Meal Nutrition Analysis

Sreedhar Reddy Pacharla, Chetan Sai Borra

***Abstract*—In today's health-conscious society, there is a high demand for applications that monitor food consumption and intake of nutrients to avoid malnutrition and prevent vitamin deficiencies. Hence diet monitoring has become a prominently researched topic due to its potential to improve public health. In this project, we have developed a diet-monitoring machine learning model that predicts calorie intake based on inputs gathered from devices such as continuous glucose monitors (CGMs), food photographs, and individual user health, physical characteristics and gut micro-biota.**

***Index Terms*—Continuous glucose monitors, Machine Learning, post-prandial glucose response.**

I. Introduction

There is a need to monitor the food consumed by an individual because most diseases and health risks like diabetes, obesity, etc., are due to uneven consumption of food. In older methods, individuals used to note down the details of their food intake in a register. Many individuals find this task burdensome and may skip logging their food intake on certain days, leading to inaccurate data if they forget what they ate at specific times. Therefore, continuous monitoring of food consumption is necessary to accurately track dietary details. To address these challenges, numerous advanced techniques are being developed. In the journal [1], the authors discuss new technologies and primarily review advances in three key areas: mobile applications; wearable and handheld sensors; and innovative applications that generate personalized nutrition recommendations based on gut micro-biota and continuous glucose monitoring enhanced by artificial intelligence. By utilizing these technologies, individuals can receive personalized food recommendations tailored to their specific needs, leading to significant improvements in health. This personalized approach helps in improving dietary habits, thereby reducing the risk of diet-related diseases.

Leveraging these new technologies, this project focuses on estimating calorie intake based on CGM data, food photographs, demographic information, a range of physical characteristics like BMI, and gut health. A Continuous Glucose Monitor (CGM) is a device that continuously measures glucose levels in the interstitial fluid, allowing us to monitor fluctuations in these levels in real time. The given CGM data recorded glucose levels every five minutes from morning to evening, providing continuous monitoring of glucose fluctuations throughout the day. In addition, we were provided with photographs of the food served during breakfast and lunch, additional demographic data and gut health information, which will be described in further sections

In the following sections, we will discuss the two methodologies we have implemented, detailing their implementation, the experiments conducted, and the results obtained.

II. Methods

In this project, we implement two main methods. The first method focuses primarily on CGM data, demographic information, and gut health. After eating, glucose levels fluctuate, and these fluctuations vary depending on the content of the food consumed. This response to a meal is called the postprandial glucose response (PPGR). The PPGR depends on macronutrients such as carbohydrates, fats, and proteins. Our approach to using CGM features to predict macronutrient intake is inspired by [2].In [2], the experiment was conducted by having individuals fast for at least 8 hours, and after consuming the provided meal, they were instructed not to eat anything for another 8 hours. In contrast, in this experiment, individuals were given breakfast in the morning, and after at least 3 hours, they were provided with lunch. In [2], the first blood glucose reading was taken as a baseline since the individuals hadn't eaten anything prior. However, in our experiment, the behavior of the postprandial glucose response (PPGR) after lunch depends on the breakfast consumed and the timing of when breakfast and lunch were given. For example,

