4-21-2023

Physiological Signals and the Effects on Prediction of Future Blood Glucose Values in a Deep Learning Model

Andrew Rearson
amr8659@rit.edu

RIT

Physiological Signals and the Effects on Prediction of Future Blood Glucose Values in a Deep Learning Model

By Andrew Rearson

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Mechanical Engineering

Department of Mechanical Engineering

Kate Gleason College of Engineering

Rochester Institute of Technology

Rochester, New York

April 21, 2023

Committee Approval:

Dr. Kathleen Lamkin-Kennard Date

Thesis Advisor

Dr. Jason Kolodziej Date

Committee Member

Dr. Jamison Heard Date

Committee Member

Dr. Michael Schrlau Date

Department Representative

1 TABLE OF CONTENTS

2 TABLE REFERENCES

3 FIGURE REFERENCES

4 NOMENCLATURE AND DEFINITIONS

CGM	Continuous Glucose Monitor
AI/ML/DL	Artificial Intelligence/Machine Learning/ Deep Learning
IDDM	Insulin-Dependent Diabetes Mellitus—Also known as type 1 diabetes
Type 2 Diabetes	A chronic condition where the body does not use insulin properly.
Mathematical-Model / Model	A numerical function that simulates a system's behavior when provided an input.
Layers/Nodes	A deep learning model architecture consists of layers of nodes. Each layer contains nodes that interconnect with the surrounding layers.
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory (Specialized RNN layer)
puid/PUID	Participant Unique Identifier—a number associated with the participants in the OhioT1DM [1] dataset.
Feature	A feature is an individual measurable property or characteristic of a phenomenon. [2]
Health metrics & signals	A data component that measures or represents something found in nature.
Epoch	When a model passes over an entire data set, this metric is commonly used to count the number of passes a model makes on a data set.
Accuracy	A metric or measurement that represents model performance.
Tuning	The process of adjusting variables with the intent of improving performance.
RMSE	Root Mean Square Error
ABE	Active Burned Energy –defined as cumulative count of total calories burned per day [starting at midnight].

5 ABSTRACT

Diabetes is a chronic disease that currently has no cure. However, in the last decade, life-changing technology for people with diabetes has advanced primarily due to new sensors that continuously measure glucose. Individuals with diabetes manually use this data for diabetes management, but the artificial intelligence community has seen the increase in data as an opportunity to work towards automating diabetes control. The project developed deep-learning neural networks incorporating blood glucose measurements measured by a continuous glucose monitor (CGM) and physical activity data from a wearable health sensor (WHS) to predict future glucose values and compare how different configuration variables, such as input length or health features, impacted prediction accuracy. Model accuracies were compared using root mean squared error (RMSE). The data used to train and test the prediction models was from the OhioT1DM 2018 dataset that contains physiological signals collected from a WHS and blood glucose measurements from a CGM. The dataset was processed and organized into input and output dimensions using a custom created configuration file. The prediction model used was a deep learning Long Short-Term Neural Network (LSTM). The model was trained on all participants but tested on each participant individually by comparing experimental and measured blood glucose predictions at 0.5 and 1.0 hours.The findings suggested increased input length could improve predictions, but additional health features did not. All the health features considered, including insulin dosing, unexpectedly decreased prediction accuracy. A novel approach that compared predictions from single-value and series RMSE calculations showed that the series output approach provided additional context about how the models fit the data. Future research should address how results could be compared across literature studies and focus on event-based feature extraction from WHS.

6 PROBLEM INTRODUCTION

6.1 DIABETES MONITORING AND MANAGEMENT

One of the most significant concerns for individuals with diabetes is experiencing hypoglycemia at night [3]. When someone with insulin-dependent diabetes mellitus (IDDM) experiences hypoglycemia, the glucose in their bloodstream is below the threshold the body needs to function correctly. According to Amiel [3], “Acute hypoglycaemia stimulates a stress response that acts to restore circulating glucose, but plasma glucose concentrations can still fall too low to sustain normal brain function and cardiac rhythm.” In the last ten years, blood glucose monitoring technologies have drastically increased in quality and reliability since each decision related to blood glucose monitoring is potentially high stakes. Blood glucose monitoring measurements provide a more detailed picture of diabetes control with each improvement. Continuous glucose monitoring (CGM), with devices like the Dexcom G7 (Dexcom, Inc., San Diego, CA), is the most significant technological improvement. CGM technology utilizes a small plastic component that inserts a tiny clear canula into the interstitial fluid of a user's abdomen. The tiny sensor from Dexcom is more accurate than the historical control method of measuring blood glucose from a finger prick blood sample [4]. Dexcom recommends using the 'rule of 20' where below 80, the reading could be ±20mg/dL, and above 80mg/dL, the measurement is within 20%.

Continuous glucose monitoring has impacted people's lives with diabetes in more ways than one. The primary advantage of CGM is that it decreases the number of blood tests a person with diabetes must perform daily. Finger-pricks are considered one of the worst parts of diabetes management as they hurt, create calluses on the user's fingers, and are inconvenient throughout the day. In addition, finger-prick measurements are a one-time data point for the person with diabetes and give no context as to whether their blood glucose is rising or dropping rapidly. Alternatively, CGM sensors provide an accurate measurement every 5 min, 24 hours a day, without user intervention. The data is transmitted directly to a system receiver, an insulin pump, and the user's phone. Connected applications give the patient visual and auditory alerts if glucose is trending down quickly or quickly rising. In addition to real-time blood glucose data, users can

evaluate data trends using an app that collates the data and uses pattern evaluation for suggested treatment modifications.

It is important to note that the CGMs are not measuring blood glucose values because the CGM cannula collects information from the interstitial fluid. Interstitial fluid is closer to the skin's surface, which means that the measurement is not the amount of glucose in the blood, but the amount of glucose absorbed from the blood. However, to remain consistent with the literature and popular terminology, the term *blood glucose* was be used in this thesis to describe values reported from the CGM.

The revolutionary improvements for people with diabetes do not stop there. Continuous glucose monitoring data has led to the development of other quality-of-life and lifesaving technologies, such as closed-loop solutions that connect a CGM transmitter directly to an insulin pump to adjust insulin delivery based on current glucose measurements automatically. In recent news, Insulet (Acton, MA), the maker of the diabetes insulin pump Omnipod, was approved for a closed-loop system to control basal insulin rates [5, 6]. The approaches to improving people's lives with diabetes have evolved drastically due to advances in CGM technologies, but unpredictable nightly lows are not yet a thing of the past.

One of the variables in diabetes control depends on activity levels, such as exercise. When anyone, with diabetes or not, is active consistently, their body uses available blood glucose more efficiently than someone who is less active. Efficient glucose use is vital for people with diabetes because when their body can use available energy more effectively, their blood glucose is easier to control and keep within a target range, requiring less exogenous insulin. However, while easier to maintain, activity also increases the chance that individuals with diabetes could have hypoglycemia due to increased metabolic activity. The other issue with increased activity levels is that the effects on an individual's blood glucose are not necessarily immediate and can rise several hours later and often late into the night [7].

Recent advances in wearable health sensors, such as Apple (Cupertino, CA), Fitbit (San Francisco, CA), and Garmin (Garmin Ltd., Olathe, KS) watch models have multiple sensors that continuously measure physiological signals. The measurements are commonly saved on the

user's phone in an app or a health database, such as the Health app on iPhones, and provide a historical look at a user's daily activity trends.

This thesis project analyzes various feature combinations of physiological signals and time periods obtained from wearable health sensors (WHS) to understand better what features could improve blood glucose predictions. The predictions generated in this project integrate machine learning algorithms and activity data collected from smart health wearables and blood glucose data collected from sensors like the Dexcom G6. The outputs from this study could help improve the process of optimizing models for blood glucose prediction by identifying how key features input into machine learning models affect and impact prediction accuracy. Optimized blood glucose is an essential step in the automation of diabetes control. Furthermore, removing the need for constant user interaction in diabetes management is a high priority for many individuals with diabetes [8].

6.2 MACHINE LEARNING

Computers and computational potential have advanced significantly in the last 20 years. Artificial Intelligence (AI) is a broad topic that focuses on how computers adapt to provided information. Machine learning (ML) is a subtopic of AI that focuses on data-driven algorithm generation. The algorithms are mathematical models designed to represent and replicate some systems' functions. The models allow researchers to quickly predict system outputs in millions of ways without directly interacting with and impacting the system of interest.

The most crucial component of a mathematical model is the data that defines the model. Developing mathematical models, particularly for ML, requires significant amounts of data. The type and quality of data that goes into a model directly impacts how the model performs. Sometimes the data that produces the best model is not apparent. Determining the best data to generate a model is an iterative process of comparing results. Quantifying a successful mathematical model depends on many factors, such as the use case of the model and its expected outputs. The models developed in this project take a time series of physiological signals and output a time-series prediction of blood glucose values. Quantifying the creation of a successful or high-quality model for this project requires comparing predicted blood glucose values to

expected values. The expected values for this project are the input data set, actual blood glucose values, that are then used to produce an accuracy metric.

Perfect accuracy is not always the goal when designing a model. A model's fit is an additional consideration when defining the accuracy and expectations for how a model performs. A model's fit describes how well a model handles new data. There are two main types of fit: general and specific. A general fit represents a model that can more accurately handle outliers or values that the model has never seen before based on the model's accuracy measurement. This type of fit often comes at the cost of accuracy.

On the other hand, when a model is very good at identifying inputs that the model has already handled, the fit is described as specific. While a specific fit model is very good at handling known inputs, it is more likely to struggle with inputs it has never seen. An extreme case of specific fit is called overfitting, which occurs when the model can only classify data that it has seen before and cannot handle data previously introduced. Both specific and general models have benefits and drawbacks, and each has its use cases. The balance between general and specific is vital when working with physiological signals, like in this project. The model in this project needs to be general enough to produce high accuracy on new data but specific enough to identify unique trends or specific events, such as insulin doses that cause blood glucose to drop quickly.

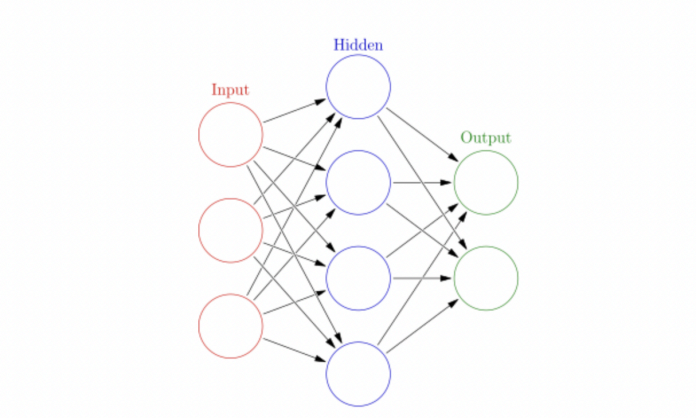


Figure 1. Visual Representation of a series of 3 interconnected layers from [9].

Deep learning is a subset of machine learning that fits data by adjusting an array of tunable units or nodes commonly referred to as neurons. Neurons are organized into interconnected layers, shown in Figure 1, creating a network of nodes where each node connects to the nodes in surrounding layers. Specific nodes will activate and pass values to a surrounding layer depending on the magnitude of the values passed through the layer of nodes.

Figure 2. A representation of the connections that lead into a node and how the output value is calculated and adjusted by an activation function [9].

An activation function can modify values as information passes from one layer to the next, as shown in Figure 2. The activation function modifies the values to give the model more control in each tuning step without significantly increasing the computational load. Activation functions can be either binary, linear, or non-linear. Rectified Linear-Unit (ReLU), Sigmoid, and Tanh functions are the most common activation functions. A ReLU activation function is a piece-wise function that linearly maps values onto a positive domain, and negative values become zero. A sigmoid activation function applies an S-shaped distribution and maps outputs from zero to one. The 'S' shape helps smooth the outputs into a more binary shape where the values trend toward 0 or +1. A Tanh function also applies an S-shaped distribution but maps values between a range of -1 and +1.

As data passes into a model, the weights between neuron connections are adjusted to impact the model's output. There are complex functions that adjust neuron connections. These complex functions are called optimizers. The most common deep learning optimizer is the Adam Optimizer [10]. Optimizer functions adjust and optimize the connection weights during model development by minimizing a loss function. The loss functions are defined to decrease as model accuracy improves. While a loss function can be the same metric that measures accuracy, the loss function is often defined differently to help the optimizer focus on specific output components.

During the fitting process, commonly called training, deep learning models can implement supervised, unsupervised, or reinforcement learning. Supervised learning is when the inputs to develop the model have outputs that have either pre-defined solutions or an observer indicates whether the outputs are successful, as determined by a metric representative of the expected outputs. The opposite of supervised learning is unsupervised learning, which lets the model identify patterns without specific guidance. The accuracy or success of these models sometimes cannot be measured. A simple example would be a model that organizes sets of seemingly random items into groups. Again, there is no success metric; the model performs this task. Reinforcement learning is the more bio-mimetic style of learning. As the model trains, it is given rewards and punishments for its outputs. The learning is similar to how human brains learn; the brain releases various chemicals as the reward and punishment mechanism for learning.

Deep learning models often also contain a unique, more computationally complex algorithm layer that is the focal point of the model. The layers for the model are generally modified around this algorithm layer to help support it. Two popular deep learning algorithms are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). A CNN is a network primarily for applying classifications to input data and is often applied to image identification and object detection. RNNs contain state components that allow them to track or remember information about previous data. A Long Short-Term Memory (LSTM) network is the RNN most used for time series or flowing data prediction, such as next-word or temperature predictions from weather data. This thesis project focuses on implementing an LSTM layer

because of the relationship between the blood glucose data and time. The features and data in this project are like stock market trends or weather data, which are common problems where LSTM cells have been applied.

The beginning and end of a model are the input and output layers, respectively. The layers pass input data into the model and retrieve the resulting outputs. The shape and type of data passed into the model determine the shape of the layers. Furthermore, the shape of the final layers in a model determines the output shape. For this study, the input data is a two-dimensional array of time-steps and feature values. The output is also a two-dimensional array of time-steps but with a single feature, predicted blood glucose values. An LSTM network does not update its states during every input step but waits for a certain number of time-steps before analyzing and adjusting the model's outputs. Each LSTM cell's internal historical state values update during each time-step until the final step. The process allows the model to keep track of previous data steps. The model updates its connection weights based on the resulting output at the final time-step.

The number of nodes, optimizer, layer order, and activation functions are variables called hyperparameters. There are many hyperparameters, each impacting the model, and tuning these values is an iterative process in model development. Hyperparameters are like seasonings in cooking; some seasonings dramatically impact the dish's flavor, while others have a less significant impact. The perfect dish has just enough of each. Fine-tuning hyperparameters is a tedious balancing act but a critical process to optimize a model's performance, speed, and fit.

Time to train and time to produce results are important considerations when developing mathematical models in general. The advantage of deep learning is that the predictions generally take less time than other machine learning applications, but the training and fitting takes significantly longer. While this thesis project does not explicitly focus on optimizing models for superior accuracy, some parameter tuning is always required to help the model produce reasonable outputs in a reasonable amount of time. Data size is one of the most significant factors related to training time. More data takes more time to train; however, not all data is provided to a model to train. An entire dataset is usually divided into training, validating, and testing sets. The training set is usually the largest and is used to tune the model. After a complete

pass of the training set, called an epoch, the model evaluates the validation set to provide insight into how the training process is going. The test set is data the model has not seen before and is used to evaluate its performance based on an accuracy metric. Again, a model's accuracy depends on the model's design criteria. The general rule for machine learning is that more training data will result in a more accurate model. However, more data can also result in an overly generalized model that cannot handle unique situations. LSTMs work better with linear data because the order of data is crucially relevant.

The evaluation metric used in this project to evaluate a model's performance is root mean squared error (RMSE). RMSE is a metric that compares the distance between each output value and the corresponding truth value, like mean squared error (MSE), except the focus of the RMSE calculations is to emphasize significant differences between predicted and actual values. The RMSE calculation takes the square root of the sum of the squared differences in predictions (Pred) and actual values (Actual) over the total value count (N) such that

$$RMSE = \left[\sum_{i=1}^{N} \frac{(Pred - Actual)^2}{N}\right]^{\frac{1}{2}} \qquad 1$$

6.3 RESEARCH QUESTION

The thesis project seeks to answer the following research question: *Using historical health data collected from daily wearable technology, how does the accuracy of an artificial intelligence algorithm that predicts future blood glucose values change with various data feature combinations?*

The project used machine learning algorithms and historically acquired health data from publicly available databases to predict blood glucose levels in diabetic individuals. The objectives of the project are:

1. To determine how the length of the time series input into the model (input length) and the length of time the model predicts into the future prediction (output length) impact the prediction accuracy of a machine-learning model.
2. To understand how physiological signals and health metrics affect the prediction accuracy of a machine-learning model.

3. To understand the value and the limitations of error metrics calculated from a series of predicted points and error metrics calculated from a single predicted point.

RMSE is the primary measurement of "accuracy" for the objectives of this project.

6.4 LITERATURE REVIEW

6.4.1 Introduction

In this project, the focus patient population was individuals with IDDM (type 1) diabetes who wear a CGM and a fitness watch that outputs activity data into the Apple Health app. Insulin-dependent diabetes mellitus (IDDM) is an autoimmune condition that occurs when pancreatic beta cells are damaged and can no longer produce insulin. Hyperglycemia results from insufficient insulin production. Many studies, found through a literature review of machine learning and diabetes, are related to detecting type 2 diabetes. Type 2 diabetes occurs when an individual has insulin resistance, and the insulin an individual's body produces is no longer efficiently used by the individual's cells. Hyperglycemia occurs when cells cannot use the endogenous insulin despite over-production by beta cells. Most people affected with diabetes are diagnosed with type 2. In addition, type 2 diabetes is generally found in people with the co-morbidities of hypertension and hyperlipidemia who are overweight [11]. This group of co-morbidities is called metabolic syndrome. The blood glucose trends for individuals who have type 2 diabetes and do not administer insulin doses are smoother and have less drastic peaks and valleys than those with Insulin-dependent diabetes mellitus (IDDM). Currently, CGMs are not as popular for individuals with type 2 diabetes, so there is not as much available data, but this is changing slowly. Ideally, after developing an algorithm to generate models for individuals with Insulin-dependent diabetes mellitus (IDDM), the algorithm could be applied to individuals with type 2 diabetes. Developing models for individuals with type 2 diabetes is becoming increasingly important as the global number of diagnosed patients with type 2 diabetes has rapidly increased [12].

Most machine learning implementations related to diabetes are broadly focused and generalize how machine learning can be used for diabetes control. Brown [13] describes 42 factors affecting blood glucose control, ranging from carbohydrate quantity to activity intensity

to puberty. The factors are associated with many controllable and non-controllable daily events impacting diabetes management. The factors influencing diabetes control are not simply measured variables since stress and illness are not predictable, measurable, or consistent in their effects. The factors are not only inconsistent, but they are inconsistent from person to person. The irregularities contribute to why developing general models for glucose control is difficult or even impossible. Every person's physiological signals related to metabolic health are unique and variable. Thus, developing a universal diabetes prediction model requires many considerations.

6.4.2 OhioT1DM Dataset

The OhioT1DM dataset [1] is a dataset that Ohio University generated for an AI/machine learning challenge in 2018. The dataset was later made available for research purposes, and the researchers added additional data with new participants in 2020. The OhioT1DM Dataset [1] provides data collected from a wearable sports watch, and the study participants were responsible for recording data themselves. To avoid human error, the data in the OhioT1DM Dataset samples only contain physiological signals from the Basis Sports Band (Intel, Santa Clara, California) in the 2018 study and the Empatica Sports Band (Empatica, Boston, Massachusetts). The dataset has data from 12 participants, six from the first study in 2018 and 6 more from the second study in 2020. Table 1 contains a subset of all the physiological signals and health metrics in the dataset considered for model development in this project. The table organization is by feature, description, measurement, sampling rate, and which study year included the measurement. In the dataset, bolus insulin is administered in a single dose, while the basal insulin rate is given incrementally over an extended period. Galvanic skin response measures the conductivity of the skin, and Sanchez-Comas et al. [14] describe this as a way to measure sweat levels. Some of the available features in the database not listed below were not considered for the thesis because they were self-reported.

Table 1. Physiological signals and health metrics used from the OhioT1DM [1] dataset for model development. The feature label is the identification string used in the dataset files. Columns highlighted in green were included in the dataset identified by the release year in the column heading, while columns highlighted were not included in the dataset.

Description	Feature Label	Measurement	Sample Rate	2018	2020
Blood glucose Measurement	glucose_level	milligrams/ deciliter	5 Minutes	Yes	Yes
Incremental insulin doses	Basal	Units/ hour	N/A	Yes	Yes
Modifications to the pre-set basal schedule	temp_basal	Units/ hour	N/A	Yes	Yes
Optical heart rate measurement	basis_heart_rate	Heartbeats/ minute	5 Minutes	Yes	No
Instantaneous insulin dose	Bolus	Units	N/A	Yes	Yes
Galvanic skin response [14]	basis_gsr	MicroSiemens	2018: 5 minutes 2020: 1 minute	Yes	Yes
Measurement of skin temperature	basis_skin_temperature	Fahrenheit	2018: 5 minutes 2020: 1 minute	Yes	Yes
Measurement of air temperature	basis_air_temperature	Fahrenheit	5 minutes	Yes	No
Total step count in the given period	basis_steps	Count	5 minutes	Yes	No
Magnitude of acceleration	Acceleration	Gravity	1 minute	No	Yes

6.4.3 Machine Learning and Diabetes

Research by Zhu et al. [15] looked at all papers published that contain research on diabetes using deep learning. Zhu et al. [15] categorized all the research into three significant groups: prediction of diabetes diagnosis, glucose management, and diagnosis of diabetes-related complications. This thesis falls into the glucose management category, as the goal of this project focuses on glucose values and trends. Some research that falls into this category, like Welsh et al. [16], focused on the blood glucose controls problem where insulin is the input variable and

blood glucose measurements are the outputs [7, 16]. The review by Zhu et al. concisely brings together key research findings related to deep learning.

