

# Midway Report - SugaRNN: Blood Glucose Predictions

Asa Adomatis, Andrew Plattel, Tucker Mackie

July 3, 2025

## Introduction

The goal we plan to explore is to create a predictive model that predicts future blood glucose (BG) levels. According to the Centers for Disease Control and Prevention, [more than 11.6% of Americans](#) have diabetes of some sort, including one of our teammates. However, it should be a critical topic of research and understanding (there is currently no cure for diabetes). Given this importance, we want to try to use new methods and techniques to predict BG levels to help people affected by diabetes manage their BG levels. Currently, there are devices called continuous glucose monitors (CGMs) that monitor and predict BG levels, and they are very powerful devices for managing diabetes. CGMs read an individual's BG level every 5 minutes and can predict future BG levels. However, one caveat is that CGMs do not display the expected level to the user. They just inform someone that they are predicted to be low/high (from their target BG) in 15 minutes. Our general goal is to use models to predict what the person's level of BG will be in 15 minutes. Using RNN models, CNN models, and time series forecasting, we are looking for the best and most accurate method to predict BG.

## Related Work

For our related work, we focus on three separate regions: CNNs, RNNs, and Time Series modeling. One of the largest takeaways was the type of modeling that was beneficial to use, with that being an LSTM model. [\[MSEM20\]](#) provides a great example depicting the LSTM model and constraints with this methodology. One of the largest issues with an RNN model is the struggle with long-range dependencies due to the vanishing gradient problem, making them difficult to train. To overcome this, the authors implement a long short-term memory (LSTM) architecture. The final LSTM output was passed into a fully connected neural network with two hidden layers (512 and 256 neurons) using ReLU activations and dropout (20% and 30%) to prevent overfitting. The output layer consists of two neurons—one with linear activation and one with exponential activation—used to model BG as a univariate Gaussian distribution. The study also experimented with a glucose-specific physiological loss function that combines mean squared error with a penalty for predictions that could lead to harmful clinical interventions, although they found it did not improve model performance. Data from each patient was split into 60% for training, 20% for validation (for early stopping), and 20% for testing, with hyperparameters tuned using a grid search based on RMSE. While the model incorporated additional physiological signals such as sleep, stress, and heart rate, it focused solely on predicting future BG levels, up to 60 minutes ahead. Notably, training on combined data from all patients yielded the best results, suggesting the presence of generalizable patterns in BG variability across individuals.

Our next article [\[RTH<sup>+</sup>21\]](#) presented a novel method for predicting blood glucose levels using a stacked LSTM-based deep recurrent neural network. To correct errors from continuous glucose monitors (CGMs), particularly the older and less accurate Medtronic CGM, the authors incorporated a modified Kalman smoothing technique. Their model considered four hyperparameters: CGM values, meal information, insulin doses, and step count, and they distinguished between increasing and decreasing carbohydrate absorption phases based on meal timing. The LSTM model used a stacked architecture with 128 hidden states, and results showed that this deeper design outperformed shallow models with a single layer.

When looking at Time Series modeling, this can be a crucial step to accurately predict where the next blood glucose level will be. Using [KNEB23], we were able to find some valuable information and testing methods to assist us. [KNEB23] showed one of the largest limitations in that every human is different. When creating a model, on average, a person may fit into the model, but breaking that model out into an individual aspect is not always going to provide great results. With every human being different, they had to enlist a lag model to provide more accurate results. As every lag length is going to be different, you fall into two separate buckets. *In bespoke analysis, either optimal lag values should be found for each individual separately, or a globally suboptimal lag value should be used for all. The former approach degenerates the analysis’s congruency and imposes extra perplexity. With the latter, the fine-tuned lag is not necessarily the optimum option for all individuals. To cope with this challenge, this work suggests an interconnected lag fusion framework based on nested meta-learning analysis that improves the accuracy and precision of predictions for personalized blood glucose level forecasting.* [KNEB23] Using this knowledge of lag fusion, we will be able to build better models to fit people or adjust our overarching model.

[KBSK24] provides a different twist to time series modeling, and runs a model on the nocturnal aspects for people with type one diabetes. Predicting the BG levels overnight is a different task compared to during the day. Unless you work third shift, you are sleeping where you have a constant rise and fall to your levels, but the impacts of food and exercise are not there. *The Ward algorithm in hierarchical clustering works by iteratively merging the two clusters that result in a minimum increase of the total within-cluster variance after merging.* [KBSK24] Using this Ward algorithm provides a new step to time series modeling and provides a clustering model. This method provided a lot of insight into how clustering the data impacts the result, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). A clustering method allowed for our errors to be more ideal, and to find the underlying patterns that are within the given data.