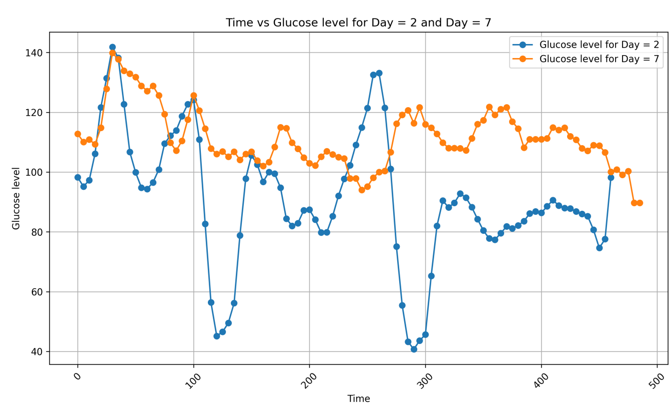

Fig. 1. Glucose levels of subject id = 1 on Day 2 and Day 7

in Figure 1, for the same individual on days 2 and 7, the same breakfast and lunch were provided, but the times at which the meals were given were different. Despite consuming the same food, it can be observed that the PPGRs differ between the two days. This indicates that the PPGR varies based on the timing of previous food consumption. To capture these variations, we extracted 10 features from the CGM data: the maximum of the peaks before lunch and after lunch; the mean value of the peaks before and after lunch; the average rise time and maximum rise time before and after lunch; and the number of peaks before and after lunch. In [2], similar features were extracted using a Kalman filter and the Gaussian area-under-the-curve (gAUC). In addition to CGM data, we considered demographic and Viome data as input features because metabolism varies between individuals and depends on factors like gut health. We also assume that metabolism depends on demographic data as well. Later, by analyzing the weights in our results, we can confirm whether these features significantly impact the predictions. Our methodology consists of three main stages: first, feature extraction, where we extract features from the CGM data and other specified inputs; second, using these features in a linear regression model to estimate macronutrients such as carbohydrates, fats, and proteins; and third, passing these estimated macronutrient values to another linear regression model to predict the calories consumed during lunch.

The second method shows the advanced strategy of data integration in which CGM data is integrated with photos of food taken during breakfast and lunch as well as other demographic and gut health data. One of the notable aspect is that the cyclical encoding of CGM time features which helps in capturing the fact that the glucose readings are taken every 5 minutes. This encoding converts temporal data in that a sine and cosine functions are used to encode the measurements so that temporal correlations between the data points are not lost or affected in any way; values at the end of one cycle are connected cycle. to The the approach beginning is of especially another important for the analysis of CGM data as it preserves the continuity of glucose measurements while recognising variability in glucose fluctuations on a daily basis. The following equations illustrate the method used to encode time in hours and minutes, effectively capturing the cyclical nature of time [3].

$$\theta = \left(\frac{t}{cyclingperiodtime}\right) \times 2\pi$$

$$sint = \sin(\theta)$$

$$cost = \cos(\theta)$$

The method also incorporates food images and the microbiome data to allow for the analysis of the interconnection between diet, glucose response and metabolic well-being. Demographic data also provide an opportunity to extend the analysis and make it more specific to an individual, taking into account differences in glucose metabolism and personal variation in the patterns of glucose response. This is because by using more than one modality, a richer and therefore, more accurate and detailed understanding of metabolic health can be obtained.

III. Experiments

A. Experiment - 1

This experiment is based on the first method discussed in the Methods section and illustrated in Figure 2. Before starting the experiment, we first need to retrieve the data from the CSV files, remove any rows with missing data, and prepare the data for analysis. Once the data is ready, we begin the first step of the experiment: the extraction of features from the CGM data.

For each day, we compute a total of 10 features from the CGM data. We calculate the following five features before lunch:

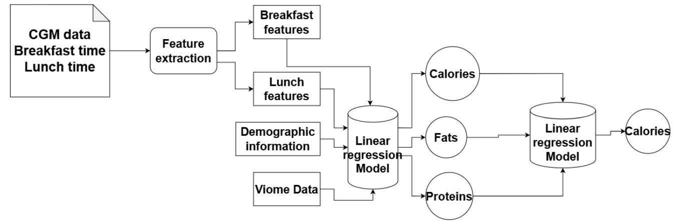

Fig. 2. Methodology for Experiment-1

1) Maximum value among the peaks of glucose levels
2) Average of the peaks of glucose levels
3) Number of peaks
4) Average time to reach the peaks
5) Maximum time to reach a peak

Similarly, these five features are calculated after lunch. These features are categorized as breakfast features (before lunch) and lunch features (after lunch). Before employing this type of feature extraction, we conducted an experiment where we used only the sum of spikes. Here, the term 'spike' refers to the difference between a local maximum (peak) and the preceding adjacent local minimum. This experiment didn't yield good results because the sum of spikes alone was insufficient to capture the complex patterns of glucose fluctuations associated with food intake. It failed to account for factors such as the timing of meals, the duration of glucose level changes, and the overall trend in glucose variations, leading to poor prediction results.