Finan et al. [17] reviewed many focused machine learning studies involving activity and diabetes. The studies are primarily direct cause-and-effect focused. The participants exercised for either a one-time or recurring period of time while wearing a CGM. The studies reviewed by Finan et al. [17] focused on using data collected from trial closed-loop systems that used blood glucose as the measurement to control insulin delivery and blood labs to get accurate data related to the quality of control and risk of developing secondary conditions. Finan et al. focused on long-term diabetes control and chronic health, while this thesis output focuses on acute management. [17]

Sowah et al. [18] looked at methods to make managing diabetes easier using machine learning. Their research approach used algorithms to macro-control diabetes. The AI algorithm used blood glucose values, geolocation, image food classification, insulin delivery values, and many other health factors. The scope of the study by Sowah et al. [18] was vast and required a significant reliance on participants actively recording constant, detailed health information. Schwartz et al. [19] attempted a similar study to this thesis project, but the study reported that it did not achieve valuable results when analyzing participants individually. Schwartz et al. [16] reported a primary limitation related to the useability of the data obtained from the wearable fitness trackers they tried to leverage. The thesis project uses the OhioT1DM [1] dataset that contains pre-verified data from wearable technology to avoid the issues identified by Schwartz et al. [19].

Doorn et al. [20] researched how activity impacts blood glucose predictions in a machine-learning model for individuals with normal glucose metabolism, prediabetes, and diagnosed type 2 diabetes. First, the impact of activity was determined through activity monitoring, measured using an accelerometer and blood glucose values from a CGM. Next, the model was trained with data from individuals with type 2 diabetes then the prediction models were re-implemented using individuals with Insulin-dependent diabetes mellitus (IDDM) in the OhioT1DM [1] dataset. The research concluded that predictions made using only blood glucose were the only predictions considered accurate enough to predict glucose values safely [20].

The research published by Li et al. [21, 22] compared prediction accuracy using the blood glucose prediction models GluNet and Convolutional Recurrent Neural Networks (CRNN). The models in the study compared the prediction horizon (PH) accuracy at 30 and 60 minutes, measured using RMSE, on the UVA Padova simulator and simulated human data [23] and data from the OhioTD1M [1] dataset. The studies mentioned other health metrics included with the datasets but do not indicate if the models considered these metrics. Furthermore, the research in these publications focused on how the GluNet and CRNN models perform on different datasets compared to other published models. This thesis focuses on how a model performs with a variable data structure. Thus, model performance and metrics are not directly comparable. However, these publications offer valuable insights about model decision-making processes and data analysis, particularly statements related to comparing RMSE results between models and studies.

Table 2. Results from Martinsson et al. [24] trained on the OhioT1DM dataset. The table displays the results per participant in the 2018 study group labeled by participant ID (PUID) and future prediction times (output times) of 0.5 and 1.0 hours.

	1.0 Hour Input Length			
	0.5 Hour Output Length		1.0 Hours Output Length	
PUID	Mean RMSE	Standard Deviation	Mean RMSE	Standard Deviation
559	18.77	0.18	33.70	0.37
563	17.96	0.19	29.01	0.17
570	15.96	0.37	28.47	0.82
575	21.68	0.22	33.82	0.27
588	18.54	0.11	31.34	0.21
591	20.29	0.11	32.08	0.18

Martinsson et al. [24] focused on generating a simple deep-learning RNN-LSTM cell-based model with the 2018 OhioT1DM [1] dataset that anyone could run offline on a laptop or mobile device. The study shares the code for their project on GitHub for others to reproduce their results. The models trained by Martinsson et al. [24] used an input time series array of blood glucose values (input length) and predicted single blood glucose values 0.5 and 1.0 hours into the

future (output length). The model consisted of an LSTM cell with 256 units, a dense layer with 512 units and ReLU activation, a 20% dropout layer, a dense layer with 256 units and ReLU activation, and a 30% dropout layer into a dense layer with 1 unit. The model then duplicated this structure so that one of the single output layers had a linear activation function and the other had an exponential function. In addition, the study calculated the mean RMSE value (μ) and standard deviation(σ), which are used in the negative log-likelihood (NLL) loss function. The results were found from training the model on 60% of the participants' data, validating each epoch on 20% of their data, and predicting the blood glucose values individually on the last 20% of each participant's data. The study scaled the blood glucose values by a factor of 0.01, used a batch size of 1024, and ran with 10,000 max epochs with an early exit condition of 200 epochs without the loss decreasing past the minimum. The Adam [10] optimization algorithm optimized the model with a learning rate of 1e-3.

A unique component of the Martinsson et al. [24] study was the implementation of different loss functions in model training and optimization that optimized the mean and standard deviation. However, the unique physiological-based loss function did not improve the study's results. One key finding of the Martinsson et al. [24] study was that a 1-hour array of blood glucose values produced the lowest RMSE value for a 0.5-hour output length compared to 0.5, 2, and 3-hour input arrays. They also found that 256 LSTM units were best for an input array of 1 hour and an output time of 0.5 hours. Table 2 shows the relevant results from the study, identified by participant number (PUID) from the Ohio T1DM Dataset. The results in Table 2 suggest that, while the model performs similarly for most participants, the RMSE is notably higher with data from participant 575.

Mirshekarian et al. [25] provided a theoretical, detailed review of the use of a deep learning RNN LSTM layer applied to blood glucose predictions [25]. The study worked with the OhioT1DM [1] dataset as well as the AIDA [26] and UVA Padova [23] human simulators. The prediction groups in this study considered agnostic and internal scenarios. The agnostic scenario trained the model on all samples, including samples defined as "what-if events." The research group used the agnostic scenario to test a model's ability to "implicitly estimate life events that are likely to happen in the prediction range" [25]. The internal scenario consisted of models that

were trained on only the data that do not contain “life events” in the input or output data range. Any missing sequences in the data was linearly interpolated for a period of time that is less than the output length. Periods of missing data that were greater than the output length were ignored. The authors do not explicitly describe why they chose the length of time input into the model; however, they do incorporate more data features. The data features from the OhioT1DM study that were considered were blood glucose, insulin, meals, skin conduction (GSR), heart rate, and time of day.

Mirshekarian et al. [25] scaled the blood glucose values by a factor of 1/600, and all other data features were normalized to a range from 0 to 1/3. The model input contained six hours of data to make a single value prediction at 0.5 and 1.0 hours. The optimizer used in the model was the RMSProp [27] optimizer because they reported that the RMSprop optimizer generated better results than the Adam [10] optimizer. The amount of data in the OhioT1DM [1] dataset is significantly less than the data from the simulators. Therefore, the authors pre-trained the model with a learning rate of 0.01, which decreased to 0.001 if the validation metrics did not improve for five epochs. Finally, the authors used the OhioT1DM data to finish the model training with an optimizer training rate that decreased from 0.001 to 0.0001 when performance did not improve for five epochs.

Although the study by Mirshekarian et al. [25] is not reproducible without having access to the simulator data, the results can be used to qualitatively compare how various features impact model predictions and comparative metrics for feature analysis. The study by Mirshekarian et al. [25] also does not justify why the input length of 6 hours was chosen, which could be a limitation of the results, shown in Table 3. The results in Table 3 depict the models with the lowest RMSE and included insulin, meals, skin conductance, and heart rate as features.

Overall, results from Mirshekarian et al. [25] suggested that including features improves prediction accuracy, but the authors do not break down the results by participant or provide analyses of how data from different participants could impact the results. However, this study provides excellent background analysis and context buildup regarding how the data was processed and analyzed with the what-if analysis, agnostic, and internal scenarios.

Table 3. RMSE results generated by Mishekarian et al. [22] using the OhioT1DM data set. © 2019 IEEE, Reprinted with permission from S. Mirshekarian, H. Shen, R. Bunescuand C. Marling, "LSTMs and Neural Attention Models for Blood Glucose Prediction: Comparative Experiments on Real and Synthetic Data", 2019.

TABLE IV

AVERAGE AND STANDARD DEVIATION OF RMSE RESULTS OVER 20 RUNS ON THE OHIOT1DM DATASET, FOR LSTM WITH AND WITHOUT VARIATIONAL DROPOUT

		Agnostic		Inertial	
Model	Features	30 min	60 min	30 min	60 min
t_0	BG	22.60	36.66	21.67	34.43
ARIMA	BG	20.17	33.47	19.36	31.45
LSTM (d = 0.0)	BG	$19.51_{0.17}$	$32.04_{0.28}$	$18.64_{0.10}$	$29.64_{0.23}$
LSTM (d = 0.1)	BG	$19.07_{0.12}$	$31.11_{0.16}$	$18.72_{0.13}$	$29.52_{0.15}$
LSTM+Mem (d = 0.1)	BG	$19.09_{0.11}$	$31.09_{0.16}$	$18.75_{0.16}$	$29.55_{0.15}$
LSTM (d = 0.0)	BG, I, M	$19.01_{0.19}$	$30.94_{0.82}$	$18.35_{0.24}$	$29.42_{0.54}$
LSTM (d = 0.1)	BG, I, M	$\mathbf{18.74}_{0.17}$	$\mathbf{30.63}_{0.27}$	$\mathbf{18.07}_{0.10}$	$28.32_{0.21}$
LSTM+Mem (d = 0.1)	BG, I, M	$18.77_{0.17}$	$30.65_{0.27}$	$18.09_{0.10}$	$\mathbf{28.28}_{0.28}$

TABLE V

AVERAGE AND STANDARD DEVIATION OF RMSE RESULTS OVER 20 RUNS ON THE OHIOT1DM DATASET, FOR LSTM WITH 10% VARIATIONAL DROPOUT WHEN DIFFERENT FEATURES ARE USED

							Agnostic		Inertial	
BG	I	M	SC	HR	ST	ToD	30 min	60 min	30 min	60 min
●	○	○	○	○	○	○	$19.07_{0.12}$	$31.11_{0.16}$	$18.72_{0.13}$	$29.52_{0.15}$
●	○	○	○	○	○	●	$19.11_{0.09}$	$31.11_{0.18}$	$18.75_{0.18}$	$29.32_{0.30}$
●	●	●	○	○	○	○	$18.74_{0.17}$	$30.63_{0.27}$	$18.07_{0.10}$	$28.32_{0.21}$
●	●	●	○	○	○	●	$18.80_{0.16}$	$30.56_{0.23}$	$18.13_{0.13}$	$28.26_{0.22}$
●	●	●	●	○	○	○	$18.81_{0.21}$	$30.31_{0.19}$	$18.19_{0.11}$	$28.29_{0.25}$
●	●	●	●	○	○	●	$18.81_{0.16}$	$30.18_{0.20}$	$18.10_{0.09}$	$\mathbf{28.19}_{0.15}$
●	●	●	●	●	○	○	$18.77_{0.16}$	$30.28_{0.19}$	$18.10_{0.14}$	$28.30_{0.32}$
●	●	●	●	●	○	●	$\mathbf{18.70}_{0.13}$	$\mathbf{30.17}_{0.22}$	$18.07_{0.08}$	$28.20_{0.19}$
●	●	●	●	●	●	○	$18.76_{0.17}$	$30.40_{0.23}$	$18.10_{0.08}$	$28.37_{0.29}$
●	●	●	●	●	●	●	$18.83_{0.17}$	$30.43_{0.23}$	$\mathbf{17.99}_{0.10}$	$28.20_{0.19}$

6.4.4 Gaps in the Literature

Building connections between diabetes, related health metrics, and machine learning research is quickly becoming a popular area to study. This project uniquely aimed to take the progress and experiments described in the related studies and expand on the scope of the projects in many ways. First, the individuality of a participant's data is a significant factor in the analysis and interpretation of results. The research in this study focused on results per individual, like in Martinsson et al. [24], to understand how well a model would perform explicitly using an individual's physiological data values. The project also utilized various combinations of time and passively collected physiological signals. Martinsson et al. [24] and Mirshekarian et al. [25] used experimental pre-testing to examine the impact of individual variable LSTM units, input time, and feature combinations. This study aimed to expand on the combinations of time and data features reported in Martinsson et al. [24] and Mirshekarian et al. [25] studies. Finally,

understanding the relationship between input time and data signals is essential in predicting blood glucose values. Martinsson et al. [24] only focused on blood glucose predictions with 0.5 to 3 hours of input time, while Mirshekarian et al. [25] adjusted the raw data values with a 6-hour input time. Not all actions cause immediate reactions in blood glucose values, such as exercise; thus, it is crucial to consider the information before the last time-step before prediction. The research in this study brought together findings from the studies like Martinsson et al. [24] and Mirshekarian et al. [25] to report on how a more extensive scope of features interact to impact blood glucose predictions. Although there are infinite potential combinations and adjustments to the feature sets and the deep learning models, this thesis also tried to generate questions to inspire future projects related to blood glucose predictions.

6.5 PROJECT OBJECTIVES

This thesis project aimed to understand how various health metrics and machine learning model features affect glucose prediction accuracy. The specific design criteria used to develop the project scope and objectives focused on assessing the impact of model parameters that would allow the models to function in a real-time, autonomous environment. The criterion insinuates that no human interaction is required to obtain, filter, or validate the data or outputs involved. The design criterion is aggressive but provides some context for future autonomous decision-making. A similar condition, in which the model should be capable of running on a personal laptop or smartphone, is found in the Martinsson et al. [24] study. The specific objectives are described in detail below.

Objective 1: To Determine How the Length of the Time Series Input into the Model (Input Length) and the Length of Time The Model Predicts into the Future Prediction (Output Length) Impact the Prediction Accuracy of a Machine-Learning Model

Factors that impact blood glucose measurements can lead to effects over a broad range of time intervals. This thesis objective focused on (a) characterizing how the lengths of time the model uses for inputs (input length) impact model accuracy and (b) how accurately the model can predict for various periods in the future (output length).

Objective 2: To Understand How Physiological Signals and Health Metrics Affect the Prediction Accuracy of a Machine-Learning Model

This thesis objective focused on how passively collected health metrics and physiological signals from wearable health sensors can be utilized in a deep-learning model to improve the accuracy of future blood glucose predictions. The specific physiological signals and health metrics considered included:

1. Galvanic skin response (GSR)
2. Movement (from accelerometer measurements)
3. Blood Glucose (from CGM measurements)
4. Heart Rate
5. Step Count
6. Skin Temperature
7. Insulin dose
    a. Basal
    b. Bolus
8. Calories burned per minute (calculated)
9. Cumulative Calorie count (calculated)

Objective 3: To Understand the Value and the Limitations of Error Metrics Calculated from a Series of Predicted Points and Error Metrics Calculated From a Single Predicted Point

Machine learning studies related to blood glucose prediction in the literature report error metrics (RMSE) calculated from a single prediction value of blood glucose at 0.5- and 1-hour output lengths. However, producing only a single blood glucose value provides minimal context regarding how or why the model determined the predicted point. This project, instead, predicted a time series of blood glucose values from the end of the input length (prediction time) through the entire output length. This objective aimed to understand the advantages, disadvantages, and limitations of error metrics calculated from a time series of points for a specified output length (single-value RMSE) and error metrics calculated at the output length time (series RMSE).

7 METHODS

7.1 PRELIMINARY STUDY

7.1.1 Overview

The initial work for this project occurred during a semester-long project as part of the Rochester Institute of Technology EEEE Bio-robotics class. The project required real-time machine learning predictions using raw physiological data. The project analyzed two individuals' Apple Health data. The project aimed to integrate physiological signals and metrics with Apple Health data to generate a machine-learning model that predicted nighttime blood glucose levels. Data obtained from two participants contained two years of data with a large variety of health metrics, including blood glucose measurements. The initial objective was to predict categorical high, normal, or low blood glucose values with different machine-learning architectures representing the nighttime blood glucose average.

7.1.2 Preliminary Study Methods

Process Flow

Figure 3 shows the high-level process flow involved in the preliminary work. The steps involved in the process flow and associated files included pulling data from a database source (RAW data), forming the signals and metrics into features (Data Filtering/Input+Output Definitions), processing the data in the model (Model->Blood Glucose Predictions), and analyzing the results (Result Analytics). The second step, working with the data, was where most of this work's objectives were focused.

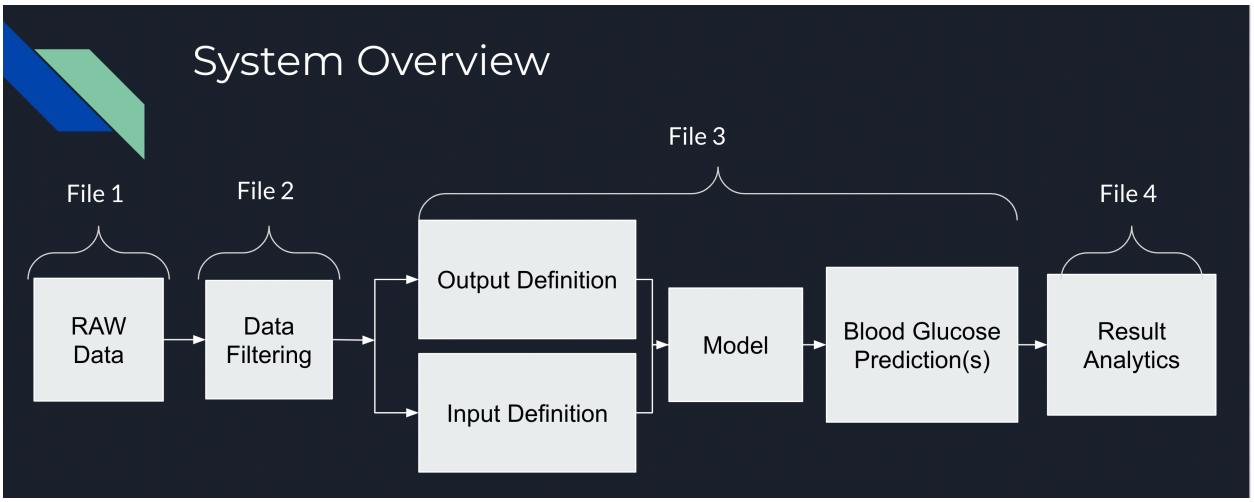


Figure 3. System overview from the preliminary project showing the major data pipeline checkpoints grouped by their Python script files.

Datasets

The datasets used in the preliminary study consisted of Apple Health data provided by three subjects with diabetes obtained using a continuous glucose monitor connected to their iPhone and an Apple Watch for over a year. While the data contained significant information, only a tiny fraction of the features were related to diabetes or contained consistent data.

Python

The models for the preliminary study were implemented in Python. Python [28] is a high-level coding language that has become increasingly popular, especially in science, technology, engineering, and math (STEM) applications. The Python community is very active, and together they have developed some of the best open-source code libraries with many tutorials and resources. For example, the pipelines that processed the Apple Health exported dataset in the preliminary project and the mathematical models were all developed using Python libraries. Libraries, called modules or packages, are groups of code containing many powerful functions that other projects quickly implement. This project's most extensively used libraries were TensorFlow for model development and Pandas for data configuration.

Pandas [29]

Pandas is a powerful library with many tools for working with data. Many functions in Pandas simplify methods to pull, manipulate, and store data using data frames. The Pandas library was crucial when working with large datasets like the OhioT1DM dataset or Apple Health Exports because of the ease of transforming a large data set. In addition, Panda's library has many math and statistical pre-defined functions, which made much of the analysis in this project more manageable.

TensorFlow [30]

TensorFlow is an open-source library for developing end-to-end machine learning projects [30]. TensorFlow contains many popular development tools for machine learning and extensive support documentation. In addition, the library is modular for research development and allows for creation of special analysis tools, such as those created to perform the root-mean-square-error calculations, which TensorFlow does not include.

7.1.3 Preliminary Study Results and Conclusions

The goal was to identify if a machine-learning model could be used to predict trends related to nighttime blood glucose values. All the models developed could predict some nighttime trends; however, the prediction capabilities heavily depended on how the nighttime and trend thresholds were defined. Figure 4 contains nine overnight predictions of normalized glucose levels for a 3-hour output prediction length, calculated in 5-minute time steps. The blue line represents model predictions, while the orange line represents the actual values. The general agreement between actual and predicted glucose values suggested that the models could predict generalized overnight trends. While the predictions were not perfect matches and did not help classify overnight highs and lows, the results indicated that generalizing a very volatile physiological metric more than 3 hours into the future was plausible. The pipeline structure and software tools provided the base platform for the thesis work.

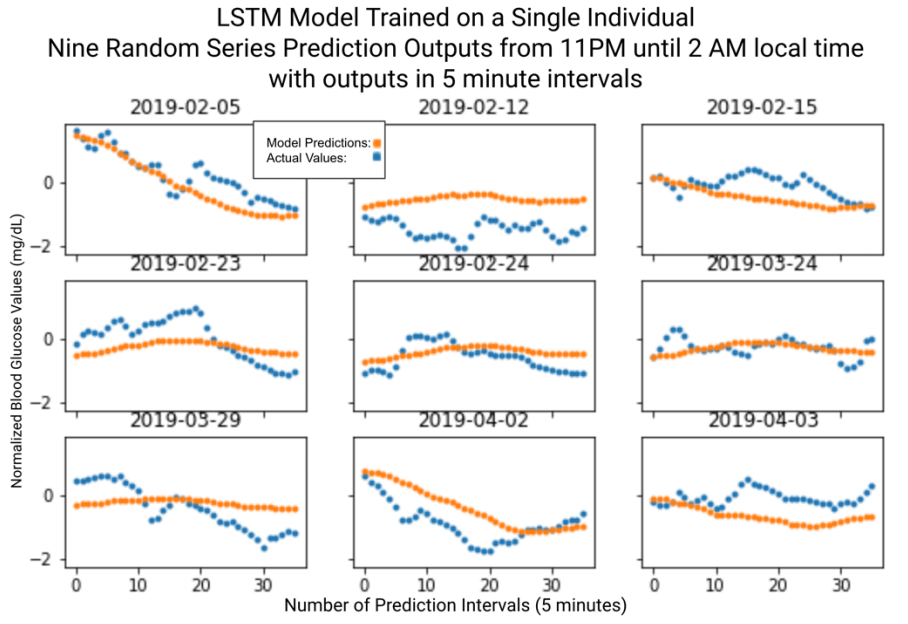


Figure 4. Example of a debugging plot from the preliminary study. The agreement between theoretical and predicted glucose values for randomly selected predictions demonstrated the feasibility of using health data to predict nighttime blood glucose trends.