Jaloli and Cescon [JC23] present a hybrid CNN-LSTM model for predicting blood glucose (BG) levels at 30-, 60-, and 90-minute horizons, using glucose measurements, meal information, and insulin intake as inputs. The model architecture includes multiple convolutional layers: two 1D convolutional layers with a kernel size of 7, followed by a 1D max pooling layer, then another pair of 1D convolutional layers with a kernel size of 5, and a final max pooling layer. The output from the convolution and pooling layers is flattened and passed to an LSTM layer, which is followed by two fully connected dense layers, ultimately producing a sequence of future predictions up to the desired prediction horizon. Dropout layers were added between LSTM layers to mitigate overfitting. Model parameters were optimized by minimizing the mean absolute error (MAE). The final model achieved an MAE of 2.69 mg/dL, an RMSE of 9.73, and an  $R^2$  score of 91.67%. The study highlighted two key findings warranting further investigation: (1) adjusting the input sequence length based on the prediction horizon, and (2) modeling data on a per-patient basis.

## Methods

Our models are built in the [SRC+23] repository using the **Hall** dataset. Within the **Hall** dataset, this provides details every five minutes for 57 different people. This data set provides many good use cases for all three model types. RNN, CNN, and Time Series. To start to build our model, we needed to clean the dataset. The data came in a fairly clean format and did not have extreme issues. We did have several "NAs" that had to be removed to make the models more accurate, but this does result in data loss. The next step was that we had to adjust the datetime format to be uniform, to ensure accurate time frames. Without the removal of NAs and adjusting the datetime format, our models would not run correctly or produce false results. We created Training and Test datasets that will be used with an 80/20 split. This will be a crucial step in being able to test our data and ensure we can have a strong testing platform for our models. When building our RNN model, we extracted the `gl`, blood glucose, variable as our target variable to train and predict. The next steps of the training process were normalizing the data using the `MinMaxScaler()` function from `tensorflow` and flattening it so we can use it in our model in the future. After which, we created sequences and created our train/test split using an 80/20 split. Since many of the studies we researched found LSTM models performed the best, we decided to approach the problem using the same method. After initializing the model, we fit it to the history, created predictions using our test data, inverted the predictions back to normal scale, and calculated our RMSE.

As for the CNN, the model contained an input layer specifying sequences of length 60, an amount representing 5 hours of measurements, 2 pairs of 1-dimensional convolution layers with a kernel size of 7 and RELU activations and max pooling layers with a pooling size of 2, followed a 1-dimensional global max pooling layer, then 2 more dense layers with 128 and 256 units respectively. The final output predicted a window of size 5, representing 25 minutes of measurements. The model was then trained using a validation split of 30% with a batch size of 32 and trained in 200 epochs.

When starting the time series data, the first step is making sure the data is aligned with the time series correctly. As we were taking the data from a CSV file, it was not presented in the correct format. The hall dataset was formatted to fit this to make sure it can be accurately tested. When creating the first time series plot and testing this method out. We used the `daily_glucose` by taking the average glucose per day. Using this data, we were able to build a plot where we can forecast what the next 10 days of Average Daily Glucose levels will be.

## Preliminary Results

When creating our RNN models, we were able to get a working model using the **Hall** dataset. [MSEM20] Using these methods, we got a training error of 0.403 and a testing error of 0.373. Potential next steps would be performing some hyperparameter tuning to lower the RMSE and to develop a model that is trained on individuals in the data rather than a global model. Creating these individual models and making them accurate based on as few data points as possible will be a powerful tool for those with diabetes. The CNN model resulted in a Root Mean Squared Error (RMSE) of 16.57 mg/dL, a Mean Squared Error (MSE) of 274.47 mg/dL and a Mean Absolute Error (MAE) of 11.95 mg/dL. Finally, our time series model resulted in an MSE of 186.46 mg/dL and an MAE of 11.04 mg/dL. When testing the RNN model further, we got an RMSE of 15.18 mg/dL. For a baseline mode, this is better than our expected accuracy and provides us with confidence in future steps. Potential next steps would be adding in extra variables and performing some hyperparameter tuning to lower the RMSE and to develop a model that is trained on individuals in the data rather than a global model. Creating these individual models and making them accurate based on as few data points as possible will be a powerful tool for those with diabetes. We got varying results, with some of the models being better than others. We know there is a lot of improvement that can be made, as these models would be entry-level and not the best models that we will have. Below is a time series forecasting model that we have created that shows the Forecasted Average Daily Glucose Levels over time. This is the average across all people within the **Hall** dataset [MSEM20] and combines everyone instead of an individual model.