After extracting the features, we proceed to the second stage, where we train a linear regression model. The inputs to the model include the previously extracted features, demographic information, and gut health data from Viome. The outputs are three labels: lunch calories, fats, and proteins. While it may seem that we have provided numerous inputs, we will discuss in the results section which features are truly significant based on their weights.

In the third stage of the experiment, we trained another linear regression model using the three labels: lunch fats, proteins, and carbohydrates as inputs and lunch calories as the output. We do not require these labels during testing, as they are estimated from the first linear regression model.

In the final stage of the experiment, we tested the model using the test CGM data, test demographic information, and test Viome data. We observed that in the test CGM data, breakfast and lunch times were missing in some instances. To address this, we filled the missing values based on adjacent times or previous entries. For example, if the breakfast time was missing, we estimated it by subtracting three hours from the lunch time. If the lunch time was missing, we estimated it by adding three hours to the breakfast time. In cases where both times were missing, we used the values from the previous entry. Following this, we extracted features in the same manner as in the first stage of the process. These features, along with the other data, were passed to the model to estimate the macronutrients. The estimated macronutrients were then input into another model that predicts the target output: lunch calories.

In our results, we found that this experiment produced sufficient outcomes to surpass the minimum benchmark. However, this experiment is not the best option for all cases. The features we extracted in the first stage do not entirely capture the complex nature of the CGM data. Additionally, the linear regression model may not adequately represent the complex functions required to produce accurate outputs. To address these challenges, we implemented another model that uses neural networks to efficiently predict the labels.

B. Experiment - 2

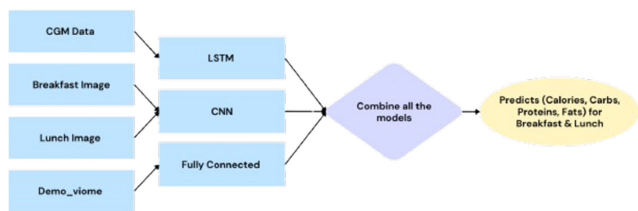

Fig. 3. Methodology for Experiment-2

This experiment is an extension of the second method in the Methods section and is represented in Figure 3. The CGM data has to be cyclically encoded in order not to lose the temporal nature of the features. In particular, the timestamp is encoded using sine and cosine transformations, which provides four temporal features: hour-sin, hour-cos, minute-sin, and minute-cos. In addition to these, the glucose values provide the final set of five features that are normalized by z-score normalization.

The dataset is divided into an 80:20 ratio for training and validation. The encoded CGM data will be fed into an LSTM model, which consists of 64 hidden units and returns an output layer of 32 features. In parallel, images of breakfast and lunch meals are passed through two different CNNs, each with a depth of 128 in the convolution layers. Both CNNs generate 16-dimensional outputs through fully connected layers.

Additionally, demographic and gut microbiome data are fed into fully connected layers with a depth of 32, producing 16-dimensional outputs. The outputs from the LSTM, the two CNNs, and the fully connected layers are concatenated to form a unified representation. This unified output is then fed through another fully connected layer to predict the final labels. The model predicts eight output variables on the caloric content, carbohydrate, protein, and fat levels concerning breakfast and lunch. This framework integrates temporal, visual, and structured data in a collaborative way to improve the prediction results.

The performance of the model was further improved by hyperparameter tuning to ensure that the architecture for each modality of data was optimal. The most important hyperparameters included the number of layers in LSTM and CNN models, the size of hidden layers, dropout rates, and the learning rate for the optimizer. The LSTM was done by exploring a range of configurations of hidden units and layer depths so that the ideal balance between underfitting and overfitting can be achieved. This would ensure better capturing of temporal dependencies in CGM data. Similar to this, CNN architecture has been tuned by changing the number of convolutional filters, kernel sizes, and pooling strategies with the aim of extracting useful features from meal images.