7.2 PROJECT SCOPE

The thesis aimed to determine how combinations of data features (physiological signals and health metrics) and time lengths (input and output lengths) impact the prediction accuracy of blood glucose values. The general approach included defining and implementing a data pipeline, validating the data processing methods, and modifying variables based on the project objectives. A Long Short-Term Memory neural network, trained and tested on the OhioT1DM dataset, generated the predictions. The project considered two high-level variable groups: the data features (physiological signals and health metrics) and time lengths. The time lengths consist of input length [of time] and output length [of time]. The specific variables considered were data features in Table 4, input lengths of 1.0, 3.0, 6.0, 7.0, and 8.0 hours, and output lengths of 0.5 and 1.0 hours. The prediction models output either a single value or a series of blood glucose values that were compared using RMSE. The single value RMSE only considered the value at a specified output length (Single Value RMSE), e.g., 0.5 or 1.0 hours, and was the metric used to compare to the literature. The prediction model also produces blood glucose outputs as a series, so a series RMSE was also calculated across the full output length. The measured points

corresponding to the predicted points are intrinsically related, so the predicted points are also assumed to be related. Both final value and series RMSE calculations are included in the results as each has different interpretations.

The description of the pipeline configuration and results are organized below based on the three project objectives. Objective 1 focused on changing input lengths with blood glucose as the only input feature. The best-performing time lengths from Objective 1 continued to Objective 2 to narrow the project's scope. Objective 2 focused on the impact of individual features or feature combinations on prediction accuracy. Finally, Objective 3 examined how series and single-value outputs compared quantitatively and qualitatively.

7.2.1 Definition of Features

An overarching goal of this project was to determine how different data features impact the prediction of blood glucose measurements. Martinsson et al. [24] and Mirshekarian et al. [25] analyzed the effects of input time and individual features, respectively. The results from these studies were used to generate the scope of features that this project investigated. Table 4 provides an overview of the features Martinsson et al. [24], Mirshekarian et al. [25], and this project investigated. While there are not many new features considered, the novelty of this research is identifying how the variables interact since the number of variable combinations significantly increases the time needed to train models.

Table 4. Comparison of model configurations.

Configurations	Mirshekarian [25]	Martinsson [24]	This Project
Features: Health Metrics and Physiological Signals	Blood Glucose Insulin Carbohydrates Galvanic Skin Response Skin Temperature Heart Rate Movement (accelerometer) Time of Day Agnostic + Internal scenarios	Blood Glucose	Blood Glucose Galvanic Skin Response Heart Rate Movement(accelerometer) Skin Temperature Metabolic metrics: - Calories - Cumulative Calories Time of Day
Time Combinations: Input Length: Output Length:	6 Hours 0.5, 1 Hour	0.5, 1, 2, 3 Hours 0.5, 1 Hours	1, 3, 6, 7, 8 Hours 0.5, 1 Hours
Training Set Size Ohio Split Ratios:	Simulation Sets and OhioT1DM 2018 60%-20%-20%	All participants in OhioT1DM 2018 60%-20%-20%	All participants in OhioT1DM 2018 60%-20%-20%
Principal Component Analysis (PCA)	No	No	Yes

Definition of Inputs

The inputs passed to the model were combinations of specific variables defined based on the project objectives. The input variables determined the structure and data shape that the model used to generate a prediction. The inputs considered in this project are in the Features and Time Combinations rows in Table 4. The input features and time define the two-dimensional shape of the input data. The time dimension consists of the input length broken into consistent timesteps. Each timestep contains a unique measurement of the input features at that time.

One of the crucial design criteria for this project was using passively collected signals and metrics to eliminate human error. Therefore, some of the metrics that the OhioT1DM [1] dataset provides and Mirshekarian et al. [25] implemented, such as meal carbohydrate counts, stressors, and work intensity, were not considered in this project. Table 1 provides a more detailed list of features considered in this project: insulin (basal and bolus), all the raw Basis

sports band physiologic signals and metrics, and the relative time of day/week. The only calculated feature represented the calories burned per unit of time and the cumulative sum of these calories/day.

Definition of Outputs

The input structure and output structure do not need to complement each other. For example, the research published by both Martinsson et al. [24] and Mirshekarian et al. [25] used a series of consistent steps to define inputs but only single-point output shapes. This project's outputs predicedt the single feature of blood glucose at a series of steps consistent with the input step rate and the output length determine the number of steps. While the series output structure is more computationally expensive, it provides more contextual information that can help understand how the model generates predictions. Also, a single-series prediction can generate multiple output predictions. For example, the output lengths for this project are 0.5 and 1.0 hours, so a series prediction for 1.0 hours also contains a 0.5-hour prediction.

7.3 PROJECT WORKFLOW AND IMPLEMENTATION—DATA PIPELINE

7.3.1 Pipeline Summary

The data pipeline includes all processing, from loading data to generating the results tables. Each step in the pipeline saves outputs as a file or files so the following step can run independently. Pre-defined configuration files determine what variables are used to train the model.

Since there are infinite experimental combinations of features, defining the scope of the features to be included in each trial was an important step. The features included were defined in a custom-created configuration file that contained a run identification value, model parameters, and data parameters. The configurations were used to help organize the feature combinations so that any debugging that occurred was equivalent to all trials.

The configuration file defined variables that determined the shape of the input and output data for a particular run. An example configuration file, Figure 24, is shown in the appendix. The data's input shape consisted of features sampled every 5 minutes for the extent of the input

length. The data's output shape consisted of blood glucose predictions every 5 minutes from 5 minutes after the input length through the output length. Other variables defined in the configuration file are model hyperparameters that adjust how the model is organized and fits the data. The hyperparameters were tuned to improve the models' overall training speed and testing accuracy.

The general pipeline structure described in the preliminary project was also used for the thesis project. While the high-level structure is the same, most of the specifics associated with each portion of the pipeline were adjusted to match the objectives of the thesis project. The data pipeline is described in detail in subsequent subsections, as indicated, and includes the following operations as shown in Figure 3:

1. Raw Data Processing (7.3.2)
2. Data Filtering (7.3.3)
3. Defining Inputs and Outputs (7.3.4)
4. Defining Prediction and Accuracy Metrics (7.3.5)
5. Defining Model Architecture (7.3.6)

The feature testing scope was defined based on the studies by Martinsson et al. [24] and Mirshekarian et al. [25], with the features considered listed in Table 4 and described in Table 1. The data pipeline was validated by reproducing methods and results found in the literature to ensure consistency with published works.

7.3.2 Raw Data Processing

Missing Data and Sensor Errors

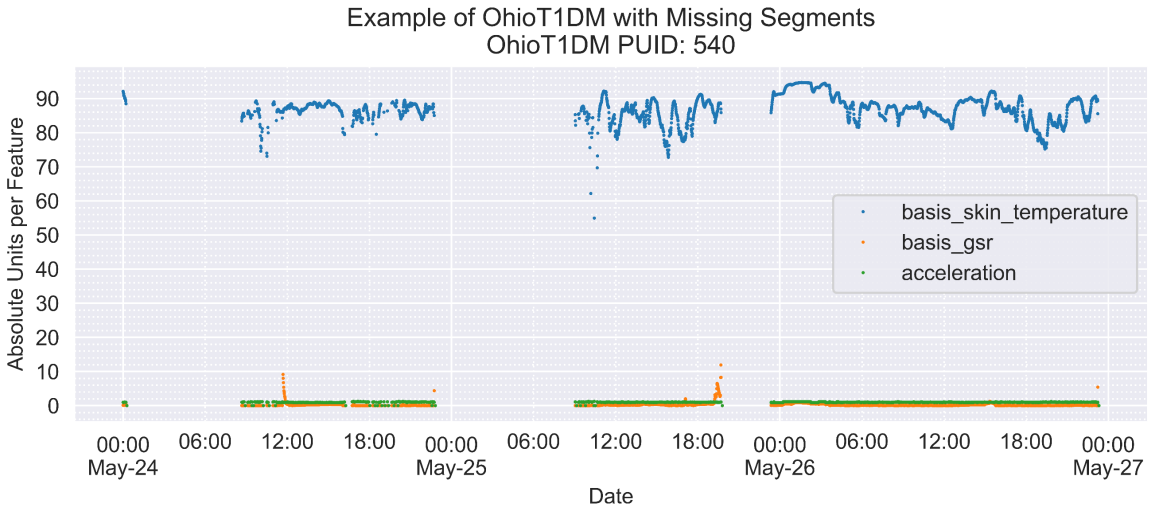


Figure 5. Health metric data from OhioT1DM [1] participant PUID: 540 with extensive periods of missing data.

Data can come in all shapes and sizes, but an inconsistently shaped dataset cannot pass into the model. For example, the OhioT1DM [1] dataset provides raw physiological signals pulled directly from consumer health-tracking wearables. Unfortunately, it is not uncommon for some lower-quality sensors to report values inaccurately or not report values at all. For example, in Figure 5, OhioT1DM [1] participant PUID: 540 has significant periods with missing data. The gaps in data could indicate that the participant likely charges their sports band every night, which could bias the data toward daytime values and decreases the total useable data. The model cannot process data containing missing values, and missing periods cannot be easily synthesized or replaced. Similarly, Figure 6 shows participant PUID: 544's accelerometer data that contains a period where the participant was experiencing an unlikely 16 Gs of acceleration. All sensor metrics included in this project were manually checked for values outside expected ranges, and extreme values were removed. The only accelerometer data that was considered extreme and removed is shown in Figure 6.

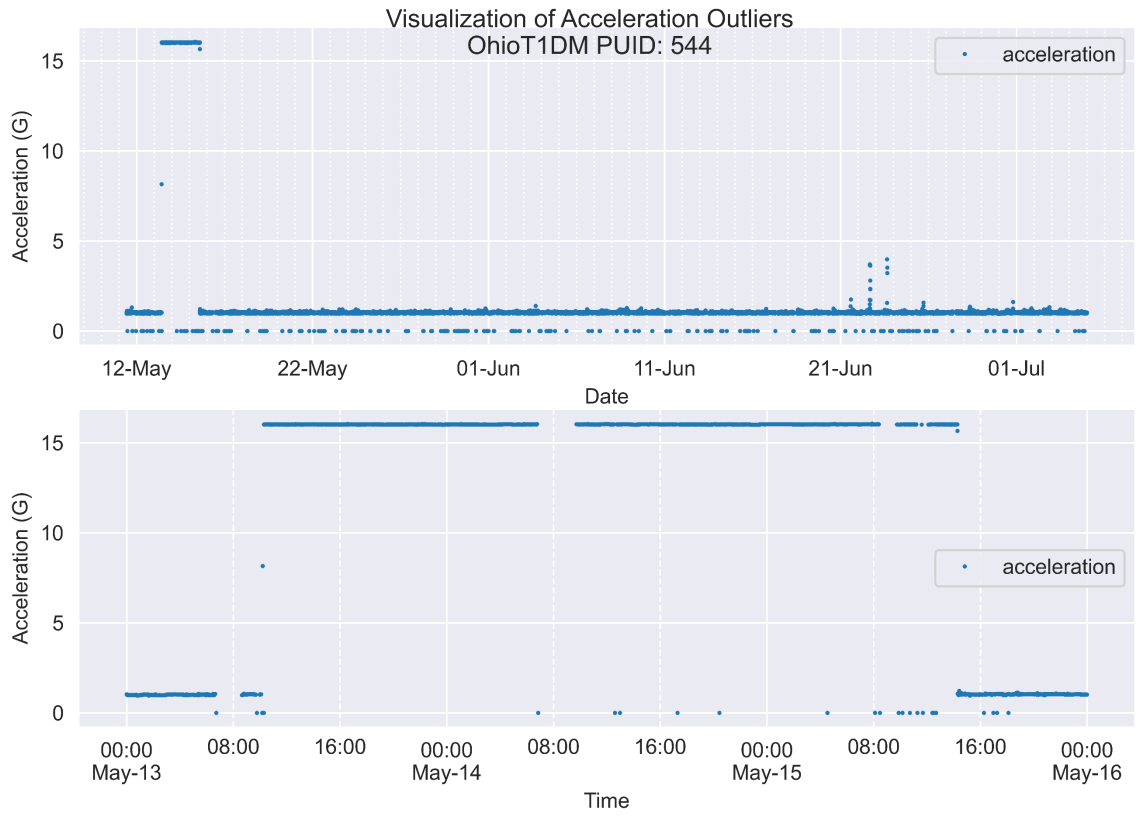


Figure 6. One month of raw accelerometer data from the OhioT1DM [1] participant PUID: 544. The top plot shows the wide-range view of accelerometer data used to identify obvious outliers, while the bottom plot is a close-up view showing likely erroneous accelerometer data showing measurements of approximately 15 G-forces.

Maximizing the amount of usable data is crucial, especially in a small study with limited continuous sequences. For example, the participant data shown in Figure 5 is consistently missing nighttime values which would limit the model's exposure to nighttime data. Smaller gaps, however, have some options for augmenting missing data, such as interpolating missing points linearly between known values.

Blood glucose measurement gaps of less than 0.5 hours were interpolated because this sequence length was less than the minimum prediction length considered. Blood glucose was the only measurement with interpolated values because blood glucose was included in every prediction configuration analyzed. Missing points from the other sensors were missing for all the sensors on the wearable sports band, and not all those sensor readings could be linearly interpolated. The process filled in 1,223 total values, increasing the usable data length by 943

points, approximately 3 hours of continuous data. Any gaps in the blood glucose series larger than 0.5 hours were removed.

Calculating Additional Metrics: Calories Burned

The only metric used in the preliminary project not included in the OhioT1DM data set was the number of calories burned per unit of time. Therefore, when transitioning to the OhioT1DM dataset, a calculation was used to synthesize the number of calories burned to avoid significant changes to the legacy code structure. The calculations to determine the number of calories burned per minute for an adult male and female are

$$Men: \frac{Calories}{minute} = \frac{(-55.0969 + 0.6309 * HR + 0.1988 * weight + 0.2017 * age)}{4.184} \qquad 2$$

$$Women: \frac{Calories}{minute} = \frac{(-20.4022 + 0.4472 * HR - 0.1263 * weight + 0.074 * age)}{4.184} \qquad 3$$

where HR is the subject's heart rate [31]. Although more accurate, an equation applying VO2 required more information about the person, making calculation impossible from the information provided in the dataset. Also, the OhioT1DM dataset does not provide specific weights and ages for the participants, only their average age ranges. Therefore, the calculation uses the median age value for each participant and the weight value associated with that age from Fryar et al. [32]. The calories/minute is then used to generate a second feature, cumulative calories, representing the total calories burned throughout the day. Due to the potential for increased error, models trained with calculated caloric metrics are identified separately.

Insulin as a feature

Insulin is the most direct factor impacting blood glucose and was essential to the prediction calculations. However, insulin is separately listed because it is not a physiological signal or health metric but a medication dosage. The participant's insulin pump was responsible for collecting the insulin delivery data, but it commonly does not capture the actual insulin delivery, such as in cases where participants administered insulin shots or the pump site failed. Therefore, every dataset and model configuration trial considered two variations: with and without insulin, as denoted in the configuration title (e.g., Insulin True or Insulin False).

7.3.3 Data Filtering

Re-sampling

An LSTM model design takes data in time steps, and each time step must represent a consistent rate. Furthermore, not all signals and metrics are reported at the same time interval or the same sampling rate. Thus, the first pipeline filtering process was re-sampling, aggregating data to report at a consistent designated rate. Each metric had a different rate, magnitude, and behavior, so it was crucial to consider each feature when defining a function for resampling. For example, the OhioT1DM [1] data set reports all data signals and metrics for the 2018 study in 5-minute increments, but the times reported do not coincide. Interval shifting can be done similarly to down-sampling or decreasing the frequency of measurements. When downsampling most metrics in the dataset, predefined functions in Pandas like 'mean' can represent a group of data as a single value, but some features, like step count, carbohydrates, and insulin bolus, need to be summed cumulatively. The predefined 'sum' function was used for these measurements because their values must represent the total within a time step. The pipeline in this project re-sampled all features to a common rate of 5 minutes because most data were reported at 5-minute intervals, so the re-sampling effectively shifted the values to the nearest interval.

Dataset Grouping and Splitting

Before more data processing could occur, the dataset was split into three datasets subsets: training, validation, and testing. The splitting needed to happen before more data processing occurred because further processing, such as normalizing or shaping the inputs, applies dataset-specific modifications, like a linear shift based on the mean feature value. Furthermore, the assumption during testing is that the model is ignorant of the testing dataset; therefore, only the training set determined the data-specific modifications.

Martinsson et al. [24] and Mirshekarian et al. [25] implemented an OhioT1DM [1] dataset split with a 60-20-20 train-validate-test ratio. Therefore, the same split as the literature was applied in this project for consistency in processing for comparisons between studies. Additionally, alternate splits that decreased validation and testing sets did not contain more than a single day of data and would not provide sufficient insight.

Dataset Normalization

Normalization was done to prevent biasing of the model on features with large magnitudes. The feature magnitudes in a dataset can vary significantly, and without scaling, they would bias the model. Normalization is a process that scales data of varying magnitudes to a consistent range. Data normalization shifts the bounds of distribution to a consistent range, commonly from -1 to +1 or ±0 to +1. The Martinsson et al. [24] and Mirshekarian et al. [25] studies differed in their normalization processes. Martinsson et al. [24] scaled the blood glucose values by 1/100, which scaled the blood glucose values with a natural range of 50 to 450 to 0.05 to 4.5. Mirshekarian et al. [25] treated the features and the blood glucose differently. First, the features were all normalized from 0 to 1/3. Next, blood glucose was scaled at 1/600, which shifted the values from 1/12 to 2/3 (8/12).

This study did not consider the Martinsson et al. [24] scaling because applying PCA requires that all data features have a max value below 1. However, this project did consider the Mirshekarian et al. [25] scaling and found that it performed better than standard normalization from 0 to 1, so this scaling method was adopted. However, this method biases the model toward blood glucose values since they have a larger magnitude. Therefore, future work should consider alternate scaling factors.

Principal Component Analysis (PCA)

PCA is a feature transformation method that reduces the feature space by determining the principal components or vectors that best describe the variance in the system. PCA was included for comparison purposes because PCA decreases the feature space without significantly decreasing total information [33]. Decreasing the space reduces computational load, increasing the model's training speed and allowing for more feature combination testing. The first component describes the most variance in the system, followed by the second, which is orthogonal to the first and describes the most variance left after the first. Then, the data features linearly transform onto the new axis. Since the first few components best describe the data, the feature space can be reduced by leaving out the last few axes that represent that variance the least. This process works better with a more significant number of features as the amount of data described by the first few components will represent a more significant percentage of the data.

PCA is a tool defined by a dataset before it reduces its space. Therefore, the processing pipeline must apply PCA after splitting the data into train/validation/test groups so that the training group defines the PCA fit without information about the other groups and then applies it to the validation and test groups. Furthermore, a feature set transformed by PCA may perform differently than the original feature set, so labels identify PCA configurations separately.

The time to train model configurations can vary significantly, increasing processing time as the number of features increases. The time that a model takes to train and test is an essential real-world use case consideration. Choosing between two models with similar results could come down to training time. For instance, if trials show that applying PCA produces the same results as trials without PCA, the model may still run the train/test segment 75% of the time, which, when models run for many iterations, can significantly impact run time.

Data Windowing

Following the grouping and normalization processes, the data was two-dimensional, organized by features in a continuous series of time steps. The prediction model was designed to process input lengths of data and predict output lengths. Windowing was done to break the large time series data into specific input and output lengths. Each window was a specified length of continuous time containing both the input and the output lengths. The windowing process did not cut the data but instead created continuous overlapping lengths of data. The shift between windows can be a single step or more, but this project used a single step between windows to maximize the amount of data the model could process.

7.3.4 Defining Inputs and Outputs

The next step in the pipeline required shaping the features into inputs and outputs. Shaping was the final processing step before the data could be passed into the model. The windows were separated into inputs and outputs for final shaping and processing. While it was outside this project's scope, future research should analyze the impact of alternate input and output step rates for prediction accuracy, computational load, and training time.

The different time lengths and feature combinations impact the number of windows generated. Increasing the input length gives the model more physiological context but fewer total

data windows to process. The input lengths and features did not require additional shaping or processing before the model could analyze them.

The outputs, on the other hand, required additional processing prior to being passed to the model. The output predictions only consisted of one feature, blood glucose, so other features were removed. Also, the output was shaped to either be a single-value prediction or a series prediction. All series predictions used 5-minute time steps.

7.3.5 Defining Prediction and Accuracy Metrics

Root mean square error (RMSE) was used to quantify error since it is the standard metric reported in the literature to compare blood glucose predictions. RMSE compares the differences in variance between two series of points. Lower RMSE values in this project suggested that the prediction values aligned with the actual measured values better than higher RMSE values. The Keras RMSE metric reported from the evaluate function in TensorFlow calculates RMSE for each prediction in a batch and then combines each batch RMSE value into a single value. The RMSE calculation for output lengths of 0.5 and 1.0 hours reported in the literature represent the accuracy at that given timestep. For a series output, RMSE can be calculated at any point in the series or on any segment length within the series. Since this project produces series outputs, all results include single-value RMSE calculations for output lengths of 0.5 and 1.0 hours and series RMSE values calculated using all values in the output length.

7.3.6 Defining Model Architecture

Hyperparameters and Custom Model

Hyperparameter tuning is an essential step in developing deep-learning models. However, while tuning was essential in this project, optimizing the models for the best possible performance was not. The objectives for this project required predictions to be consistent in performance so that they could be analyzed and compared.

Batch Size

The last processing step was grouping the inputs and outputs into batches. The hyperparameter batch size defines the number of windows passed to the model for each training step. The model processes one batch per training step, and after each batch, the model adjusts the

internal weights associated with each layer to fit the data better. An epoch counts the number of times all batches have been processed. Smaller batch sizes produce more batches and, therefore, more adjustments per epoch than larger batch sizes. After all the batches have been processed, the model's training status is evaluated. Therefore, the smaller batch sizes are more likely to overfit and process longer per epoch than larger batch sizes, but the smaller batch sizes will require fewer total epochs to fit the model. Batch size tuning is vital for large datasets because batch size drastically impacts training time and memory optimization. For example, Martinsson et al. [24] model's batch size was 1024, while Mirshekarian et al. [25] reported their batch size as 512. This project used a batch size of 512 to maintain consistency with the literature.

LSTM Units

The LSTM model contains an LSTM layer comprised of a variable number of LSTM units or cells. The preliminary study found that varying these units did not significantly affect RMSE values but impacted training time. Moreover, the research published by Martinssonet al. [24] corroborated this and identified that any units above 256 produced the same results. Therefore, all configurations reported in this project used 256 units.

Loss Functions

Loss measures how far the model outputs deviate from the expected outputs. The loss function calculates the loss and is a critical hyperparameter because it controls how the optimizer fits the model. The preliminary study, based on series outputs, used mean squared error (MSE), but Martinsson et al. [24], based on single value outputs, found that 'negative log-likelihood loss (NLL) performed the best.

