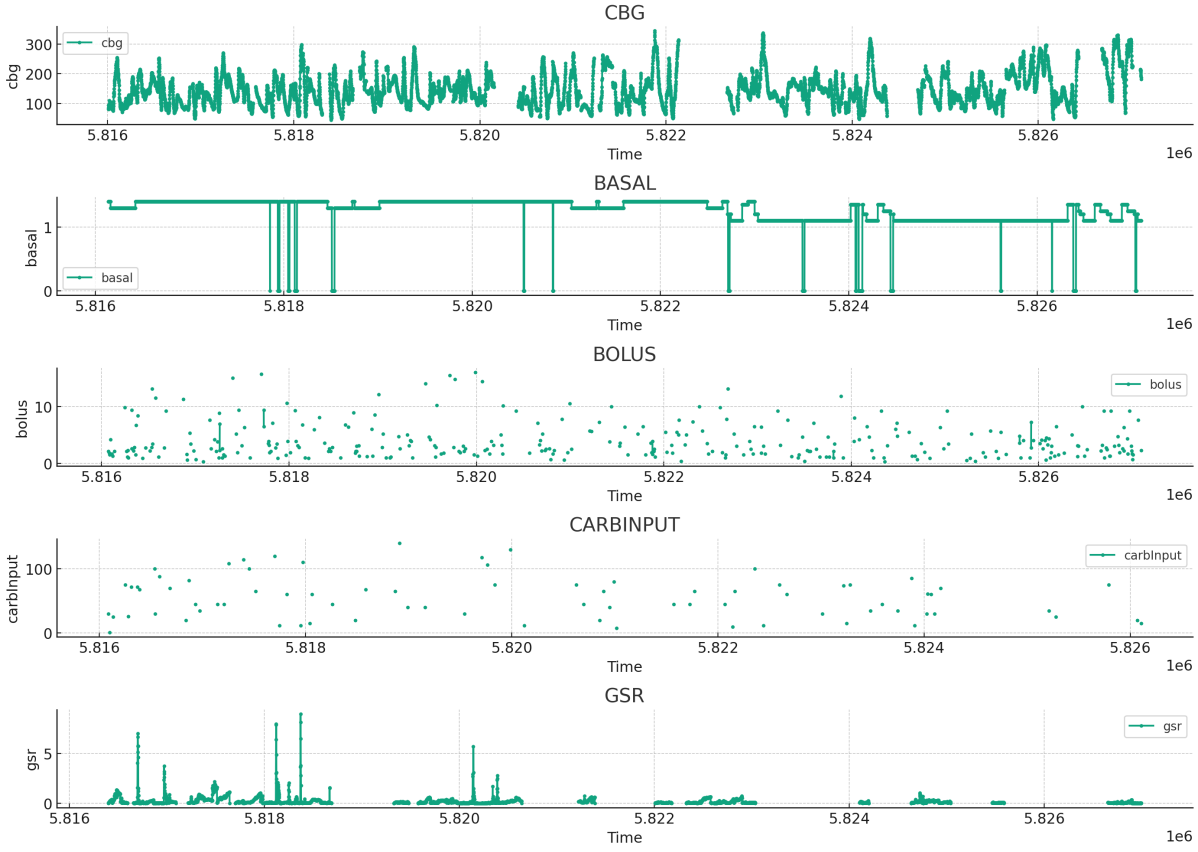
Blood Glucose Prediction using Federated Learning and Graph Neural Networks