Dropout layers were added to avoid overfitting, and the dropout rate varied in the experiment to find the most robust value. The learning rate is one of the most important parameters in the Adam optimizer, tuned across multiple runs for stable and efficient convergence of the model during training.

During the tuning process, each model configuration was run against benchmark loss metric RMSRE. Almost all the tuned configurations outperformed the baseline benchmarks, indicating the effectiveness of the parameter search and the architecture's adaptability to the multi-modal nature of the dataset. This systematic approach to hyperparameter optimization played a very important role in achieving superior predictions for the target labels.

IV. RESULTS

In Experiment 1, the linear regression model was implemented using the PyTorch library. The model employed the Adam optimizer and used RMSE as its loss function. It was trained for 30,000 epochs. After training, the model achieved a training loss of 0.4671 and a validation loss of 0.5449. The progression of these losses over the training iterations is illustrated in Figure 4.

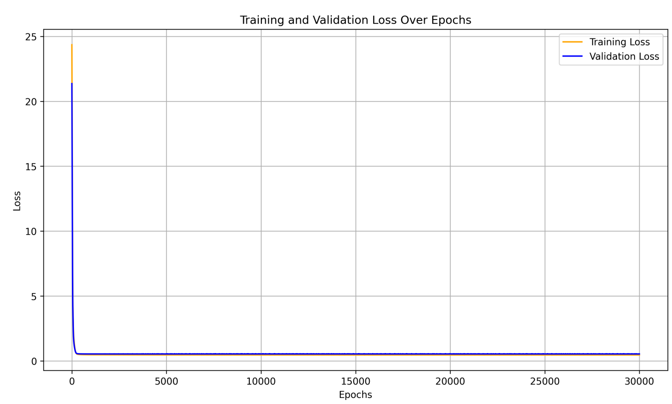

Fig. 4. Training and Validation Loss over iterations for first Linear regression model

For the second model, we used the same optimizer and the same number of epochs. After training, the model achieved a training loss of 0.4276 and a validation loss of 0.4477. The progression of these losses over the training iterations is illustrated in Figure 5. This trained model was used to predict calorie intake, and the observed loss value on the test data was 0.5054.

In Experiment 2, the multimodal network was implemented in PyTorch, performing a systematic hyperparameter tuning.

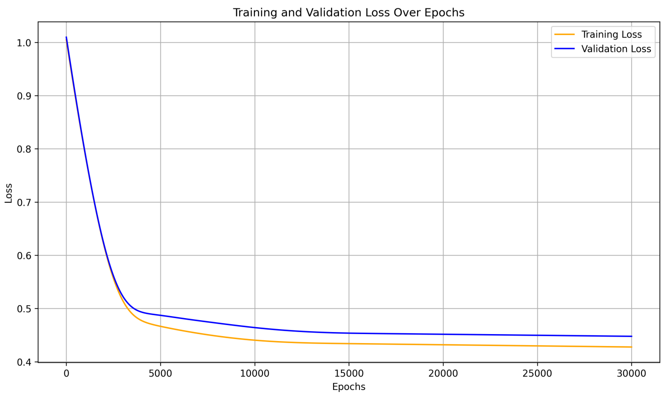

Fig. 5. Training and Validation Loss over iterations for Second Linear regression model

The model performance was evaluated by the Root Mean Squared Relative Error loss function.

The training was set to run for 1500 epochs initially. After the examination of the training and validation loss curves, the number of optimal epochs was 300. The learning rate after testing a few values was found to be 0.0001.

