	p_name	p_description
0	101	case=1 id=1	pid=101 case=1 data=OurMNIST
1	102	case=1 id=2	pid=102 case=1 data=OurMNIST
2	103	case=1 id=3	pid=103 case=1 data=OurMNIST
3	104	case=1 id=4	pid=104 case=1 data=OurMNIST
4	105	case=1 id=5	pid=105 case=1 data=OurMNIST
5	106	case=1 id=6	pid=106 case=1 data=OurMNIST
6	107	case=1 id=7	pid=107 case=1 data=OurMNIST
7	108	case=1 id=8	pid=108 case=1 data=OurMNIST
8	109	case=1 id=9	pid=109 case=1 data=OurMNIST
9	110	case=1 id=10	pid=110 case=1 data=OurMNIST
10	201	case=2 id=1	pid=201 case=2 data=OurMNIST
11	202	case=2 id=2	pid=202 case=2 data=OurMNIST
12	203	case=2 id=3	pid=203 case=2 data=OurMNIST
13	204	case=2 id=4	pid=204 case=2 data=OurMNIST
14	205	case=2 id=5	pid=205 case=2 data=OurMNIST

p_participate table

	fpid	fid	rid	fullset contribution
0	1	0	0	0.079171
1	2	0	0	0.080514
2	3	0	0	0.080433
3	4	0	0	0.080489
4	5	0	0	0.081112

fpid_to_pid table

	fid	rid	fpid	pid
0	0	0	1	101
1	0	0	2	102
2	0	0	3	103
3	0	0	4	104
4	0	0	5	105

fedtopic_fid table

0	NLP	0
1	NLP	1
2	CV	2
3	CV	3
4	IR	4
5	IR	5
6	NLP	6}