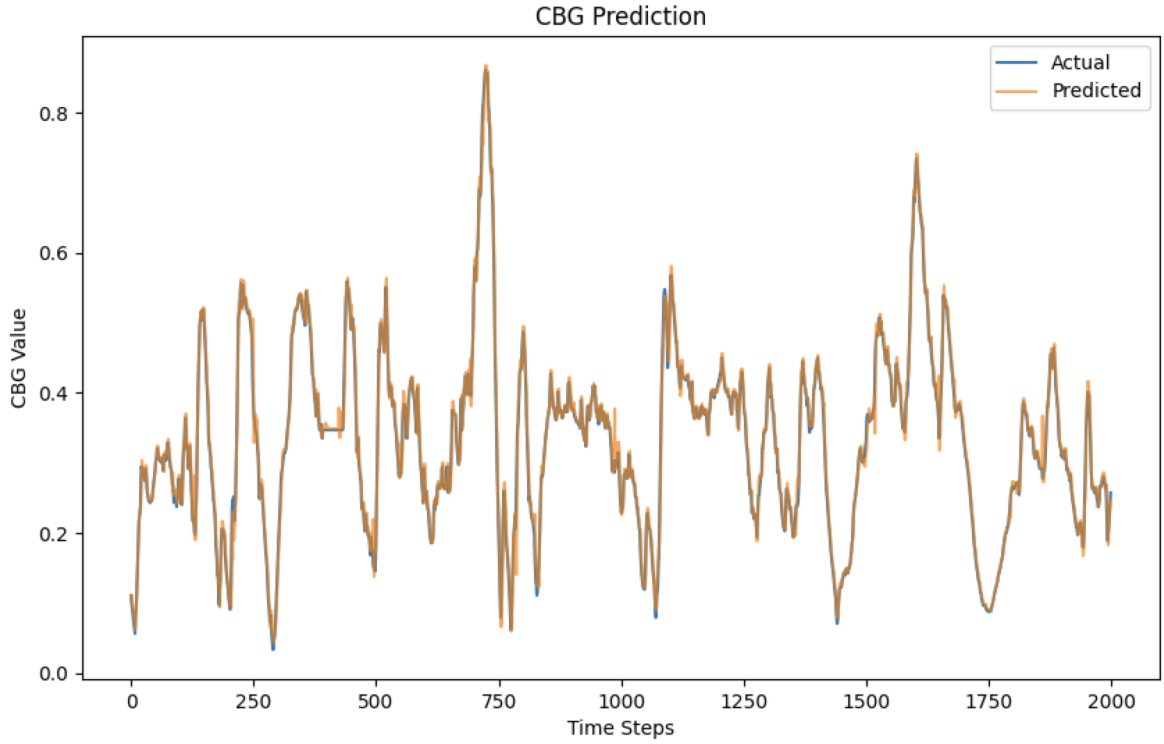
1 Data Overview



2 Simple LSTM

The preprocessing phase involved handling the 'cbg' column with mean imputation while taking into account the 'missing cbg' column. Time was transformed into actual datetime format and rounded to minutes, which was later disregarded as each row represented equidistant time steps. Due to substantial missing data, heart rate and finger information was omitted from the analysis. The features 'cbg', 'basal', and 'gsr' were normalized using the MinMaxScaler to ensure uniform scaling.

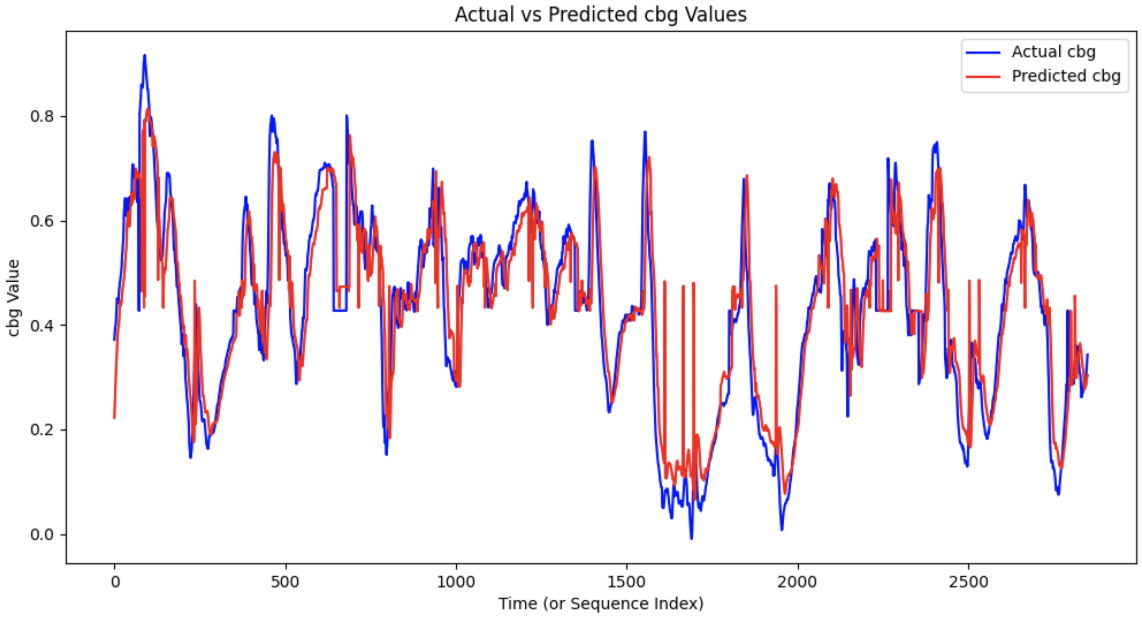
A simple LSTM model was developed, utilizing a window sliding technique to predict glucose levels 5 time steps ahead, based on a 10-step historical context. This configuration was fine-tuned to yield optimal results. The model was specified with a mean squared error loss function and the Adam optimizer.