The difference between the loss functions was not easily compared because the methods used by Martinsson et al. [24] were not designed for a series output. Thus, the NLL was re-designed to work with a series output. Initial tests with NLL and MSE often produced a flat prediction line that over-generalized the outputs. To counteract the overgeneralization, correlation loss was factored into the loss function.

A small multi-run experiment comparing various loss functions and combinations determined the loss function used for this project. The loss functions included MSE, NLL, and

correlation. NLL consists of the reworked series and the original final value loss functions. The results in Figure 7 include the top-performing loss function configurations for single-value RMSE with output lengths of 0.5 and 1.0 hours and series RMSE with an output length of 1.0 hours. The MSE, NLL, and Correlation combined loss function performed most consistently; therefore, it was chosen for the rest of the project.

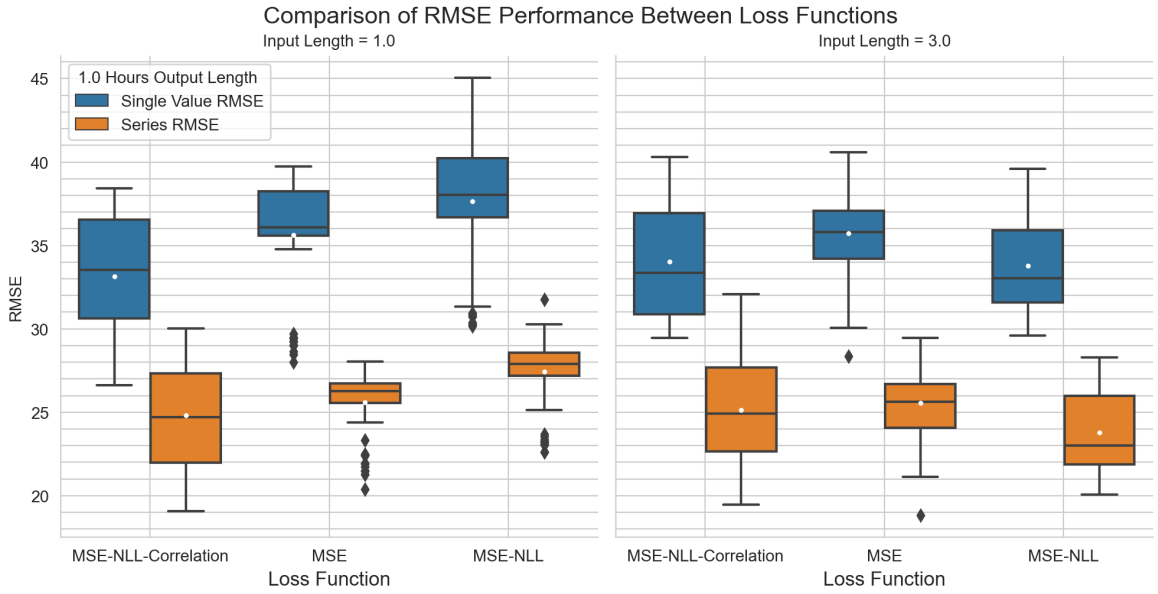


Figure 7. Comparison of series and single-value RMSE values between the custom loss functions Mean Squared Error (MSE) + Negative Log Likelihood (NLL) + correlation function, MSE alone, and MSE + NLL. The prediction model configurations used input lengths of 1.0 and 3.0 hours, an output length of 1.0 hours, and all raw OhioT1DM signals and insulin. The white points indicate the mean RMSE value.

Optimizer

The optimizers considered for use were Adam and RMSprop. Martinsson et al. [24] implemented Adam, while Mirshekarian et al. [25] reported that RMSprop slightly outperformed Adam. In this study, the Adam optimizer consistently reported significantly lower RMSE measurements than RMSprop in experiments with a learning rate of 0.001.

Model Summary

The project approached the model design differently than other studies in the literature. The project used a customized model-tuning process focused on series outputs instead of a single value for each final output time. Also, the model tuning process was not focused on optimizing

the results but on setting parameters for fast and consistent results for many time and feature combinations.

The resulting customized model consisted of an LSTM cell with 256 units; a ReLU-activated dense cell with 512 units, a 20% dropout layer; a ReLU-activated dense layer with 256 units and a 10% dropout layer, and a final dense layer with as many cells as there were timesteps to the output length.

The maximum number of training epochs was 500 for all runs. That many epochs were rarely necessary because the exit criterion that analyzed the validation loss exited early to avoid overfitting. The exit criteria occurred when there was no improvement in the validation loss metric in 10 epochs.

7.4 PROJECT VALIDATION

An initial set of validation experiments was done to reproduce the results reported by Martinsson et al. [24] using the OhioT1DM dataset and their published configurations. Validation consisted of two experiments to ensure the data pipeline for single and series outputs compared to the literature. The first experiment was done to ensure that results generated with the configuration reported by Martinsson et al. [24] were comparable to their published results. The second experiment compared the final-value RMSE results for a single-value output with a series output with the same configuration used for the first experiment.

The comparisons were made to ensure that the pipeline developed for this project produced results like those obtained using related literature methods. The Martinsson et al. [24] results were the best to reproduce because the code was shared on GitHub and only used blood glucose values to make predictions. The code from this project produced similar results when passing in the parameters from Martinsson et al. [24] and justified that the processing leading up to the model design and predictions was consistent with related studies. Direct comparisons of RMSE values could not be made due to uncertainties associated with the filtering steps in the literature; nevertheless, comparing trends associated with the RMSE results was used to ensure consistency with other published studies. Reproducing the Mirshekarian et al. [25] study could not be done

without simulation data, but the results were still valuable for benchmarking feature combinations.

7.5 PROJECT VARIATION

Once the validation experiments were shown to reproduce published results, the validation results were used as the starting point for subsequent experiments. Each objective required experiments that focused and pivoted around a single adjustable pipeline or model configuration component. The experiments that required changes to multiple variables were divided into trial groups. Each trial included a single variable modification from the previous trial configuration and was run multiple times to determine repeatability. Each experiment and trial aimed to determine the impact of a particular configuration on the trained model's reported accuracy metrics.

Objective 1 focused on the effects of the varying input length. Varying input length added the least complexity to the prediction model because it only modulated the input length and blood glucose was included as the only feature. Previously, Martinsson et al. [24] considered input lengths of 0.5, 1.0, 2.0, and 3.0 hours because the RMSE values increased with the increased length. Alternatively, Mirshekarian et al. [25] only reported results using 6.0 input hours. Longer input lengths gave the model more historical context but increased computational time and reduced the total amount of prediction data. Therefore, this objective aimed to determine how the input lengths larger than 3 hours impacted the reported prediction accuracy. The input lengths reported include 1.0, 3.0, 6.0, 7.0, and 8.0 hours. The row in Table 5 labeled Obj. 1 shows the input lengths and configurations associated with Objective 1.

Objective 2 focused on varying the combinations of input features derived from the OhioT1DM dataset. Input features add significantly more complexity to the pipeline because unlimited features and combinations exist. Mirshekarian et al. [25] reported on multiple feature combinations using the raw data from the OhioT1DM dataset. This project aimed to expand the combination of features with a particular focus on fitness metrics requiring no human intervention. For example, data features passively collected without human interaction: health metrics, physiological signals, and medication dosing. The experiments associated with

Objective 2 compared the impact of select features on the dataset's raw data, calculated metrics, processed metrics, and medication dosing. The specific features used are shown in Table 5, with the two input lengths with the lowest RMSE from the input length experiments.

Objectives 1 and 2 focused on quantifying the impact of modifying the input dimensions, while Objective 3 focused on the output and value-associated error metrics. An output dimension consisting of only a single blood glucose value provides minimal context regarding how or why a model determines a predicted point. This project structured the output dimension into a time series of blood glucose values from the end of the input length (prediction time) through the entire output length. Objective 3 aimed to identify the advantages, disadvantages, and limitations of the error metrics calculated from a time series of points for a specified output length (single-value RMSE) and error metrics calculated at the output length time (single-value RMSE). The row labeled Obj 3 lists the configurations associated with Objective 3.

Table 5. Configuration parameters used for the validation and objective experimental trials. The Total Runs column combines all the configuration parameters considered in the experiment and multiplies it by the number of times each trial was repeated, resulting in the total number of runs required for the experiment.

Test	**Input Length (hours)**	**Features**	**Insulin**	**PCA**	**Total Runs**
Val.	1, 2, 3 Total: 3	Blood Glucose 0.5 Hours Single-Value Blood Glucose 1.0 Hours Single-Value Blood Glucose Series Total: 3	No Total:1	No Total:1	9*100 runs Total: 900
Obj. 1	1, 3, 6, 7, 8 Total: 6	Blood Glucose Total: 1	No Total:1	No Total:1	6*50 runs Total: 300
Obj. 2	6, 8 Total: 2	Blood Glucose *[control]* Blood Glucose + Time BG + Raw OhioT1DM Signals + Time BG + Cal + CumCal + HeartRate +Time BG + Calories/Minute (Cal) + Time BG + Cal + CumCal + Time BG + Cumulative Calories + Time BG + CumCal + Heart Rate + Time BG + Heart Rate + Time BG + All Metabolic + Time Total: 10	Yes & No Total:2	Yes & No Total:2	60*25 runs Total: 1500
Obj. 3	6, 8 Total: 2	BG *[control]* BG + Raw + Time BG + Metabolic + Time Total: 2	Yes & No Total:2	Yes Total:1	6*10 runs Total 60
				Total	2,760 Runs
			Run time:	~4 min	11,040 Min
					184 Hours
		Blood Glucose = BG Cumulative Calories = CumCal Galvanic Skin Response = GSR Raw = Heart Rate + GSR + Skin Temp.			~ 8 Days

8 RESULTS

The outline below describes the organization of the following results sections and the associated experiments. In addition, Table 6 contains a list of the nomenclature used to describe the configuration associated with each experiment.

The results in 8.1, Validation Experiments, were used to validate the data pipeline developed for this project. The results include the following:

- Comparison of results obtained using the Martinsson et al. configuration parameters
- Comparison of single-value RMSE results from models trained for single-value and series outputs
- Comparison of single-value and series RMSE results.

The results in 8.2, Input Length Analysis (Objective 1), include the following:

- Comparison of results obtained using input lengths of 1.0, 3.0, 6.0, 7.0, and 8.0 hours

The results in 8.3, Feature Combinations (Objective 2), include the following:

- Comparison of results from models trained with blood glucose from 8.2 (baseline) versus all OhioT1DM raw features
- Comparison of results from models trained with OhioT1DM raw features versus calculated metabolic features
- Comparison of results obtained using individual and grouped metabolic features
- Comparison of results from models trained with all OhioT1DM raw features or all metabolic features, with and without principal component analysis (PCA)
- Comparison of results from models trained with all OhioT1DM raw features, with and without insulin
- Comparison of results from models trained with all metabolic features, with and without insulin

The results in 8.4, Series vs. Single Value RMSE (Objective 3), include the following:

- Example of Series Prediction versus Single-Point Prediction

- Comparison of predictions with mean single-value RMSE greater than 90% and less than 90% of the test set predictions
- Comparison of predictions with mean series RMSE greater than 90% and less than 90% of the test set predictions
- Comparison of the predictions using the features blood glucose only or raw OhioT1DM signals that had mean series RMSE greater than 90% of the test set predictions

Table 6. Nomenclature used for names and notations found in results and appendix.

Terminology	**Description**
Martinsson	Results as described in the Martinsson et al. [24] study
Validation	Model designed in the likeness of the Martinsson et al. [24] study
Single-Value RMSE	RMSE calculated at the indicated output length for all predictions
Series RMSE	RMSE calculated on all points in output length for all predictions
ONLY_BG	Only feature in the dataset was the blood glucose (BG)
PCA Raw	PCA was applied to the raw OhioT1DM signals
PCA Raw Insulin	PCA was applied to the raw OhioT1DM signals with insulin included
ABE	Active Burned Energy (ABE) was calculated as a cumulative sum of all energy burned throughout the day starting at midnight
CAL	Feature combination included calorie count (CAL), BG, and a time metric
HR	Feature combination included Heart Rate (HR), BG, and a time metric
CAL_ABE	Feature combination included ABE, CAL, BG, and time metric
ALL	Feature combination included CAL, ABE, HR, BG, and a time metric
InsulinTrue	Feature combination included insulin
InsulinFalse	Feature combination did not include insulin

8.1 VALIDATION EXPERIMENTS

Results from the first validation experiment demonstrated that the pipeline developed in this project was consistent with the literature. The experiment used the configuration from Martinsson et al. [24] that generated their lowest reported RMSE. The pipeline was considered validated if the results aggregated from this experiment trended with the published results.

Each trial in the experiment trained a prediction model on all participant data. The trained prediction models were tested individually on each participant testing set with aggregated results in the row labeled Validation in Table 6. The row labeled Martinsson corresponds with the published RMSE results. The published Martinsson et al. [24] configuration used an input length of 1.0 hours, the feature blood glucose, and reported a single-value RMSE at 0.5 and 1.0 hours. In Table 7, the results are separated for each output length. The *mean* is the average RMSE value across the trials of the experiment, and the *std* is the standard deviation. The 'All' column reports all participants' cumulative average RMSE and standard deviation. Percent difference quantifies the difference between the published and experimental mean RMSE results.

Every mean RMSE value reported from the first experiment, except one, had a larger magnitude than the published results. The maximum percent difference between RMSE in this experiment and the published results for all participants was less than 10%. Although this project reported a larger average RMSE, it should be noted that the windowing in Martinsson et al. [24] was not identical to this study; thus, model performance based on the RMSE values could not be directly compared. However, the magnitudes of the mean RMSE values for each individual participant trended identically. For example, the participant with the lowest RMSE value in both studies was PUID 570, and the highest RMSE value was associated with PUID 575. Overall, the results demonstrate that this project's data pipeline produced predictions with similar RMSE trends to those published in the literature.

Table 7. Comparison of single-value RMSE mean, standard deviation, and percent difference between the results published by Martinsson et al. [24] and this project's validation experiment that trained models using the same configuration reported by Martinsson et al. [24]. The top and bottom subtables correspond to output lengths of 0.5 and 1.0 hours of blood glucose predictions. The models generated by both configurations predict a single blood glucose value at either 0.5 or 1.0 hours. The cell denoted in yellow indicates the lowest percent difference.

0.5-Hours Output								
OhioT1DM PUIDs		559	563	570	575	588	591	All
Martinsson	mean	18.77	17.96	15.96	21.67	18.54	20.29	18.87
	std	0.18	0.19	0.37	0.22	0.11	0.11	1.81
Validation	mean	20.41	17.50	17.35	22.97	19.38	20.83	19.74
	std	0.27	0.21	0.73	0.23	1.37	0.15	2.07
Percent Difference	%	8.72	-2.56	8.72	6.02	4.55	2.64	6.63

1.0-Hours Output								
OhioT1DM PUIDs		559	563	570	575	588	591	All
Martinsson	mean	33.69	29.01	28.45	33.82	31.34	32.08	31.40
	std	0.37	0.17	0.82	0.27	0.21	0.18	2.12
Validation	mean	34.75	29.05	30.02	36.44	32.28	32.93	32.58
	std	0.45	0.28	1.56	0.32	3.00	0.27	2.91
Percent Difference	%	3.10	0.11	5.44	7.74	3.01	2.63	3.73

The second validation experiment used the same number of input hours and configuration as the first experiment except for the output dimension, which configured the model output to generate a series of predictions using 5-minute timesteps. The series prediction model is more computationally complex than the single-value prediction model because the series output requires more parameters to optimize for multiple output values instead of a single output value. Therefore, this experiment aimed to ensure that the single-value RMSEs obtained at 0.5 and 1.0 hours from both the single-value and the series prediction models were comparable.

The single-value RMSEs are aggregated in Table 8 for the 0.5 and 1.0-hour outputs, respectively. While the results from the first experiment required a model for each output length (0.5 and 1 hour), the series configuration captures the 0.5-hour prediction value in the 1.0-hour output length prediction.

The series predictions generally had larger single-value RMSEs indicated by the negative percent differences. A notable exception, highlighted in green, was associated with the 1.0-hour output length predictions for participant PUID 559, where the series prediction model reported a 4.16% lower average RMSE than the single-value prediction model. Conversely, the series prediction model, 0.5-hour output results for participant PUID 559, reported an 8.95% larger average RMSE than the single-value prediction model, which is insignificant compared to the other participants. The percent difference between the 0.5-hour output average RMSE is 7% larger than the difference between the 1.0-hour prediction RMSE.

When trained using the same configuration as the single-value models, the series prediction model reporting a larger RMSE was expected because of the additional complexity requiring more fine-tuning during training. However, the lower percent difference between the 1.0-hour output prediction RMSE and the 0.5-hour prediction was unexpected. The low percent differences between RMSE values for the 1.0-hour output predictions suggest that the series predictions produce comparable results to the single-value prediction models.

Table 8. Comparison of single-value RMSE mean, standard deviation, and percent difference between this project's validation experiment that trained models using the same configuration reported by Martinsson et al. [24] and models trained using the same configuration but with a series output. The top and bottom subtables correspond to the output lengths of 0.5 and 1.0 hours of blood glucose predictions. The validation prediction models output a single prediction point at 0.5 or 1.0 hours, and the series prediction models output a series of prediction points through the output length of 1.0 hours. The series prediction models calculate the single-value RMSE at the 0.5 and 1.0 hours prediction points. Cells in green identify the only instance that the series prediction model reports a lower single-value RMSE than the validation model.

0.5-Hours Output								
OhioT1DM PUIDs		559	563	570	575	588	591	All
Series Prediction Model	Mean	22.41	19.65	17.8	26.07	20.67	23.77	21.73
	Std	0.97	1.06	2.34	0.70	1.10	0.94	3.01
Validation	Mean	20.41	17.50	17.35	22.97	19.38	20.83	19.74
	Std	0.27	0.21	0.73	0.23	1.37	0.15	2.07
Percent Difference	%	-8.95	-10.94	-2.55	-11.87	-6.22	-12.38	-9.15

1.0-Hours Output								
OhioT1DM PUIDs		559	563	570	575	588	591	All
Series Prediction Model	mean	33.36	29.05	30.55	38.76	34.63	33.85	33.36
	std	1.01	0.88	2.93	0.63	1.57	0.51	3.44
Validation	mean	34.75	29.05	30.02	36.44	32.28	32.93	32.58
	std	0.45	0.28	1.56	0.32	3.00	0.27	2.91
Percent Difference	%	4.16	0	-1.73	-5.98	-6.78	-2.71	-2.36

8.2 Input Length Analysis (Objective 1)

The time feature analysis compared series prediction model RMSE values for input lengths of 1.0, 3.0, 6.0, 7.0, and 8.0 hours with blood glucose as the only other included feature. The time analysis from Martinsson et al. [24] assumed that after 3 hours, the prediction RMSE would continue to increase following the positive trend indicated by input hours of 1.0, 2.0, and 3.0 hours. However, the methods reported by Mirshekarian et al. [25] used a 6.0-hours input length. Therefore, this objective aimed to expand on input length experiments reported by Martinsson et al. [24] to include longer input lengths.

The results from this experiment, shown in Table 9, depict the same mean RMSE increase reported by Martinsson et al. [24] between 1.0 and 3.0 hours. However, this increasing RMSE trend did not continue as the input lengths increased but instead decreased. Furthermore, configurations with longer input lengths reported smaller mean RMSE values than the 1.0-hour input length. However, the larger input lengths have significantly fewer available prediction windows than the 1.0-hour input length. The window count, included in Table 9, represents the number of predictions used to calculate RMSE; therefore, the number of available prediction windows decreases as the input length increases. The decrease in the number of predictions could explain the decrease in standard deviation as the input length increases.

Table 9. Single-value and series RMSE results for various input lengths output length of 1.0 hour. The RMSE calculations for the output lengths of 0.5 and 1.0 are separated into subtables. The test split window count is the maximum number of predictions for input lengths in the testing data split. The models in this experiment used the maximum number of prediction windows for the corresponding input length. Cells in yellow denote the input length with the lowest mean single-value RMSE result for each output length.

0.5-Hours Output						
Input Hours		1	3	6	7	8
0.5 Hour single-value RMSE	mean	21.73	22.65	20.48	20.33	20.29
	std	3.01	3.97	2.67	2.65	2.76
1.0-Hours Output						
Input Hours		1	3	6	7	8
1.0 Hour single-value RMSE	mean	33.36	33.92	32.25	31.70	31.43
	std	3.44	4.03	3.17	3.25	3.29
1.0 Hours Series RMSE	mean	23.34	24.06	22.28	22.02	21.96
	std	2.66	3.77	2.50	2.51	2.58
Test Split Windows	Window Count	16,490	15,698	14,710	14,134	13,755

8.3 FEATURE COMBINATIONS (OBJECTIVE 2)

Objective 2 focused on varying the feature dimension of the input data, where the feature dimension experiments incrementally considered additional features or processes. The experiments associated with Objective 2 performed comparisons that included the following feature combinations and processes:

1. Blood Glucose against all OhioT1DM raw features
2. OhioT1DM raw features against calculated metabolic features
3. Individual calculated metabolic features
4. All OhioT1DM raw features and all metabolic features with principal component analysis (PCA)
5. All OhioT1DM raw features with and without Insulin

Each results plot includes a reference line that denotes the lowest mean RMSE value obtained from previous experiments. For example, the RMSE value calculated using an 8-hour input length was the first reference line because it was the lowest RMSE value obtained from the input length analysis. All experiments in this section used 8.0- and 6.0-hour input lengths.

8.3.1 Comparison of Results From Models Trained Using Blood Glucose Against All Raw OhioT1DM Signals

The first experiment compared the prediction errors (RMSE) obtained from models trained using blood glucose measurements (same as the time feature analysis) against models trained using all the raw signals in the OhioT1DM dataset. The simplest configuration was using blood glucose as the primary feature to predict blood glucose. Training the model with all the raw metrics from the OhioT1DM dataset was used to assess how well the model could identify the blood glucose trends when trained with unrefined raw signals. The method used unrefined raw signals as features hoping that the AI could blindly identify these features' impact on blood glucose predictions. The raw OhioT1DM features were used because these were measurements associated with participants' activity levels, which are known to impact future blood glucose trends. However, while the information associated with the raw OhioT1DM features was valuable, the raw signal measurements could be too noisy, overwhelm the optimization process, and produce inconsistently trained models limiting the applicability for real-time applications.