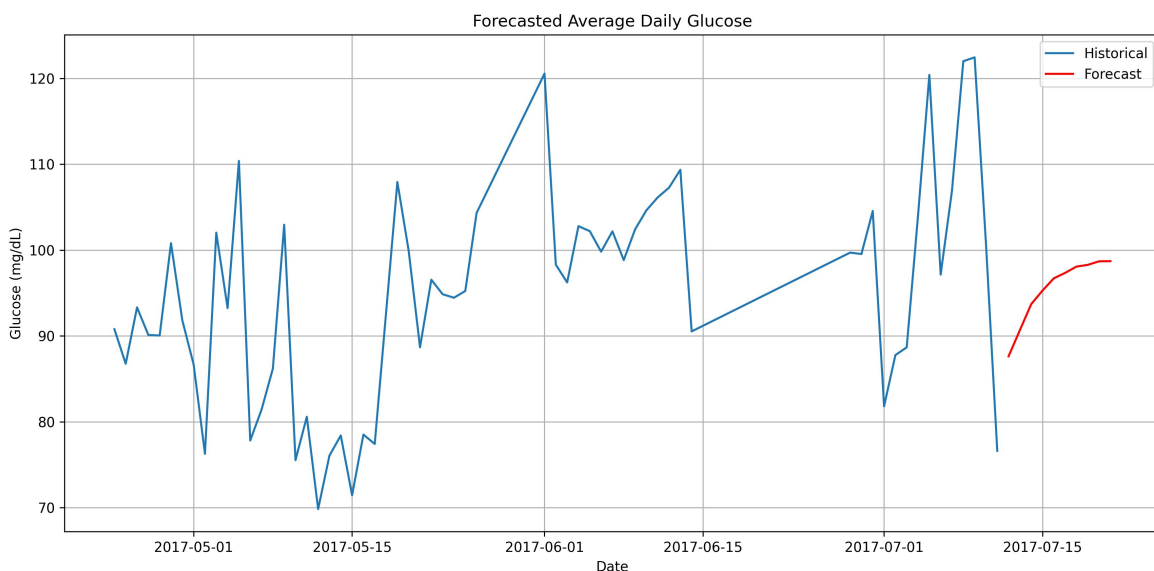


Figure 1: Forecasted Average Daily Glucose Levels

## Future Plans

There are many methods that we want to improve on for the final models. One of the largest things that we want to fix is the data we are using. The **Hall** dataset [MSEM20] includes the majority of people who do not have Diabetes or are diagnosed as pre-diabetic or diabetic later in life. The goal is to use a dataset that is predominantly people with diabetes for us to accurately track our models. Using this methodology, we would be able to test against people who have diabetes and try to predict where their blood glucose levels will be.

For the remainder of the project, we each will focus on developing one of the three models to obtain the best accuracy possible. That being said, we want to have our final and most accurate model developed by Friday, July 25th. That will give us enough time to collaborate and get on the same page with an expected presentation on Tuesday, July 29th. We should have our final repository uploaded and public by Monday, July 28th. We should plan on meeting that Monday night to finalize the slides and report. Then we plan on submitting the final report by Friday, August 1st.

## References

- [JC23] Mehrad Jaloli and Marzia Cescon. Long-term prediction of blood glucose levels in type 1 diabetes using a cnn-lstm-based deep neural network. *Journal of Diabetes Science and Technology*, 2023.
- [KBSK24] Danil E. Kladov, Vladimir B. Berikov, Julia F. Semenova, and Vadim V. Klimontov. Machine learning algorithms based on time series pre-clustering for nocturnal glucose prediction in people with type 1 diabetes. *Diagnostics*, 2024.
- [KNEB23] Heydar Khadem, Hoda Nemat, Jackie Elliott, and Mohammed Benaissa. Blood glucose level time series forecasting: Nested deep ensemble learning lag fusion. *Journal Article*, 2023.
- [MSEM20] John Martinsson, Alexander Schliep, Björn Eliasson, and Olof Mogren. Blood glucose prediction with variance estimation using recurrent neural networks. *Journal of Healthcare Informatics Research*, 2020.
- [RTH<sup>+</sup>21] Md Fazle Rabby, Yazhou Tu, Md Imran Hossen, Insup Lee, Anthony S. Maida, and Xiali Hei. Stacked lstm based deep recurrent neural network with kalman smoothing for blood glucose prediction. *BMC Medical Informatics and Decision Making*, 2021.
- [SRC<sup>+</sup>23] Renat Sergazinov, Valeriya Rogovchenko, Elizabeth Chun, Nathaniel Fernandes, and Irina Gaynanova. Glucobench: Curated list of continuous glucose monitoring datasets with prediction benchmarks. *arXiv*, 2023.