Further tuning was performed for hyperparameters like dropout probability, weight decay, and the number of hidden layers in LSTM and CNN sub-models. After a lot of experimenting, the best performance had the following configuration:- Learning rate: 0.0001 - Dropout probability: 0.5 - LSTM hidden layers: 64 - Weight decay: 1e-3

These are some of the parameters for the lowest loss during training. Several minor changes were made in order to finalize the model. Below is a plot showing the training loss and the validation loss curves for the tuned model:

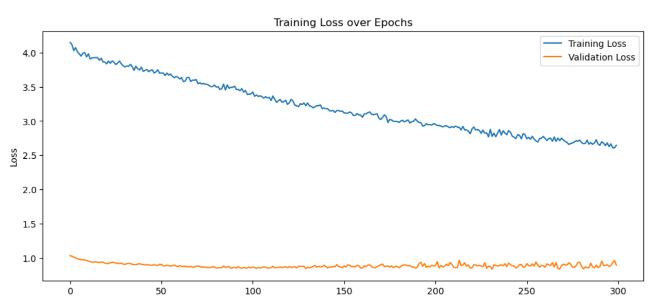

Fig. 6. Training and Validation Loss over epochs.

The training loss of the model is about 1.7, while the validation loss is about 0.83. Since the loss of validation is lower than the training loss, this means the model generalizes well and performs well on data it has not seen.

The model also manages to predict all eight target labels: breakfast and lunch calories, carbohydrates, proteins, and fats. With the main objective of predicting lunch calories, it does not go out of the range between 0.34 and 0.36 on the test dataset. This already exceeds the benchmark set, hence proving the accuracy and efficiency of the model in reaching the desired predictions.

V. Implementation

This section discusses the contributions of our team members to the project. First, we want to state that both of us contributed equally to the project, and the details below convey each individual's specific contributions.

Sreedhar is responsible for Experiment-1 implementation and method. He reviewed research papers to understand the important features of CGM data and the dependence of the postprandial glucose response (PPGR) on macronutrients. He fully implemented the model for Experiment 1 and conducted various experiments to improve the results. He wrote the abstract, Introduction section, detailed his methodology, and documented his experiment and their results in the report.

Chetan Sai implemented Experiment 2 and developed the methodology. He conducted an in-depth review of research papers in order to explore techniques for encoding the time features of CGM data effectively. He then used this knowledge to design and fully implement the model for Experiment 2, carefully conducting a series of experiments to optimize its performance. Chetan documented his approach, detailing the methodology and providing comprehensive reports on the experiments conducted and their respective results.

VI. Conclusion

The second experiment presented in this work showed the strong generalization and outperformed the benchmark regarding lunch calorie prediction by utilizing a multimodal neural network, which combines multiple data types: CGM data on glucose levels, meal images for visual analysis, and demographic and gut microbiome information to further enhance the model's understanding. The model was able to capture intricate relationships and patterns in the data by utilizing LSTM for temporal dependencies and CNNs for meal image analysis. This led to an improvement in the predictive performance.

Even though the results from the above models are promising, their performance can be further enhanced. An important enhancement might be the addition of attention layers. Attention mechanisms allow the model to focus more selectively on the important features or time steps, effectively enhancing its ability to capture and emphasize relevant information while ignoring less informative data. In tasks like this, where multiple modalities are present, attention layers can be helpful in learning for the model which parts of the data-for example, specific meal images or glucose fluctuations at certain times-are more critical for accurate predictions. This could result in more accurate predictions, especially in domains like personalized nutrition.

References

[1] Mortazavi BJ, Gutierrez-Osuna R. A Review of Digital Innovations for Diet Monitoring and Precision Nutrition. Journal of Diabetes Science and Technology. 2023;17(1):217-223. doi:10.1177/19322968211041356

[2] S. Sajjadi et al., "Towards The Development of Subject-Independent Inverse Metabolic Models," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 3970-3974, doi: 10.1109/ICASSP39728.2021.9413829. keywords: Computational modeling;Standardization;Machine learning;Signal processing;Predictive models;Glucose;Indexes;Continuous glucose monitors;diet monitoring;meal macronutrients;machine learning,

[3] Chen, J.; Yang, Z. Revolutionizing Time Series Data Preprocessing with a Novel Cycling Layer in Self-Attention Mechanisms. Appl. Sci. 2024, 14, 8922. https:// doi.org/10.3390/app14198922