The comparison of RMSE values obtained from the series prediction models with blood glucose as the only feature (Glucose Baseline) and the raw OhioT1DM signals are shown in Figure 8 as box and whisker plots for input lengths of 6.0 and 8.0 hours. The orange box plot represents the single-value RMSE distributions obtained using a 1.0-hour output length, and the blue box represents the series RMSE values obtained from all the experimental trials using a 1.0-hour output length. The input lengths were compared separately because the different input lengths resulted in different window counts. The table contains the mean series and single-value RMSE values shown in the plot.

The mean RMSE value is denoted as a single white dot, and the median RMSE value is the horizontal line within the colored box. Prediction models with lower mean and median RMSE values will have less prediction error, implying a more accurate model. Furthermore, since the prediction model's training process is not deterministic, the same setup configurations and training dataset can yield different results. An ideal prediction model training configuration should have a low mean and median RMSE with a tight distribution, as indicated by the girth of the box and whiskers in a box plot. The boxplot’s bottom whisker is valuable because it denotes the models producing the lowest RMSE value from a particular configuration. While low error results could imply that the training for the prediction model found an optimal connection between the inputs and outputs, it could also have randomly predicted better on the given test set and may not accurately predict additional data.

In Figure 8, the RMSE values obtained using the raw OhioT1DM features have larger distribution ranges, shown by the more extensive whiskers, than the series baseline for both input lengths. The results obtained using the raw OhioT1DM features with an 8.0-hour input length have a significantly larger distribution range than the 6.0-hour input length. The series baseline also has lower mean series and single-value RMSE values for both input lengths and lower median values for all input lengths, except for the median value associated with the single-value RMSE and the 8.0-hour input length. Overall, the RMSE results obtained using the series baseline configurations with blood glucose as the only feature had less error than those obtained

using the raw OhioT1DM features because the series baseline results had lower mean RMSE values with a tighter distribution spread.

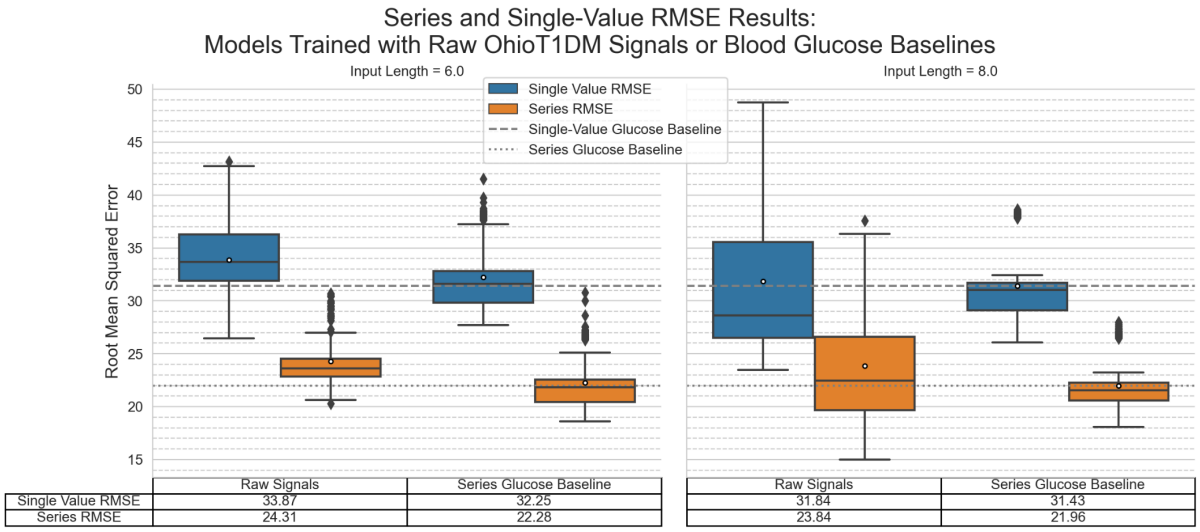


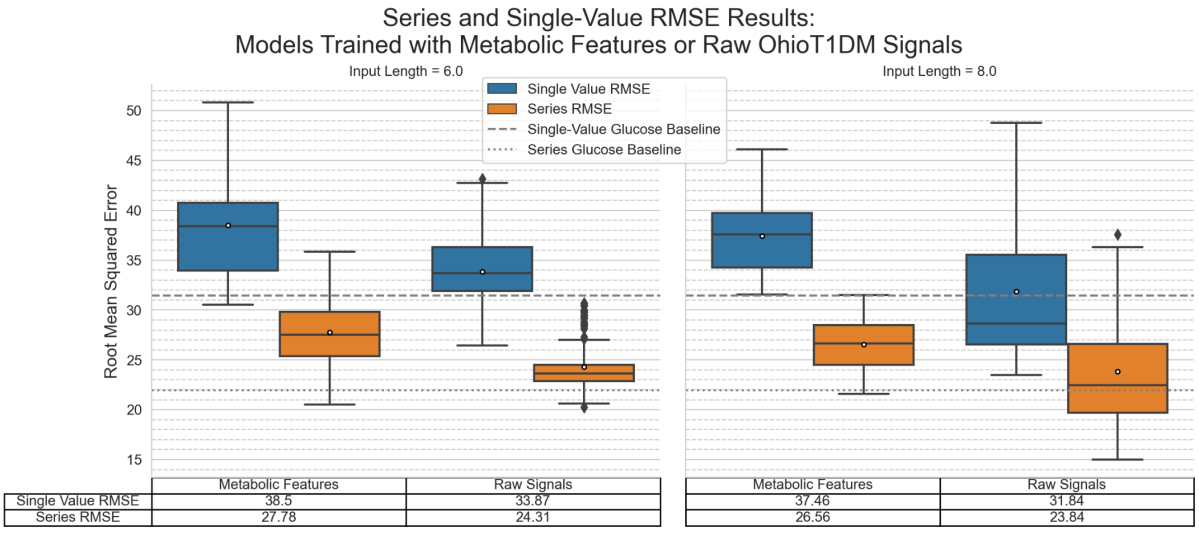
Figure 8. Comparison of single-value and series RMSE results for models trained using glucose only (Glucose Baseline) and the raw OhioT1DM signals (Raw Signals) separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results.

8.3.2 Comparison of Results From Models Trained using the OhioT1DM Raw Features against Calculated Metabolic Features

The second experiment compared the RMSE values obtained from models trained using the raw OhioT1DM signals to the results obtained using metabolic feature prediction models. The metabolic features were derived from the signals captured in the OhioT1DM dataset to try and represent the internal biological processes that directly impact the amount of glucose in the bloodstream. The metabolic features should more directly impact blood glucose than the raw signals for predicting blood glucose because they quantify the impact of physical activity on blood glucose more directly. Therefore, the expectation was that the results obtained using metabolic features would have lower mean and median RMSE values and a tighter deviation range than results obtained from the raw OhioT1DM signals. If the signals do not have lower RMSE values, that could imply that the calculations do not accurately represent the metabolic processes or that there were valuable connections between the raw signals and future blood glucose values not captured in the calculations.

Figure 9 compares the RMSE values obtained from the series prediction models with metabolic features (Metabolic) to the results obtained using the raw OhioT1DM signals (Raw) for 6.0- and 8.0-hour input lengths. The orange box plot represents the single-value RMSE distributions obtained using a 1.0-hour output length, and the blue box represents the series RMSE values obtained from all the experimental trials using a 1.0-hour output length.

The results in Figure 9 show that training with raw OhioT1DM signals produced lower mean and median RMSE values for both input lengths than training using metabolic features. In addition, training using raw OhioT1DM signals resulted in a smaller distribution range and smaller whiskers than with the metabolic features for both single-value and series outputs with an input length of 6.0 hours. A larger distribution range was observed for an input length of 8.0 hours. The results indicate that training using the metabolic features is less valuable than using the raw OhioT1DM signals for predicting future blood glucose values.



	Metabolic Features	Raw Signals
Single Value RMSE	38.5	33.87
Series RMSE	27.78	24.31

Metabolic Features	Raw Signals
37.46	31.84
26.56	23.84

Figure 9. Comparison of single-value and series RMSE values for models trained using the metabolic features (Metabolic Features) and the raw OhioT1DM signals (Raw Signals) separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results.

8.3.3 Results From Models Trained Using Individual Metabolic Features

The third experiment compared how models trained using the individual metabolic features shown in Table 5 performed when the features were included individually and in

subgroups. The previous experiment demonstrated that the prediction errors obtained with calculated metabolic features were larger than those obtained using the raw OhioT1DM signals. Therefore, this experiment aimed to identify if there was value in including any metabolic features in future experiments.

The comparisons between models trained using individual and subgrouped metabolic features are shown in Figure 10. The features were given abbreviated names to minimize the plot size. The associated nomenclature is provided in Table 6. The orange box plot represents the single-value RMSE distributions obtained using a 1.0-hour output length, and the blue box represents the series RMSE values obtained from all the experimental trials using a 1.0-hour output length. Models trained using just blood glucose (BG) for both input lengths had the lowest mean, median, series, and single-value RMSE values and the smallest distribution range (shortest whiskers). Models trained using all the metabolic features reported similar mean and median, series, and single-value RMSE values for both input lengths. Notably, training with metabolic features and an 8.0-hour input length yielded tighter deviations than those obtained with the 6.0-hour input length. The results collectively indicate that the metabolic calculations, individually or in subgroups, did not dramatically improve RMSE prediction accuracy or prediction of future blood glucose values.

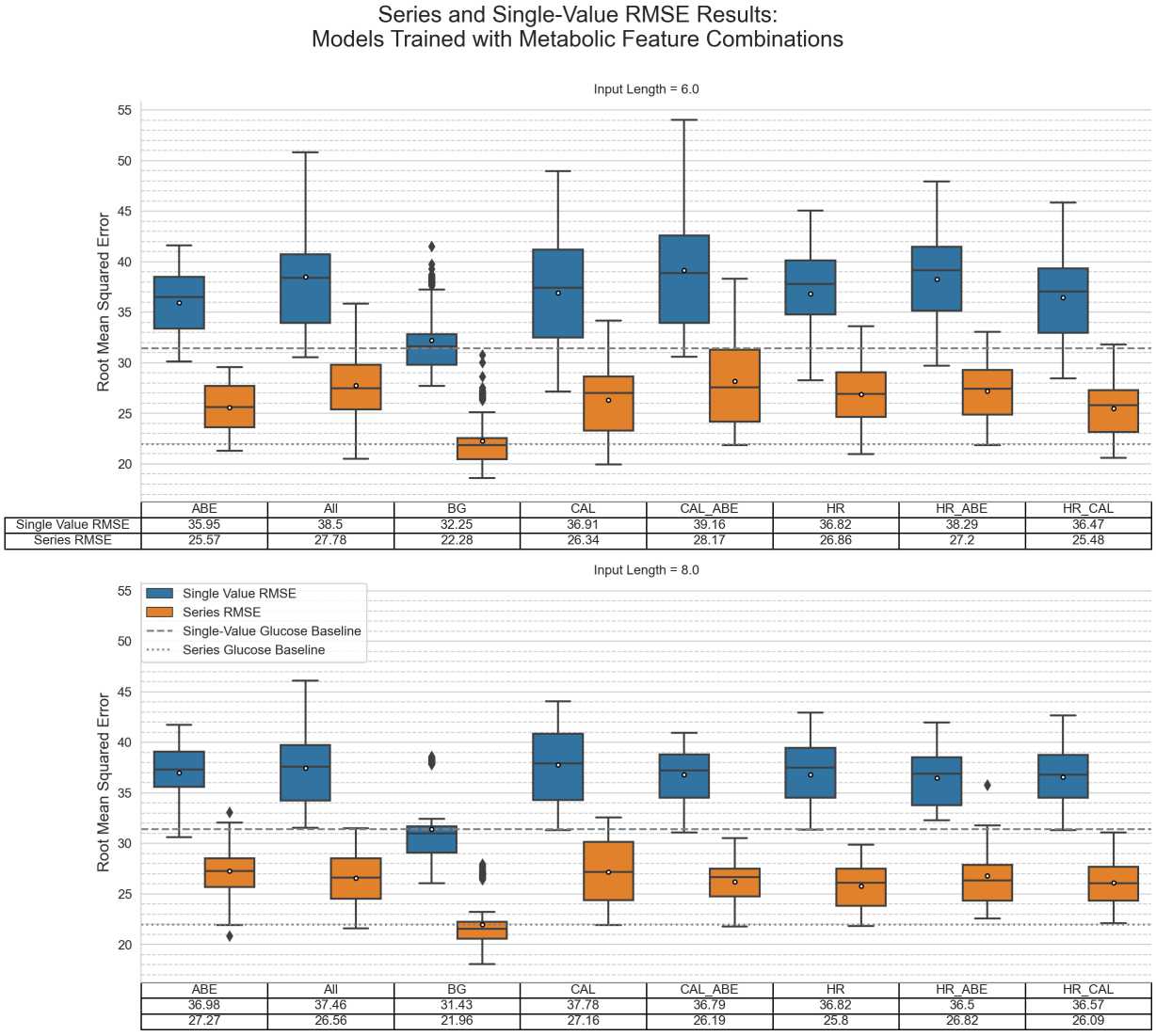


Figure 10. Comparison of single-value and series RMSE values for models trained using the metabolic feature combinations separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results. Table 6 decodes the feature abbreviations and provides definitions.

8.3.4 Comparison of Results Obtained Using all OhioT1DM Raw Features and all Metabolic Features with Principal Component Analysis (PCA)

There are infinite ways to manipulate and modify the feature space, but adding features increases computational complexity. Furthermore, while each feature intends to represent something unique, the features can be redundant. This objective examined the feature space

reduction method principal component analysis (PCA) and how it impacted the prediction error on models trained using the OhioT1DM dataset or metabolic features. Principal component analysis decreases the feature space in an enveloping method and could reduce trial volatility and eliminate redundancy in the features. This experiment compared the impact of applying PCA to models trained using the metabolic and raw OhioT1DM features for both 6.0- and 8.0-hour input lengths. Furthermore, the experiment aimed to determine if the inclusion of PCA impacted prediction results since the inclusion of PCA in future experiments could reduce processing demand and increase training speed.

The results in Figure 11 compare RMSE values obtained from training with all the metabolic features (Metabolic) to those trained using all raw OhioT1DM signals (Raw), both with and without PCA applied (PCA *, applies PCA). Figure 12 represents the results as box and whisker plots for 6.0- and 8.0-hour input lengths. The orange box plot represents the single-value RMSE distributions obtained using a 1.0-hour output length, and the blue box represents the series RMSE values obtained from all the experimental trials using a 1.0-hour output length.

As seen in Figure 11, including PCA did not impact the mean or median RMSE values for a 6.0-hour input length. However, the deviation spread for the 8.0-hour input length was lower when PCA was applied. In addition, using PCA reduced the total RMSE deviation, shown by shorter whiskers for the 6.0-hour input length, but had a larger 25% to 75% population, shown by the increased colored box size.

PCA did not impact the metabolic feature prediction models as much as the prediction models that used the raw OhioT1DM features. For example, using PCA with the 6.0-hour input length slightly decreased the mean, and median RMSE values, but no decrease in the RMSE deviation spread was observed. Conversely, for the 8.0-hour input length, PCA slightly increased the mean and median RMSE values and the spread of the RMSE deviation.

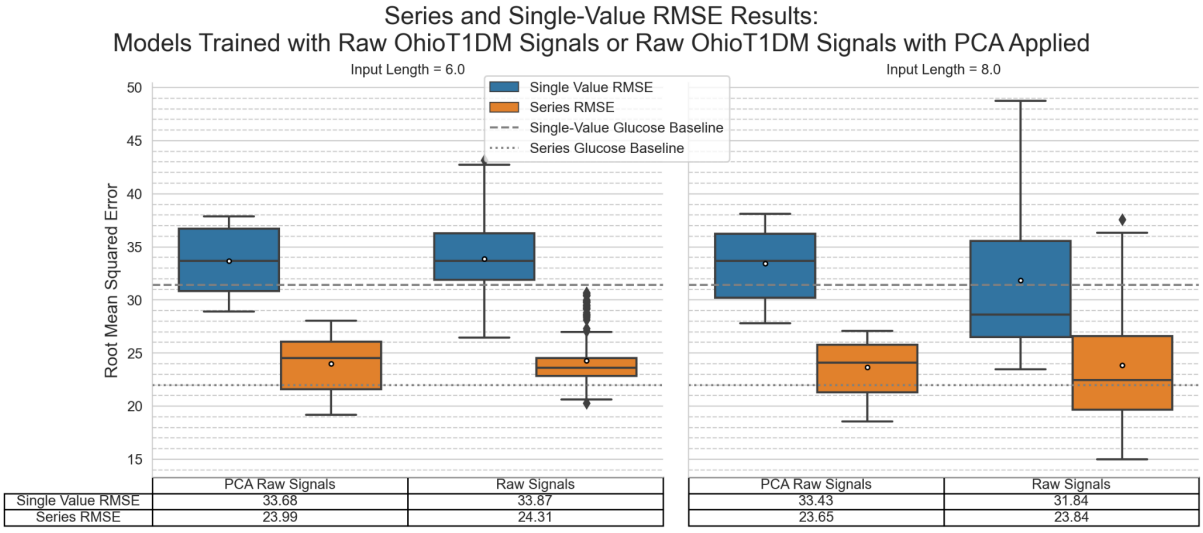


Figure 11. Comparison of single-value and series RMSE values for models trained using the raw OhioT1DM signals (Raw Signals) and the raw OhioT1DM signals with PCA applied (PCA Raw Signals) separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results.

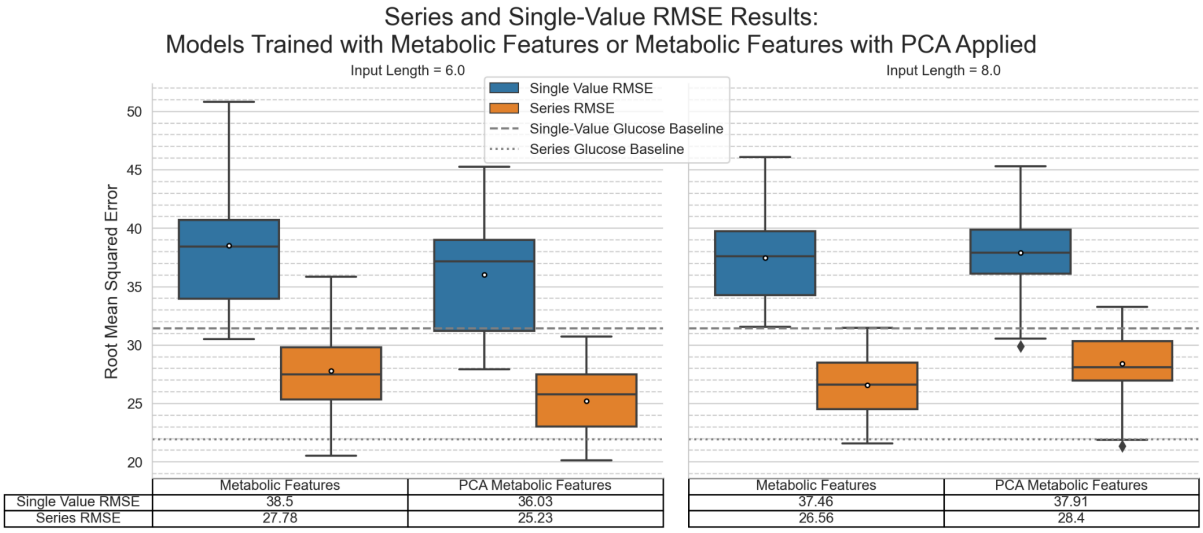


Figure 12. Comparison of single-value and series RMSE values for models trained using the metabolic features (Metabolic Features) and the metabolic features with PCA applied (PCA Metabolic Features) separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results.

8.3.5 Comparison of Results Obtained Using All OhioT1DM Raw Features with PCA Applied and All Raw OhioT1DM Features and Insulin with PCA Applied

The final feature analysis experiment focused on the effects of insulin dosing. Insulin dosing is not a physiological signal, metric, or representative of activity, so it did not fit into activity-focused predictions. On the other hand, insulin is directly related to blood glucose levels, so this experiment compared the impact of insulin when included in training with the metabolic features or the raw OhioT1DM signals. Including insulin with any of the previous configurations should decrease the mean and median RMSEs, as insulin should provide crucial context to some previously unexplained decreases in blood glucose.

Figure 13 compares series and final value RMSE results obtained when insulin dosing was included with the raw OhioT1DM signals for training, and PCA was applied. Figure 14 compares series and final value RMSE results obtained when insulin dosing was included with the metabolic features, and PCA applied. Surprisingly, the inclusion of insulin slightly increased the mean RMSE values for models trained using the raw OhioT1DM features. On the other hand, mean and median RMSE values reported for models trained using the metabolic features decreased with insulin included, as expected.

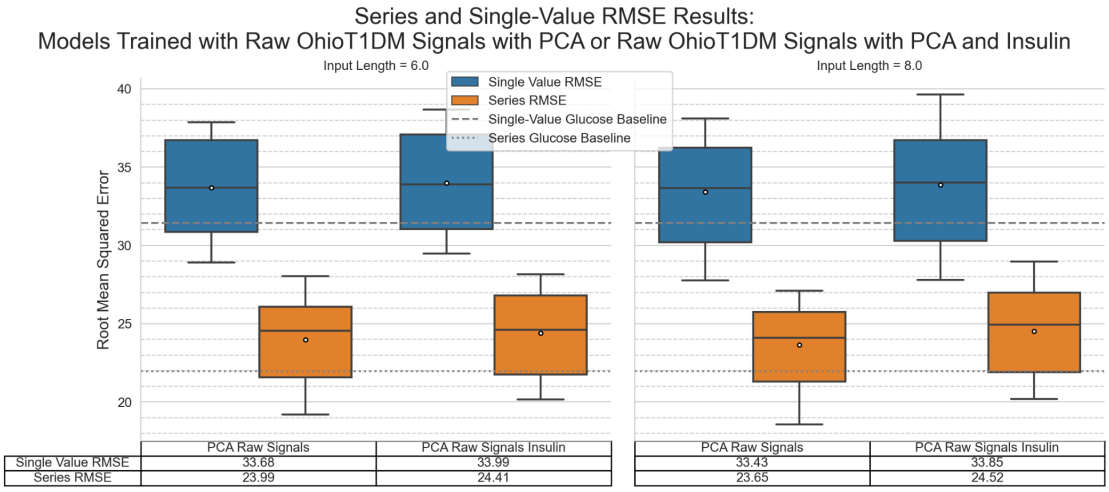


Figure 13. Comparison of single-value and series RMSE values for models trained using the raw OhioT1DM signals with PCA Applied (PCA Raw Signals) and the raw OhioT1DM signals and Insulin with PCA applied (PCA Raw Signals Insulin) separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results.