3 Graph Neural Network (non-federated)

A spatial-temporal graph structure was implemented, where each graph corresponded to a specific time point, with nodes representing the features 'cbg', 'basal', 'bolus', 'carbInput', and 'gsr'. The edges were defined to represent the influence of carbohydrate intake, basal insulin, and bolus insulin on 'cbg', as well as the effect of 'cbg' on 'gsr'. The network received a sequence of graphs as input and outputted the predicted 'cbg' value for future time steps.

The architecture comprised GCN layers for processing the graph structure, an LSTM layer to capture temporal dependencies, and a final output layer. Mean pooling was employed to aggregate node features across each graph in a batch. Challenges regarding batch size, sequence length, and network dimensions were addressed during the implementation phase. The average test loss was reported as 0.026, indicating room for further improvement through minor adjustments to the network.



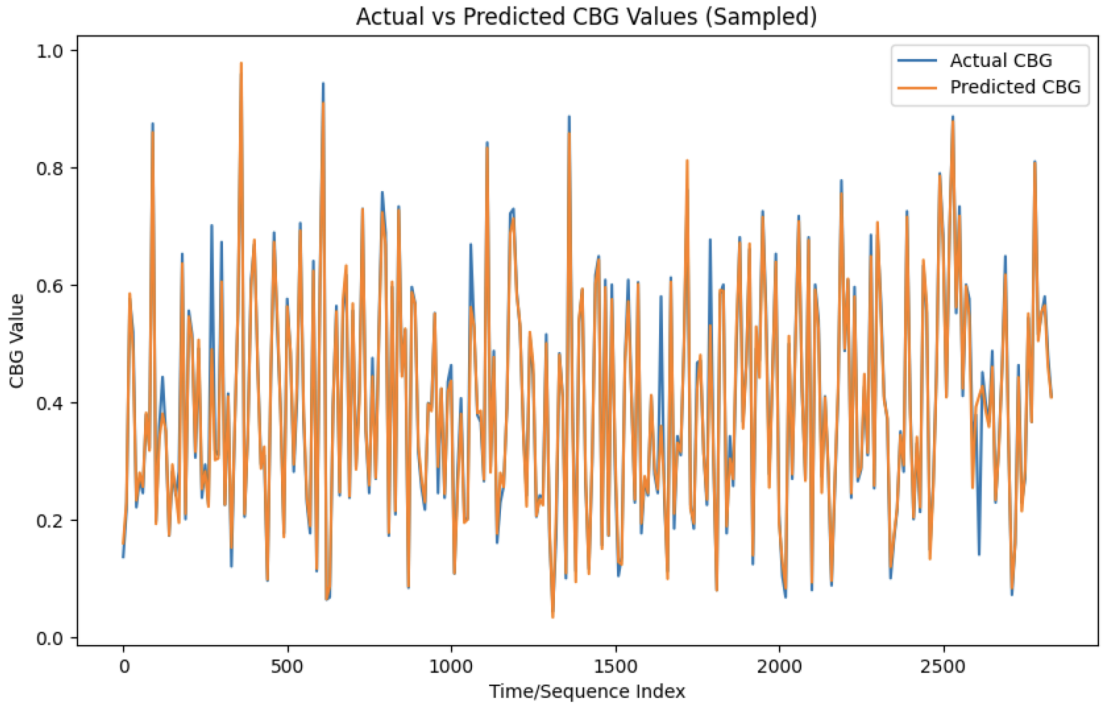
4 Federated LSTM

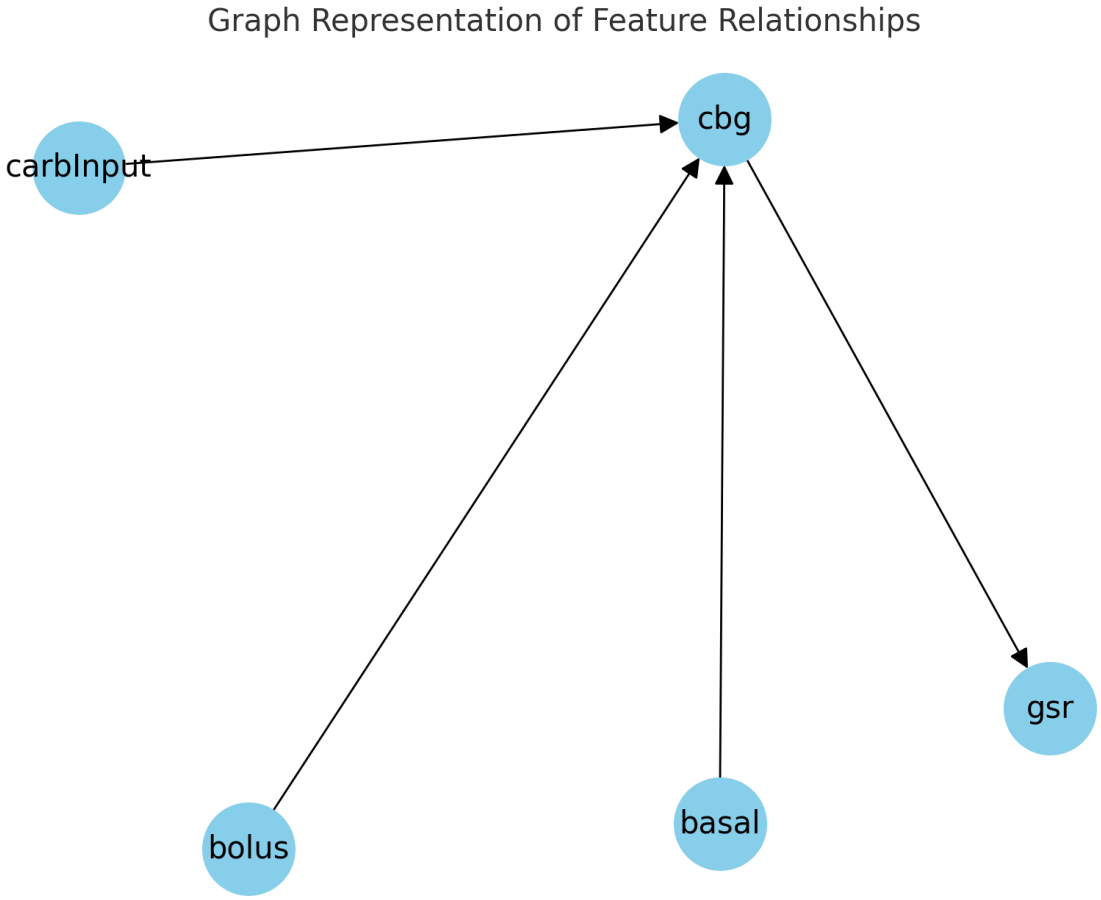
Initial attempts to implement federated learning encountered technical difficulties with the PySyft and TensorFlow Federated frameworks. The project proceeded with a custom implementation of federated learning, starting with the initialization of a global model. A subset of clients was randomly selected for each training round. During local training, each client performed multiple epochs of training on their dataset, based on the current state of the global model. This process encompassed the typical steps of a forward pass, loss computation, backward pass, and an optimizer step.

After local training, the updated model weights and average training loss were compiled from each client. These updates were then aggregated to update the global model, employing a weighted average approach. Experiments with different numbers of clients (e.g., 4 and 12) were conducted, with the 4-client configuration yielding an average MSE of 0.0005 and an average MAE of 0.011. These results suggest potential for improvement through further tuning.

5 Federated Graph Neural Network

The federated learning approach was extended to the graph neural network model, combining the methodologies of the previous sections. The model was trained over 10 rounds, with 10 epochs per round, involving 4 clients. This approach resulted in an average MSE of 0.0401 and an average MAE of 0.163 across all clients. Despite the accumulation of inaccuracies from the individual models, the integrated approach demonstrated potential for significant enhancements with additional refinement and training.


Actual vs Predicted CBG Values (Sampled)
Actual CBG
Predicted CBG
CBG Value
Time/Sequence Index
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500


Graph Representation of Feature Relationships
carbInput
cbg
bolus
basal
gsr