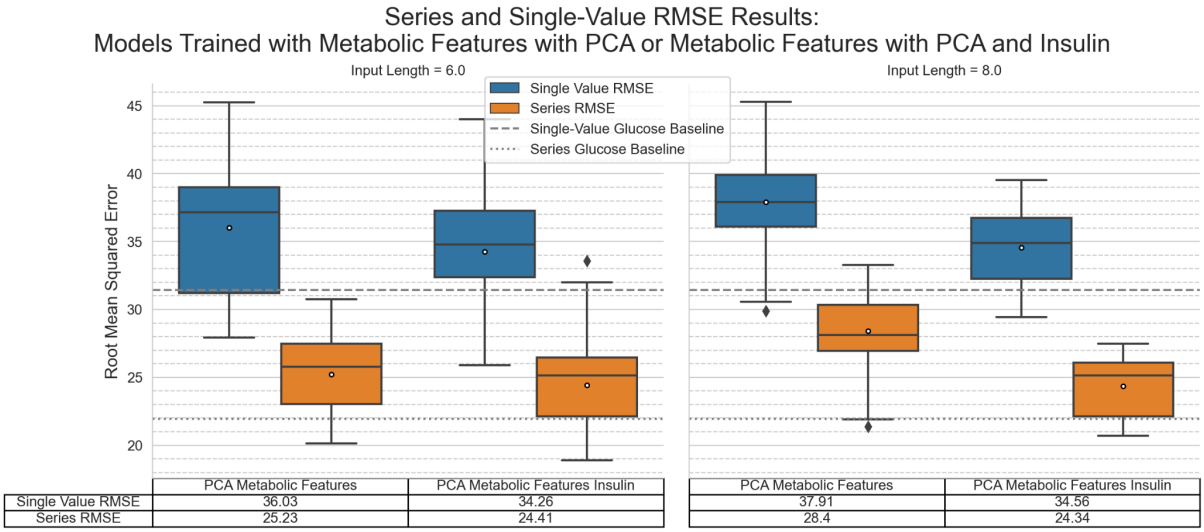


Figure 14. Comparison of single-value and series RMSE values for models trained using the metabolic features with PCA Applied (PCA Metabolic Features) and the metabolic features and Insulin with PCA applied (PCA Metabolic Features Insulin) separated into subplots for 6.0- and 8.0-hour input lengths. The single-value and series baselines, denoted as dotted lines, are the lowest mean RMSE results with an output length of 1.0 hours from Table 9 (8.0-hour output length). The values listed at the bottom are the mean RMSE values for the Single-Value and Series RMSE results.

8.4 SERIES VS. SINGLE VALUE RMSE (OBJECTIVE 3)

The following section looks more closely at the advantages, disadvantages, and limitations of only looking at single-value blood glucose predictions at the end of a specified output length period versus a series of predictions throughout the output length period. Models that produce single-value RMSEs are standardly reported in the literature and are simpler to train, but the models do not provide prediction context, such as how external events impact the rate at which blood glucose changes. The results in this section show measured blood glucose and predicted values from nine randomly sampled prediction windows sub-titled with the prediction index from the testing array. Each sub-plot is a single prediction window from the testing split of the data set. The input sequence of blood glucose values from the OhioT1DM dataset, shown as the blue line, is used by the model before the prediction time to predict the output sequence of blood glucose values. The orange dotted line (measured values) represents the blood glucose measurements from the OhioT1DM dataset after the prediction time. The model's average predicted blood glucose values from all trials are shown as a green point (single value) or line (series) with t green error bands representing the 90% confidence interval. The measured values are compared with the model's predicted output (Prediction) to calculate the error.

Figure 15 shows an example of randomly selected blood glucose predictions vs. time. The left plot in Figure 15 only shows the single-value glucose predictions (green point) at 1 hour, while the right plot shows the entire predicted series of glucose values (green points). The final point in the figure indicates that the predicted final blood glucose value is close to the measured value, but the initial predicted values significantly deviated from the measured values. These two prediction approaches tell different stories, but both are valuable. The model anticipated the input trajectory to continue rising, but the initial blood glucose values decreased before rising toward the end of the output length. On the other hand, the final point alone does not provide any context about how the model interpreted the data and blindly shows that the prediction at 1.0 hours was accurate.

All the experiments in this section aimed to understand the advantages, disadvantages, and limitations of using single-value RMSE versus series RMSE values. The figures in this section look at representative prediction windows that compare the individual predictions with error

values greater than 90% and less than 10% of RMSEs reported by a model configuration using the dataset's testing split. Understanding the trends associated with these cases can be used to identify unique qualitative characteristics that can be used to guide further feature tuning and processing or to determine better qualitative metrics needed to identify unanticipated events. The experiments in the sections below addressed the following:

- Identifying Trends Using Single-Value RMSE
  - Nine predictions with mean single-value RMSEs less than 90% of test set predictions
  - Nine predictions with mean single-value RMSEs greater than the top 90% of test set predictions
- Identifying Trends Using Series RMSE
  - Nine predictions with mean series RMSEs less than 90% of test set predictions
  - Nine predictions with mean series RMSEs greater than 90% of test set predictions
- Comparing features' influence on Series RMSE Predictions Greater than 90% of the Test Set
  - Nine predictions using only blood glucose with mean single-value RMSEs greater than 90% of test set predictions
  - Nine predictions using the raw OhioT1DM signals with mean single-value RMSEs greater than 90% of test set predictions

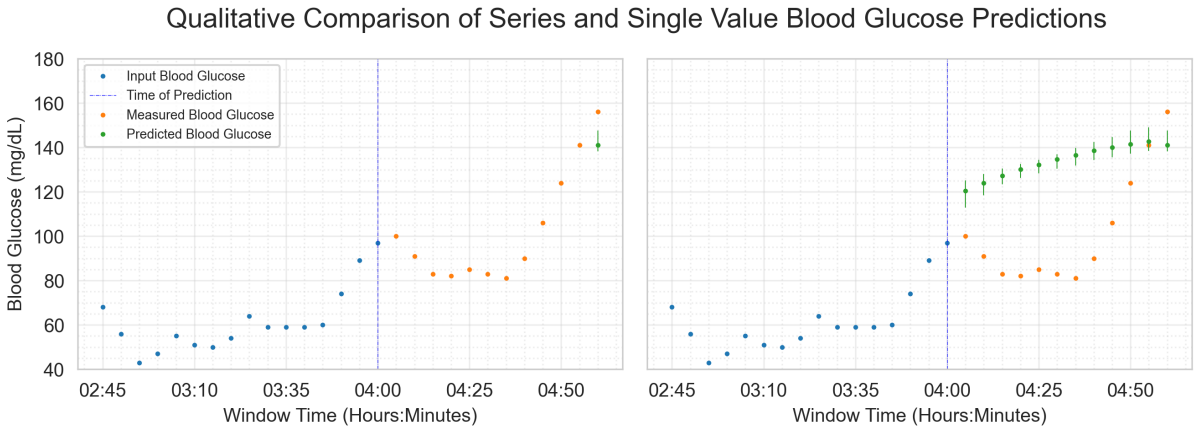


Figure 15. Qualitative comparison between a single-value and a series blood glucose prediction. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

8.4.1 Blood Glucose Predictions with Single-Value RMSEs Greater and Less than 90% of the Test Set

This experiment looked to identify trends and outliers associated with the predictions with the highest and lowest mean single-value RMSE. The single-value RMSE calculation uses

the blood glucose prediction value at 1.0 hours, representing the model's prediction accuracy at 1.0 hours. However, achieving perfect prediction accuracy at the output length of 1.0 hours is extremely difficult because factors such as eating and taking insulin during the prediction window could impact the 1.0-hour blood glucose value. Therefore, prediction windows with low mean RMSE should have small changes in blood glucose during the output length.

Figure 16 and Figure 17 show prediction windows sampled from the experiment that used the raw OhioT1DM signals with PCA applied, an 8.0-hour input length, and a 1.0-hour output length. The predictions in Figure 16 are nine randomly selected prediction windows with a mean single-value RMSE value of less than 90% of the test set. The predictions in Figure 17 are nine randomly selected prediction windows with a mean single-value RMSE greater than 90% of the test set. The titles of each sub-plot identify the prediction in the percentile subset.

The prediction windows in Figure 16, excluding prediction 0, have fairly constant input and output blood glucose values. The single-value RMSE calculation only uses the predicted value at the output length of 1.0 hours, but for all predictions besides prediction 0, the entire predicted blood glucose series matched the measured values. The prediction windows in Figure 17 contain measured values with increasing or decreasing trajectories. Most of the series predictions in Figure 17 do not match the measured values. The predictions in Figure 16 and Figure 17 show constant blood glucose values indicating that the model is more general than specific.

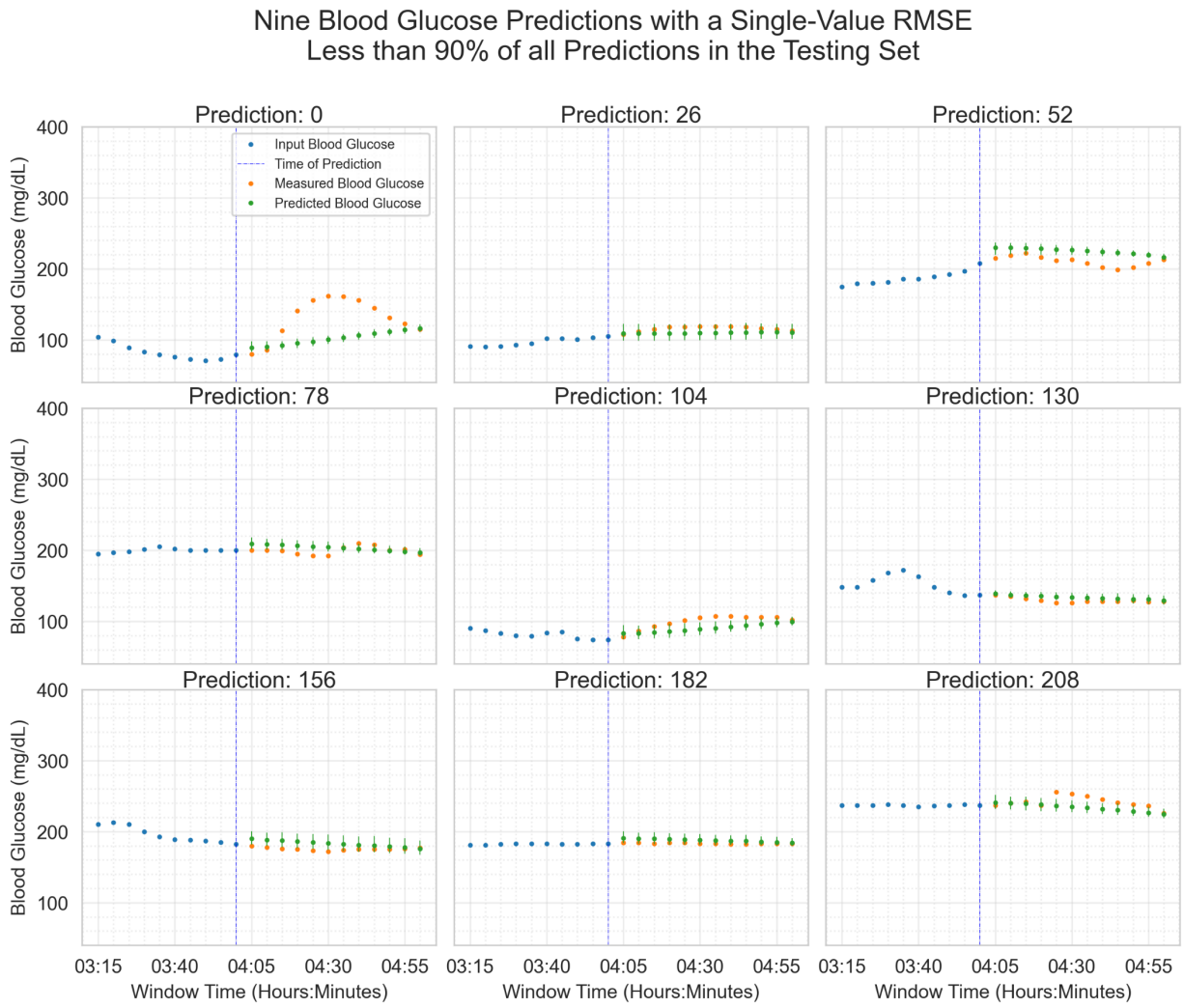


Figure 16. Nine random blood glucose prediction windows with a single-value RMSE less than 90% of all predictions in the testing set. The model configuration used to generate these blood glucose predictions had an input length of 4.0 hours and an output length of 1.0 hours and included the rawOhioT1DM features with PCA applied. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

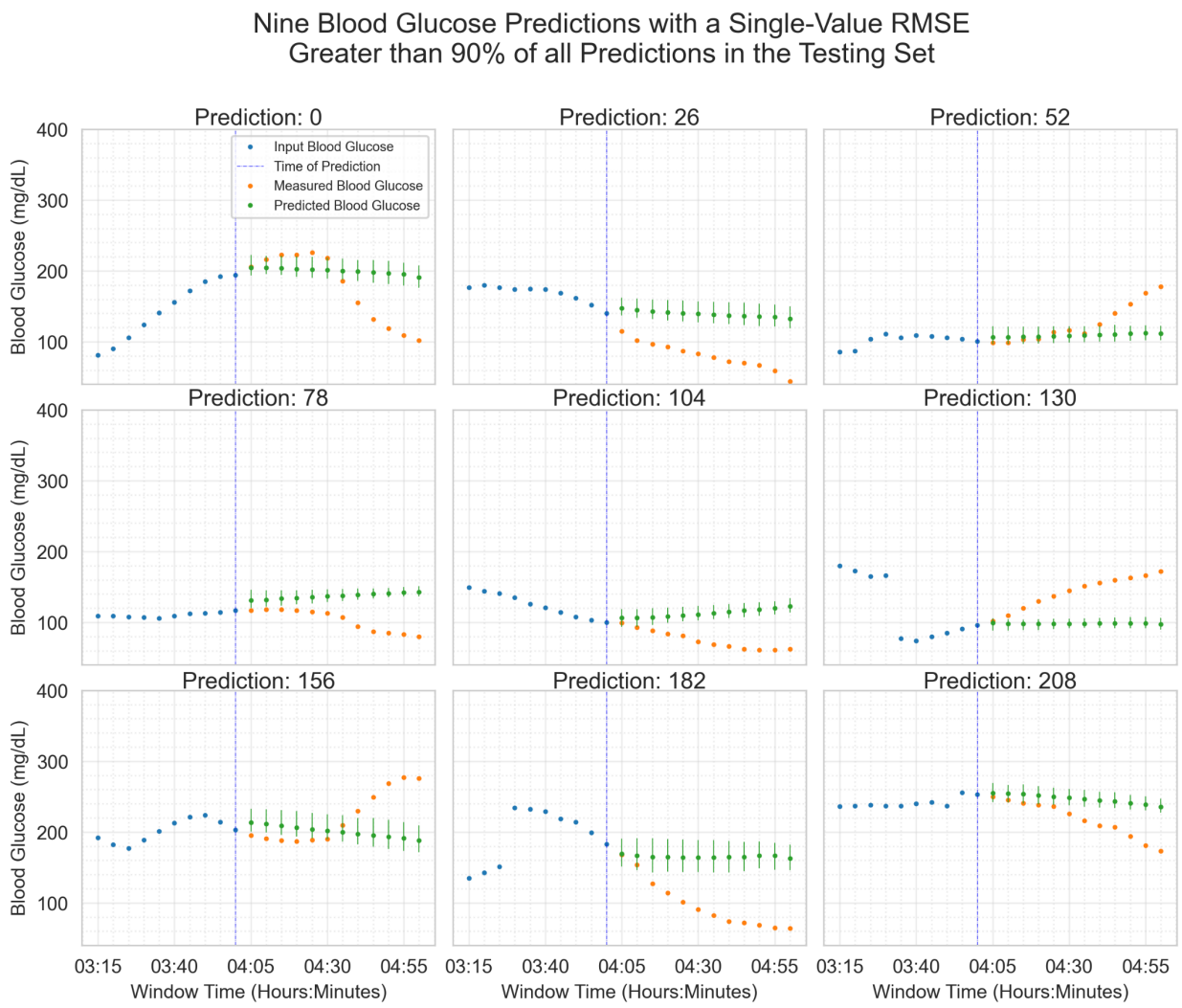


Figure 17. Nine random blood glucose prediction windows with a single-value RMSE greater than 90% of all predictions in the testing set. The model configuration used to generate these blood glucose predictions had an input length of 4.0 hours and an output length of 1.0 hours and included the rawOhioT1DM features with PCA applied. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

8.4.2 Blood Glucose Predictions with Series RMSEs Greater and Less than 90% of the Test Set

The second experiment looked to identify trends and outliers associated with the predictions with the highest and lowest mean series RMSE. The series RMSE is calculated using all predicted values through the output length. The experiment used the series RMSE with an output length of

1.0 hour, representing the model's prediction accuracy of all the predicted points through 1.0 hour. Achieving perfect prediction accuracy through the output length of 1.0 hour is extremely difficult because factors such as eating and taking insulin during the prediction window could impact the 1.0-hour blood glucose value. The series RMSE is less susceptible to blood glucose-impacting events occurring in the middle of the prediction sequence because all the predictions up to the event should still match.

Figure 18 and Figure 19 show prediction windows sampled from the experiment that used the raw OhioT1DM signals with PCA applied, an 8.0-hour input length, and a 1.0-hour output length. The predictions in Figure 18 are nine randomly selected prediction windows with a mean series RMSE value within the top 90% percentile of the sampled set. The prediction in Figure 19 shows nine randomly selected prediction windows with a mean series RMSE below the 90% percentile of the sampled set. The titles of each sub-plot identify the prediction in the percentile subset.

In Figure 18, the blood glucose prediction and measured values remain constant during the output length. The predictions in Figure 19 are also primarily constant, notably different from the measured blood glucose values. In addition, the confidence interval bars on the predictions in Figure 18 are significantly larger than the confidence interval bars on the predictions in Figure 19.

The predictions in Figure 18 have fewer differences between measured and predicted blood glucose values compared to the single-value RMSE results in Figure 16, such as prediction 0, which has some misaligned points throughout the prediction. Furthermore, since all the predicted values generally match the measured values in Figure 18, the predictions with a series RMSE that is less than 90% of all predictions could appear with single-value RMSE predictions that are less than 90% of the population. On the other hand, the predictions included in Figure 17 may not have all the predicted values match the measured values and therefore are less likely to appear in Figure 18. For example, prediction: 26 in Figure 19, the mean series RMSE greater than 90% of predictions, could be a prediction that would appear in Figure 16.

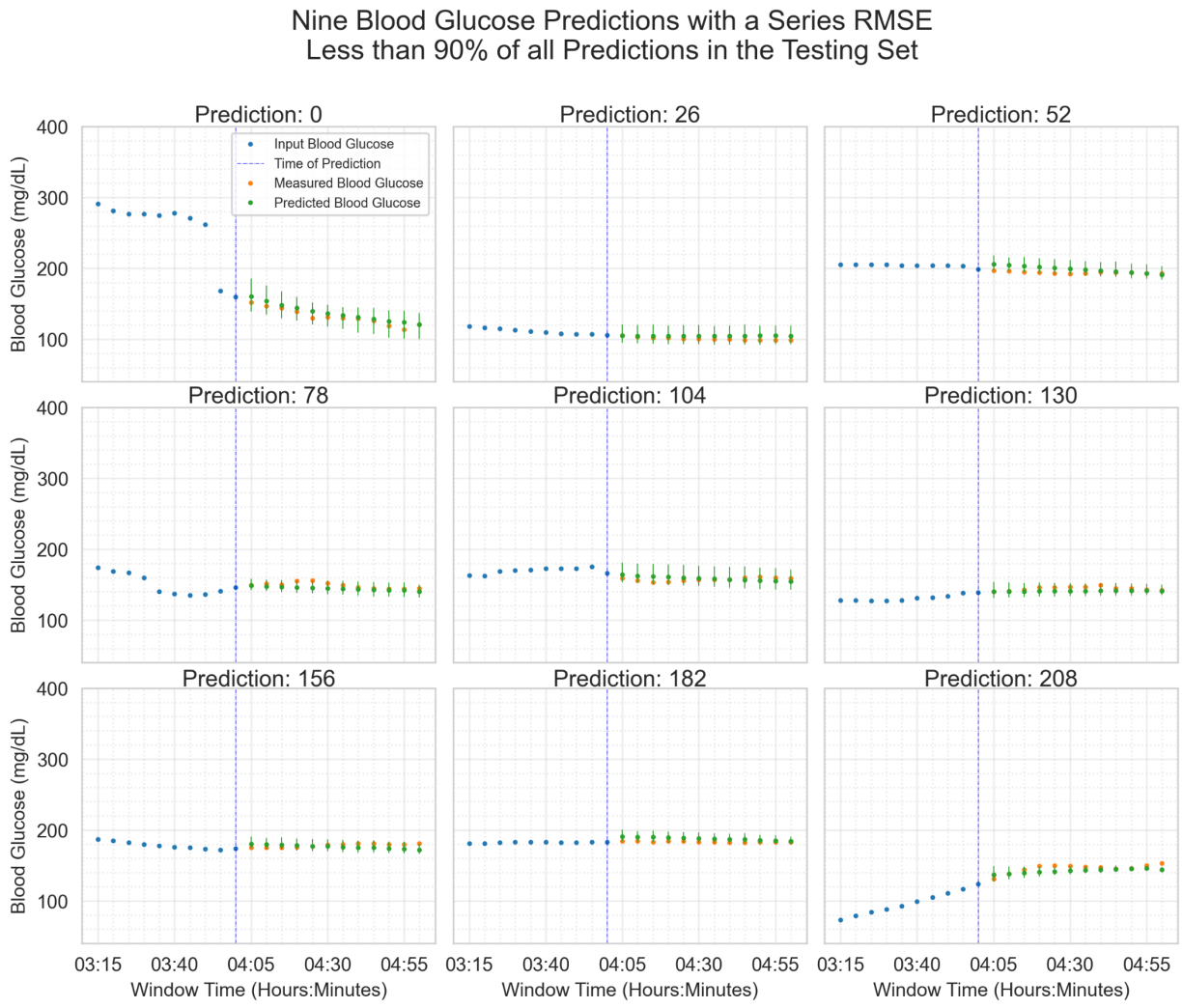


Figure 18. Nine random blood glucose prediction windows with a series RMSE less than 90% of all predictions in the testing set. The model configuration used to generate these blood glucose predictions had an input length of 4.0 hours and an output length of 1.0 hours and included the rawOhioT1DM features with PCA applied. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

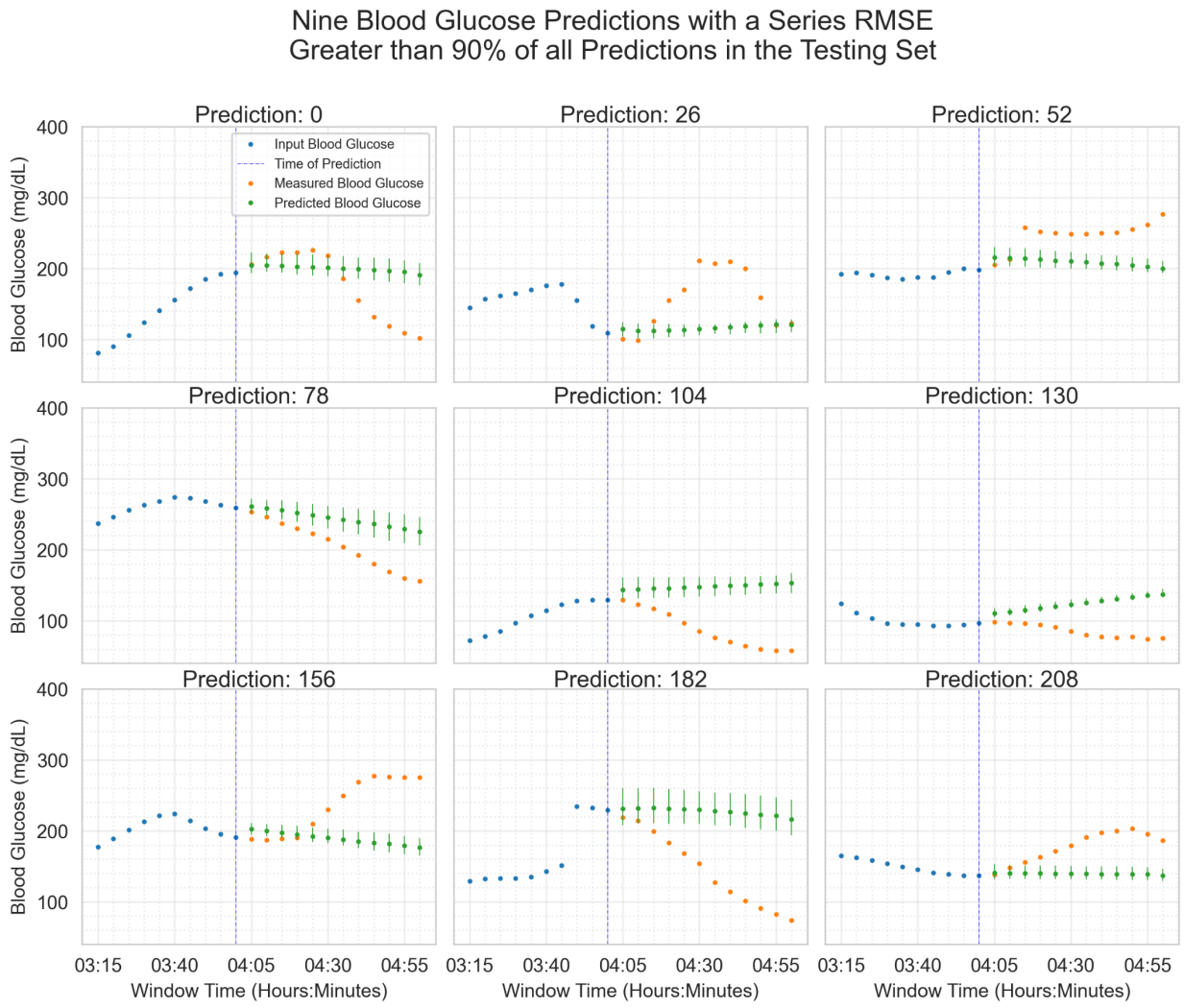


Figure 19. Nine random blood glucose prediction windows with a single-value RMSE greater than 90% of all predictions in the testing set. The model configuration used to generate these blood glucose predictions had an input length of 4.0 hours and an output length of 1.0 hours and included the rawOhioT1DM features with PCA applied. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

8.4.3 Comparison Between Blood Glucose and the Raw OhioT1DM Signals with RMSE Values Greater than 90% of the Testing Set

The final qualitative experiment results compared prediction models that used only blood glucose against prediction models that trained using the raw OhioT1DM signals. Figure 20 and Figure 21 contain both configurations' bottom ninetieth percentile series RMSE results. The goal was to qualitatively assess if there was a distinct difference in the blood glucose trends within the predictions that reported low series RMSE. The prediction models with blood glucose as the only

feature reported the lowest mean series and single-value RMSE values. The results were surprising since models considering additional contextual information were anticipated to extrapolate more about the factors that impacted blood glucose trends.

Figure 20 contains blood glucose predictions from training using only blood glucose, and Figure 21 contains blood glucose values on a model trained using the raw OhioT1DM signals. Both figures contain blood glucose outputs with significant blood glucose changes within the output length. However, the predictions produced by both configurations are horizontal lines which imply little change in blood glucose. Note that all the predictions that used blood glucose have a smaller confidence interval, indicated by the green error bars, than those trained using the OhioT1DM signals. The smaller error bars could indicate that the predictions made using only blood glucose did not have enough contextual information to anticipate the changes in blood glucose during the output length. On the other hand, the large error bars on the predictions that used the raw OhioT1DM signals suggest that the model may have identified an event that could cause changes in blood glucose values, but the predictions were inconsistent between trials and averaged to a horizontal line. Notably, both configurations predicted minimal changes in blood glucose through the output length, potentially indicating that the prediction models are more general than specific.

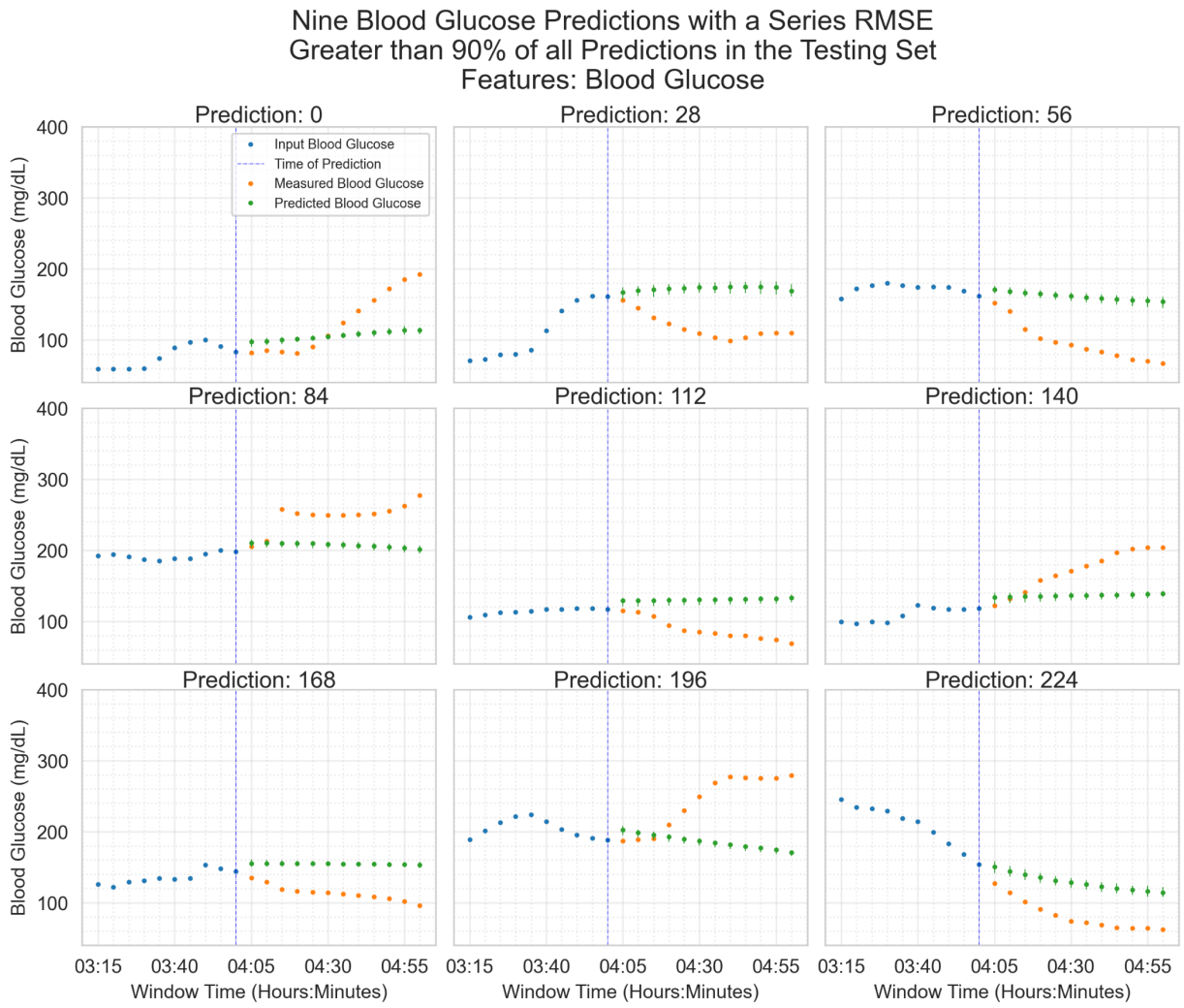


Figure 20. Nine random blood glucose prediction windows with a series RMSE greater than 90% of all predictions in the testing set. The model configuration used to generate these blood glucose predictions had an input length of 4.0 hours, an output length of 1.0 hours, and the blood glucose feature alone. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

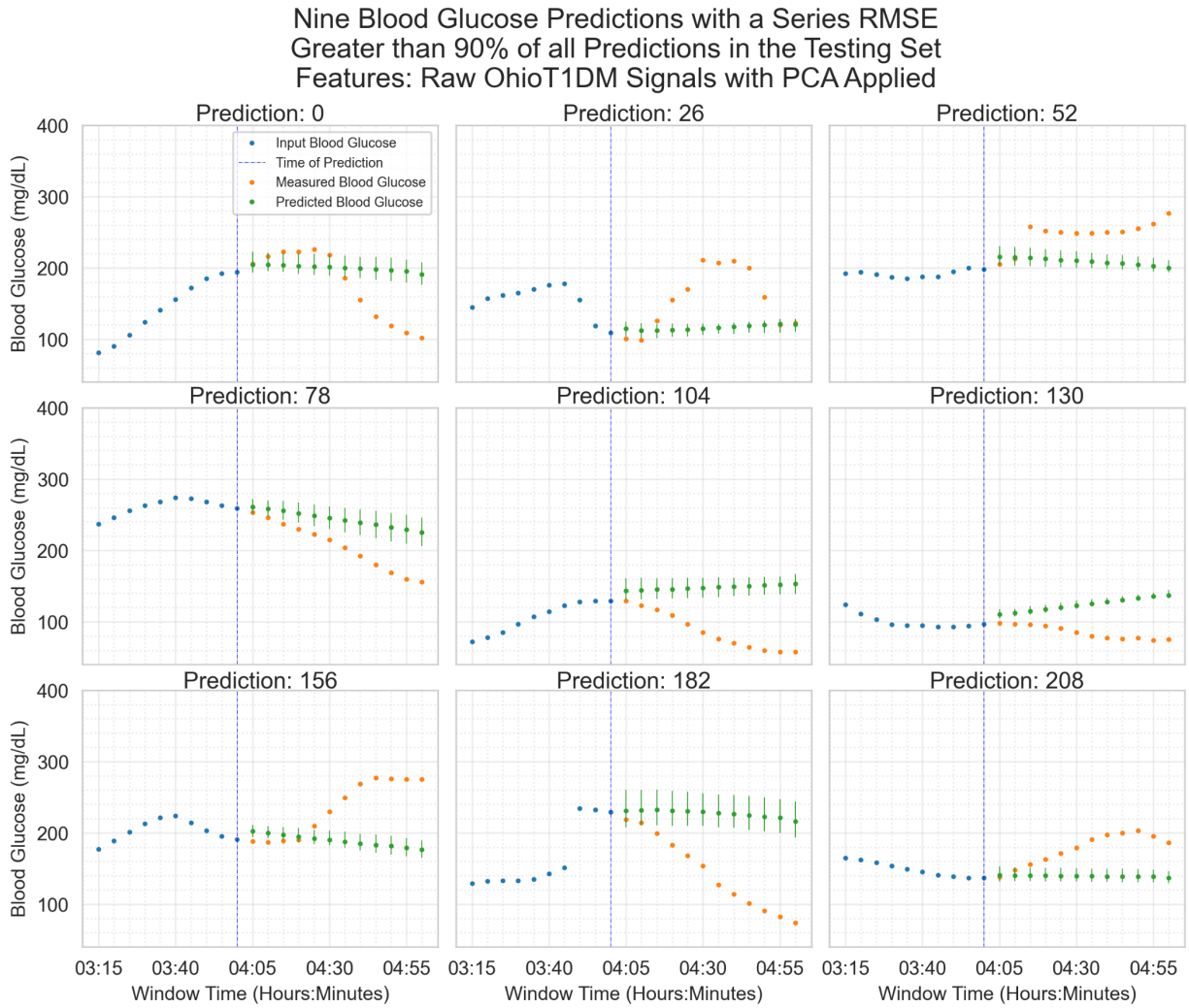


Figure 21. Nine random blood glucose prediction windows with a series RMSE greater than 90% of all predictions in the testing set. The model configuration used to generate these blood glucose predictions had an input length of 4.0 hours and an output length of 1.0 hours and included the raw OhioT1DM features with PCA applied. The error bars on the prediction points represent the 95% prediction interval for all the models trained using the specified configuration.

9 DISCUSSION

The project analyzed how time and activity-focused metrics impact blood glucose model predictions. Specifically, the objectives focused on how a prediction model's input and output dimensions impact prediction accuracy. The impact of a model's input or output length was measured by comparing the differences in RMSE results between experimental configurations. A unique component this project considered was the potential value of using a series prediction instead of the single-value predictions commonly used in literature. The series prediction research was potentially the most valuable contribution because it better depicted how the model utilized the inputs to generate predictions. In addition, using series predictions could provide context that illustrates if the model has a general or specific fit, which is important for understanding how the training data influenced the model's optimization during training.

9.1 INPUT LENGTH (OBJECTIVE 1)

The first experiment that looked at input length expanded on the input analysis done by Martinsson et al. [24]. The results in Table 8 demonstrated the same trend reported by Martinsson et al. [24] with a low mean RMSE for an input length of 1.0 hours that increased through 3.0 hours. The additional input lengths tested in this project beyond 3.0 hours reported lower mean RMSE values with tighter distributions than for input lengths 3.0 hours and below, but the number of prediction windows also decreased. The results of this experiment imply that input lengths beyond 3.0 hours improve prediction accuracy and consistency. The concern with these results is that comparing literal RMSE values calculated using similar but not identical testing sets could be misleading since the methods are not identical.

There is a fundamental need to improve comparison methods when there is the potential that the number of prediction windows will vary. The RMSE values represent the prediction error on a specific dataset, and the same prediction model on different testing sets would inherently result in different RMSE values because the prediction windows were not the same. This project focused on maximizing the number of prediction windows for each trial rather than using the same number for each trial because a larger number of windows covered more unique situations, so the calculated RMSE represented the model's overall performance more.

Additional experimentation could be done to generate standard windows that could be applied to all configurations in the experiment. Comparisons between results that use the same number of prediction windows would be directly comparable, but there would be potentially fewer windows, and RMSE values reported would represent less unique situations. A solution could be to expand the dataset so that there are enough windows representing enough unique situations to be used by all configurations. Alternatively, creating subsets that can be used for all configurations focusing on performance on specific predictions, like highly volatile inputs, could create identifiable results.

9.2 Feature Combinations (Objective 2)

The OhioT1DM data set included many fitness and activity-related signals that could be used to generate feature combinations for model training. Objective 2 aimed to expand on the range of feature combinations published by Mirshekarian et al. [25]. The results by Mirshekarian et al. [25] reported that mean single-value RMSE decreased with additional features included for training. However, across the board, this study found the opposite. Not a single feature combination outperformed models trained using blood glucose alone. The primary differences between this study and Mirshekarian et al. [25] were the use of pre-training datasets and the differentiations between agnostic and internal events. These differences could be why Mirshekarian et al. [25] reported decreased mean RMSE values with additional metrics. In addition, pretraining using simulators allowed the researchers to generate training sets of prediction windows that explicitly demonstrate scenarios that emphasize how each input signal impacts blood glucose. Therefore, models trained using these specialized datasets should have decreased RMSE results because the trained model will initially understand how the signals relate to blood glucose.

Furthermore, the agnostic and internal testing sets used by Mirshekarian et al. [25] contained prediction windows focused on specific scenarios, which were subsets of all windows. Having specific prediction windows should lower the average RMSE by removing unpredictable outliers. Including pretraining and agnostic and internal scenarios allowed Mirshekarian et al. [25] to look at the effects of these methods for a particular research area; however, their RMSE

values would not necessarily be representative of live data in real-world scenarios or comparable to the RMSE values reported in this project.

Few studies reported on the impact of feature reduction techniques, such as PCA. While the overall consideration of features beyond blood glucose did not result in lower mean RMSE values, the inclusion of principal component analysis (PCA) did. Furthermore, the results from the PCA experiments demonstrated that incorporating PCA helped reduce the feature space, input complexity, and training consistency. The results indicate that further research should be done that expands the feature space and considers PCA and other feature space reduction methods.

Of the features considered in this project, insulin is the feature most directly linked to blood glucose and was anticipated to be most likely to affect prediction accuracy. Surprisingly, the results in Figure 13 and Figure 14, that compared models trained using the raw OhioT1DM signals with and without insulin as a feature, showed that mean RMSE increased when insulin was included. The mean RMSE increase could be because the training method did not identify insulin as uniquely important because the insulin did not proportionally change with the blood glucose values. Another reason for the increase in RMSE might be because this project did not consider the meals feature that was always included with insulin in the Mirshekarian et al. [25] experiments. Fundamentally, however, insulin should decrease blood glucose levels following a meal. Thus, the combination of a meal and insulin should not cause significant changes in blood glucose since the insulin decreases the amount of glucose in the blood that the meal introduced. Further research should ensure that the insulin features are accurately represented because insulin is fundamental to blood glucose control and should not decrease prediction accuracy.

The internal and agnostic testing sets are examples of event-based features. Event-based features identify specific scenarios within the dataset, such as a workout. Martinsson et al. [24] used event-based features to test the models in specific scenarios, but the identified events could also be used as feature input into the model. The processing needed to identify the event-based features in an existing dataset is a preprocessing step that requires manually identifying a scenario and then algorithmically identifying the windows within the dataset containing the scenario. Some event-based features that could be identified are events associated with the labels

not used in the project, such as exercise and meals. Further research on feature selection could focus on identifying additional event-based features within the dataset.

9.3 Output Shape - Single-Value and Series (Objective 3)

The series predictions were a unique contribution of this project. The prediction models that predict a blood glucose time series are more computationally intensive but provide invaluable context about how a model processes the input features to generate predictions. Objective 3 focused on a more qualitative evaluation of the commonly used quantitative metrics to ensure that the metrics used to compare models are appropriately measuring the accuracy of a model.

The results from this objective indicated some limitations in relying on single-value RMSE calculations as the sole accuracy metric. For example, predictions like prediction 0 in Figure 16 [single-value RMSE less than 10% of the testing set] would report a low mean single-value RMSE value, but the series output demonstrates that the model did not capture the actual blood glucose trends. The series RMSE calculation measures the prediction error across the entire output length better than the single-value RMSE but is not specific about where the predictions match within the series.

Additionally, all the models tended to predict outputs with little variation, which implies that the models became over generalized. The identification of these general shapes was possible because of the series outputs. The models indicating an extremely general fit to the data are concerning because a general fit is less likely to catch unanticipated events. Unanticipated events are the most important cases to anticipate because these cases could be associated with more life-threatening events like hyperglycemia. Therefore, future research should focus on identifying specifically dangerous cases because the prediction value is more valuable when it can capture unanticipated critical event scenarios.

9.4 Is RMSE The Best Way to Compare Results?

In the literature, the primary measurement of model accuracy is RMSE. However, RMSE is challenging to compare across published results because of differences in testing datasets or

the generation of windows on standard datasets. When comparing RMSE results between datasets, the measured values used to calculate the RMSEs vary. For instance, in the reproduction of the Martinsson et al. [24] study, the RMSE values do not align as closely as expected. The reason for the differences could be a slightly different dataset filter or windowing setup. Without the same sliced dataset, the absolute RMSE value does not represent performance on the same dataset. Li et al. [21, 22] expressed similar concerns about comparisons using RMSE without the same clinical datasets. Furthermore, results with different input lengths cannot be compared since prediction windows exist in shorter input lengths that will not be included in longer lengths.

The best use of RMSE is for comparing directly related data, such as multiple model configurations using the same data split. For example, comparing the experiment results, as defined in this project, helps provide context for how the different configurations changed how the models processed the input features. However, comparisons of experimental RMSE values cannot be made directly if the testing sets are different. Alternatively, the relative performance between methods should be compared across testing sets. This is important because it could examine how well methods perform in additional scenarios, providing context about the universal applicability of the methods.

An option to improve comparisons across literature would be to develop a common testing dataset with standard prediction windows. For example, multiple studies use the OhioT1DM dataset, but the methods used to develop the testing set are inconsistent. One of the issues with using the OhioT1DM dataset as a common testing set is that the signals are inconsistent even between the two released versions. Ideally, a universal testing dataset would contain all sensor measurements and signals that could be considered, but as sensor technology continues to evolve, the set would also have to evolve.

9.5 Potential Limitations

Currently, the primary dataset used to evaluate blood glucose prediction algorithms is the OhioT1DM dataset. While the dataset is easily accessible, comparing RMSE results obtained from the dataset is difficult. Each study may approach removing outliers and interpolating

missing segments differently, leading to mismatched prediction windows. Differences in the study approach limited how the results in this project could be compared to other related studies.

The filtering and windowing of the dataset particularly hindered the comparison of input time lengths. The way the pipeline was designed, a trial's input length was used to generate the prediction windows. Therefore, the number of windows was inconsistent between trials where the input lengths varied. Future work should scale the windows to be consistent for all potential trials and configurations to ensure valid comparisons.

The project also incorporated the calculation of health metrics from raw signals. The initial purpose of these calculations was to simplify the transition from the preliminary study using features in Apple Health to the features in the OhioT1DM dataset. The transition between datasets was significant because it emphasized that not all datasets contained the same signals and metrics. For example, the 2018 and 2020 OhioT1DM datasets contained different signals, so the 2020 participants were not compared in this project. The project's codebase and pipeline were designed to implement the signals found in the OhioT1DM 2018 dataset but are incompatible with other datasets. Furthermore, the fitness band participants in the 2018 study used a discontinued product. Future research should consider developing methods that generate universal features from various signals.

The models developed in this project were designed to produce consistent, high-volume results for many configurations. The methods did not focus on optimizing every model for each configuration because running all the possible configurations until they converged optimally for many iterations takes extreme time. Therefore, the decision was made to sacrifice the minimization of RMSE to decrease the model training time. In addition, comparing combinations was the focus of this project, so the decision was made to maximize the number of combinations compared. This project aimed to evaluate as many combinations as possible to make future research more streamlined.

Smaller pilot experiments were performed throughout the study, including testing batch sizes 32, 256, 512, and 1024. The batch size experiment aimed to determine how batch size impacted predictions and training time. The experimentation uncovered a discrepancy in RMSE calculations which heavily impacted smaller batch sizes. For example, Figure 22 shows RMSE

values obtained using the TensorFlow RMSE method in blue and orange and results from a manual RMSE calculation that used the predicted blood glucose outputs in red and green (single batch). The lower batch size results showed large differences between the two calculations. The differences occurred because the TensorFlow RMSE method calculates RMSE per batch and averages all the batches together, while the manual calculations determine RMSE across all predictions. Since RMSE magnitude cannot be compared outside of an experiment and all the experiments used a common batch size, the pipeline in this project implemented batch-based calculations. Furthermore, since the TensorFlow RMSE method is commonly implemented in literature, the method was used for consistency. However, further investigation into RMSE and RMSE calculations should be done while investigating better prediction analysis methods.

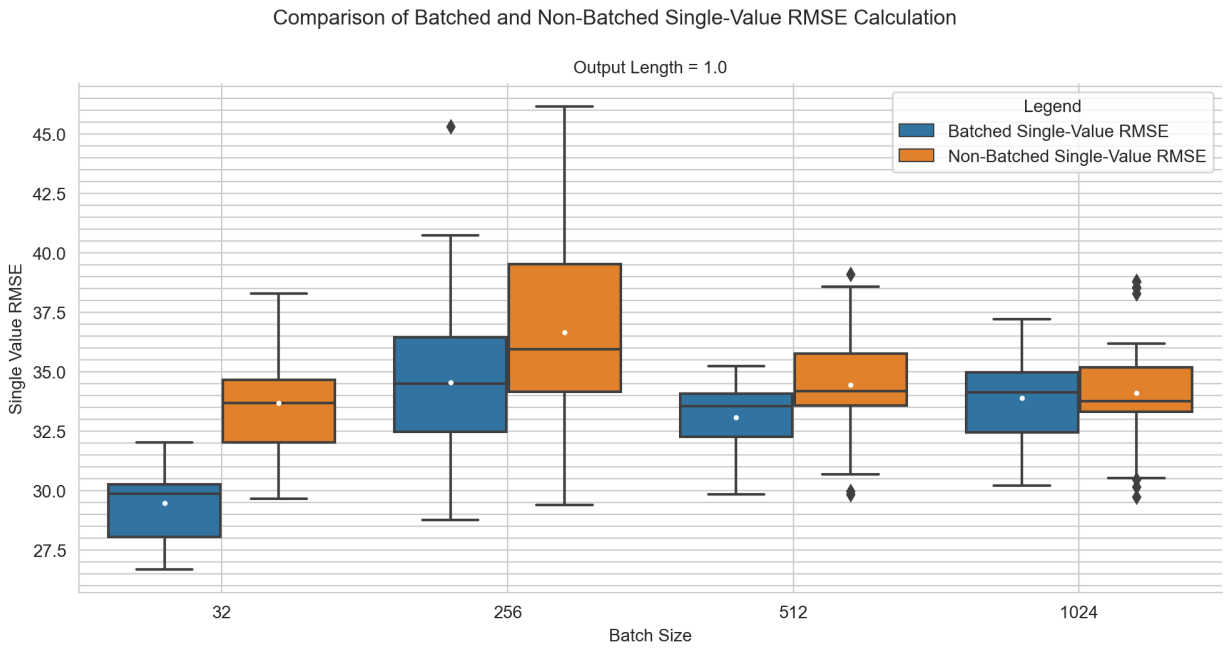


Figure 22. Comparison of single-value RMSE calculations of batched and non-batched predictions. The results were collected during method development using a model configuration with the raw OhioT1DM signals with an input length of 3.0 hours. The white dot indicates the mean single-value RMSE.

9.6 Next Steps

This project uncovered more questions than answers. The number of potential variables that impact model development in human physiology is vast. Determining how to measure loss best and compare results best are some of the most significant areas for continued research.

Along with the loss functions, generating a standardized way to embed health metrics into a consistent shape would be an invaluable research opportunity, no matter what metrics are included. A goal for future research is to develop a model that does not rely on specific health metrics because a model that can work with a variety of related inputs would be easier to implement universally.

The first and most crucial step would be obtaining a more extensive health and physiological signal database. The OhioT1DM dataset is great for initial research, but the datasets only contain six participants, and the 2018 study uses a wearable sensor that has been discontinued. A larger dataset with blood glucose and health metrics would be ideal but not always possible. A parallel research path could be implementing research methods that generate additional features from health metrics collected from common fitness trackers. Identifying and processing additional features would assist researchers who want to expand the feature combinations in this project.

The larger dataset would also allow fine-tuning pre-trained general models for individuals using transfer learning. Transfer learning starts with a general model trained on a large data set. Then, the training is finished using a specific subset of data so that the model explicitly fits the subset of data. The initial training on the larger data set requires significant processing time, but the specific training requires very little. The specific training is ideal for weaker platforms, such as mobile devices, which was a design goal in the related literature. The research by Mirshekarian et al. [25] incorporated transfer learning by pretraining on simulation data which accomplished the same goal as a larger dataset, but the larger dataset would provide a better testing set. Research into transfer learning is recommended, especially if the research use case is consumer products.

The most significant issue uncovered in this project was comparing predictions using the literature standard RMSE metrics. The results and discussion specifically related to Objective 1 focused on the issues with comparisons using RMSE when the configurations require modifying variables that impact window length or the number of windows. For example, the RMSE metrics required comparisons on identical datasets, which caused problems for the Objective 1 experiments that modified the input length, directly impacting the window length and the number

of windows. Continued research will likely encounter similar challenges, so the next step should address the window generation or comparison methods.

The biggest challenge in this project was determining what was considered an "accurate" prediction. Qualitative assessment of accuracy was particularly problematic while debugging the loss functions. The loss function design was only briefly considered in this project due to scope but is another component incorporating a formula to determine model accuracy. Many models generated during this project were often overgeneralized, with the glucose prediction variability significantly lower than the measured glucose values. While generalized models can be a consequence of multiple factors, the loss function directly affects how the model was optimized to become more general in nature. Future research on accuracy measurements should also look at loss functions since the metrics are inherently related.

10 CONCLUSIONS

During the 53rd American Diabetes Association Annual Meeting [34] in 1993, the groundbreaking Diabetes Control and Complications Trials (DCCT) [35] was presented, confirming for the first time that the management of glucose in individuals with diabetes was important. The DCCT study from 1993 is the foundation of all diabetes research, including this project, focused on assisting and improving glucose control for individuals with diabetes. The health and diabetes data used in this thesis are a product of the ongoing diabetes research that advanced glucose monitoring technologies to where they are today.

This project aimed to incorporate historical health data collected from wearable technology into a machine-learning prediction algorithm and identify how the incorporated health data impacted prediction accuracy. The first two objectives aimed to determine how the amount of time-series data and activity-focused feature combinations impacted the prediction accuracy of blood glucose values output from a deep-learning neural network. The findings from these objectives suggest that using an increased input length could improve predictions, but including additional physiological features did not. The health features included as additional inputs to the blood glucose prediction models unexpectedly decreased the prediction accuracy. Additionally,

difficulties were encountered when comparing results because the RMSE calculations alone provided limited context around why the predictions were inaccurate.

Objective 3 focused on qualitatively comparing series and single-value outputs to understand better the missing context around how the model generated predictions for the different inputs. The challenges encountered when directly comparing RMSE values resulted in Objective 3. The results from Objective 3 suggest that both single-value and series RMSE calculations perform best on steady-state blood glucose values. Furthermore, the series output approach emphasized that models provided a more general than specific fit. Overly general model fits could be more concerning as a predictive tool since these might not predict if glucose values moved quickly toward a dangerous zone.

The most unique contribution of the thesis was the implementation and analysis of using both single value and series outputs. The series outputs provided essential context for analyzing the results and debugging the pipeline during method development. Additionally, this project uniquely focused on the results and how they demonstrated that RMSE should not be used as a universal comparison metric in every situation because if prediction windows are not identical, the accuracy metric is not directly comparable.

The fundamental goal was to identify how key features input into machine learning models impact prediction accuracy so that the predictive techniques can be used in closed-loop technology for blood glucose control for people with diabetes. The contributions from this project are an important step towards the continued research on implementing additional signals and features and how these improve prediction accuracy. The next step for research in this area should be to address how to compare prediction models so that the results provide more context to the researchers and end users.

11 ACKNOWLEDGMENTS

I want to acknowledge the financial support of the Rochester Institute of Technology for this project.

I want to thank my advisor, Dr. Lamkin-Kennard, for providing guidance, feedback, and support throughout this research project. I would also like to thank the rest of my thesis committee: Dr. Kolodziej and Dr. Heard, for their time and patience during these unprecedented times.

12 REFERENCES

[1] C. Marling and R. Bunescu, "The OhioT1DM Dataset for Blood Glucose Level Prediction: Update 2020," (in eng), *CEUR Workshop Proc,* vol. 2675, pp. 71-74, Sep 2020.

[2] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning* (no. 4). Springer, 2006.

[3] S. A. Amiel, "The consequences of hypoglycaemia," *Diabetologia,* vol. 64, no. 5, pp. 963-970, 2021-05-01 2021, doi: 10.1007/s00125-020-05366-3.

[4] srunyon, *Is my Dexcom CGM sensor accurate?* Dexcom.

[5] B. A. Buckingham *et al.*, "Safety and Feasibility of the OmniPod Hybrid Closed-Loop System in Adult, Adolescent, and Pediatric Patients with Type 1 Diabetes Using a Personalized Model Predictive Control Algorithm," *Diabetes Technology & Therapeutics,* vol. 20, no. 4, pp. 257-262, 2018-04-01 2018, doi: 10.1089/dia.2017.0346.

[6] J. L. Sherr *et al.*, "Safety and Performance of the Omnipod Hybrid Closed-Loop System in Adu lts, Adolescents, and Children with Type 1 Diabetes Over 5 Days Under Free-Living Conditions," *Diabetes Technology & Therapeutics,* vol. 22, pp. 174-184, 2020/3//, doi: 10.1089/dia.2019.0286.

[7] T. Teich, D. P. Zaharieva, and M. C. Riddell, "Advances in Exercise, Physical Activity, and Diabetes Mellitus," *Diabetes Technology & Therapeutics,* vol. 21, no. S1, pp. S-112-S-122, 2019-02-01 2019, doi: 10.1089/dia.2019.2509.

[8] "The Official Journal of ATTD Advanced Technologies & Treatments for Diabetes Conference 22 ‐ 25 February 2023 I Berlin & Online," *Diabetes Technology & Therapeutics,* vol. 25, no. S2, pp. A-1-A-269, 2023, doi: 10.1089/dia.2023.2525.abstracts.

[9] N. McCullum, *Deep Learning Neural Networks Explained in Plain English*. freeCodeCamp.org.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[11] H. F. Jelinek *et al.*, "Clinical profiles, comorbidities and complications of type 2 diabetes mellitus in patients from United Arab Emirates," *BMJ Open Diabetes Research & Care,* vol. 5, no. 1, p. e000427, 2017-08-01 2017, doi: 10.1136/bmjdrc-2017-000427.

[12] D. J. Kim, "The Epidemiology of Diabetes in Korea," *Diabetes & Metabolism Journal,* vol. 35, no. 4, p. 303, 2011-01-01 2011, doi: 10.4093/dmj.2011.35.4.303.

[13] A. Brown. "42 Factors That Affect Blood Glucose?! A Surprising Update." diaTribe. (accessed 2020).

[14] A. Sanchez-Comas, K. Synnes, D. Molina-Estren, A. Troncoso-Palacio, and Z. Comas-González, "Correlation Analysis of Different Measurement Places of Galvanic Skin Response in Test Groups Facing Pleasant and Unpleasant Stimuli," *Sensors,* vol. 21, no. 12, p. 4210, 2021-06-19 2021, doi: 10.3390/s21124210.

[15] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Deep Learning for Diabetes: A Systematic Review," *IEEE Journal of Biomedical and Health Informatics,* vol. 25, no. 7, pp. 2744-2757, 2021-07-01 2021, doi: 10.1109/jbhi.2020.3040225.

[16] J. B. Welsh *et al.*, "Accuracy, Utilization, and Effectiveness Comparisons of Different Continuous Glucose Monitoring Systems," *Diabetes Technology & Therapeutics,* vol. 21, no. 3, pp. 128-132, 2019-03-01 2019, doi: 10.1089/dia.2018.0374.
[17] D. A. Finan *et al.*, "Experimental Evaluation of a Recursive Model Identification Technique for Type 1 Diabetes," *Journal of Diabetes Science and Technology,* vol. 3, no. 5, pp. 1192-1202, 2009-09-01 2009, doi: 10.1177/193229680900300526.
[18] R. A. Sowah, A. A. Bampoe-Addo, S. K. Armoo, F. K. Saalia, F. Gatsi, and B. Sarkodie-Mensah, "Design and Development of Diabetes Management System Using Machine Learning," *International Journal of Telemedicine and Applications,* vol. 2020, p. 8870141, 2020/07/16 2020, doi: 10.1155/2020/8870141.
[19] F. L. Schwartz, C. R. Marling, and R. C. Bunescu, "The Promise and Perils of Wearable Physiological Sensors for Diabetes Management," *Journal of Diabetes Science and Technology,* vol. 12, no. 3, pp. 587-591, 2018-05-01 2018, doi: 10.1177/1932296818763228.
[20] W. P. T. M. Van Doorn *et al.*, "Machine learning-based glucose prediction with use of continuous glucose and physical activity monitoring data: The Maastricht Study," *PLOS ONE,* vol. 16, no. 6, p. e0253125, 2021-06-24 2021, doi: 10.1371/journal.pone.0253125.
[21] K. Li, C. Liu, T. Zhu, P. Herrero, and P. Georgiou, "GluNet: A Deep Learning Framework for Accurate Glucose Forecasting," *IEEE Journal of Biomedical and Health Informatics,* vol. 24, no. 2, pp. 414-423, 2020-02-01 2020, doi: 10.1109/jbhi.2019.2931842.
[22] K. Li, J. Daniels, C. Liu, P. Herrero, and P. Georgiou, "Convolutional Recurrent Neural Networks for Glucose Prediction," *IEEE Journal of Biomedical and Health Informatics,* vol. 24, no. 2, pp. 603-613, 2020-02-01 2020, doi: 10.1109/jbhi.2019.2908488.
[23] C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli, "The UVA/PADOVA Type 1 Diabetes Simulator," *Journal of Diabetes Science and Technology,* vol. 8, no. 1, pp. 26-34, 2014-01-01 2014, doi: 10.1177/1932296813514502.
[24] J. Martinsson, A. Schliep, B. Eliasson, and O. Mogren, "Blood Glucose Prediction with Variance Estimation Using Recurrent Neural Networks," *Journal of Healthcare Informatics Research,* vol. 4, no. 1, pp. 1-18, 2020-03-01 2020, doi: 10.1007/s41666-019-00059-y.
[25] S. Mirshekarian, H. Shen, R. Bunescu, and C. Marling, "LSTMs and Neural Attention Models for Blood Glucose Prediction: Comparative Experiments on Real and Synthetic Data," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019-07-01 2019: IEEE, doi: 10.1109/embc.2019.8856940.
[26] E. D. Lehmann, "Research use of the AIDA www.2aida.org diabetes software simulation program: a review--part 2. Generating simulated blood glucose data for prototype validation," (in eng), *Diabetes Technol Ther,* vol. 5, no. 4, pp. 641-51, 2003, doi: 10.1089/152091503322250668.
[27] K. Team, *Keras documentation: RMSprop*. Keras.io.
[28] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[29] W. McKinney, "Data Structures for Statistical Computing in Python," presented at the Proceedings of the 9th Python in Science Conference, 2010, 2010. [Online]. Available: http://dx.doi.org/10.25080/Majora-92bf1922-00a.
[30] A. Martín *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015 2015.
[31] L. R. Keytel *et al.*, "Prediction of energy expenditure from heart rate monitoring during submaximal exercise," (in eng), *J Sports Sci,* vol. 23, no. 3, pp. 289-97, Mar 2005, doi: 10.1080/02640410470001730089.
[32] C. D. Fryar, M. D. Carroll, Q. Gu, J. Afful, and C. L. Ogden, "Anthropometric Reference Data for Children and Adults: United States, 2015-2018," (in eng), *Vital Health Stat 3,* no. 36, pp. 1-44, Jan 2021.
[33] L. Ferath Kherif and Adeliya, "Chapter 12 - Principal component analysis," in *Machine Learning*, V. Andrea Mechelli and Sandra Ed.: Academic Press, 2020, pp. 209-225.
[34] S. Kumar, "The 53rd American Diabetes Association annual meeting and scientific sessions, Las Vegas, Nevada, USA," (in eng), *Diabet Med,* vol. 11, no. 1, pp. 120-2, Jan-Feb 1994, doi: 10.1111/j.1464-5491.1994.tb00242.x.
[35] "The Effect of Intensive Treatment of Diabetes on the Development and Progression of Long-Term Complications in Insulin-Dependent Diabetes Mellitus," *New England Journal of Medicine,* vol. 329, no. 14, pp. 977-986, 1993-09-30 1993, doi: 10.1056/nejm199309303291401.

13 APPENDIX

13.1 CODE BASE AND FILE STRUCTURE

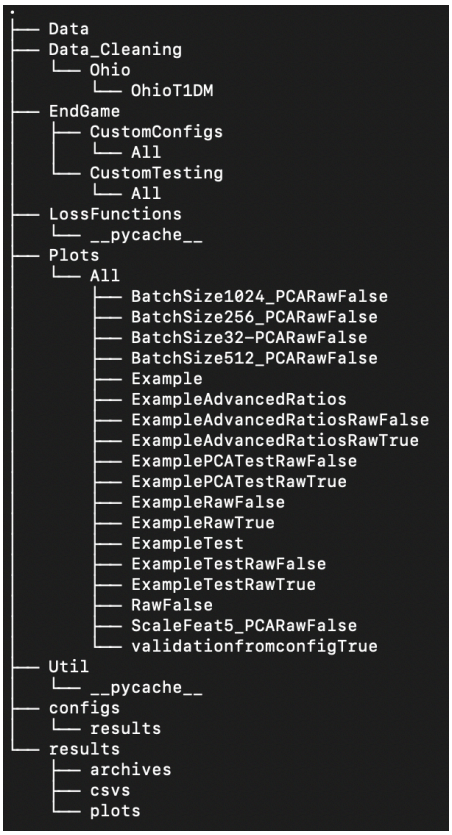


Figure 23. Tree diagram depicting the project's directory organization. The code was uploaded at https://github.com/arearson/BloodGlucose-Prediction. Description of specific files can be found in the README.md at the root of the repository.

The project files are organized in the GitHub repository, https://github.com/arearson/BloodGlucose-Prediction, with the file structure visualized in Figure 23. The repository 'README.md' contains specific information on installing and running the specific files in the project. The data and data_cleaning directories contain the data files and initial filtering Python code. The EndGame directory contains the results for all configurations. The plots directory contains the results from development and methods experiments like the batch size results shown in Figure 22. The custom loss functions described in section 0, **Loss Functions**, are found in the LossFunctions directory. The Util directory contains the discord

notification tool that tracks the training process and sends notifications when the training fails. The results directory contains all the figure, tables, and results files described in this project.

13.2 Configuration Files

The configuration file shown in Figure 24 is an example file written in the Config ini format that would be interpreted using the configparser.py library to define the experimental parameters. The DEFAULT category was used to define where results were saved and how the results were named. The DATA category was used to define how the dataset was preprocessed before being split into inputs and outputs. The MODEL category defined the dataset's batch size and the model's hyperparameters. The RUN category defined the number of iterations each configuration would run and what input and output lengths were used. The ADVANCED category was used for experimental variables during pipeline development.

```
[DEFAULT]
folder = Plots
word = Example
model-id = 600-3-1024-example-custom2
foldervalue = All
timecolumn = ts

[DATA]
interp = 1
scale-bg = 600
scale-all = 3
pca = True
only_bg = True
last = False

[MODEL]
batchsize = 1024
learning_rate = 0.0050
model = Custom
loss = Custom2
amsgrad = False
epochs = 100

[RUN]
count = 1
in-hours = 1, 3, 6, 7, 8
out-hours = 0.5, 1.0

[ADVANCED]
ratios = 0.100, 0.000, 0.700
embed = False
```

Figure 24. Example configuration used to define the experimental parameters.

13.3 Example of Training and Validation Loss Plot

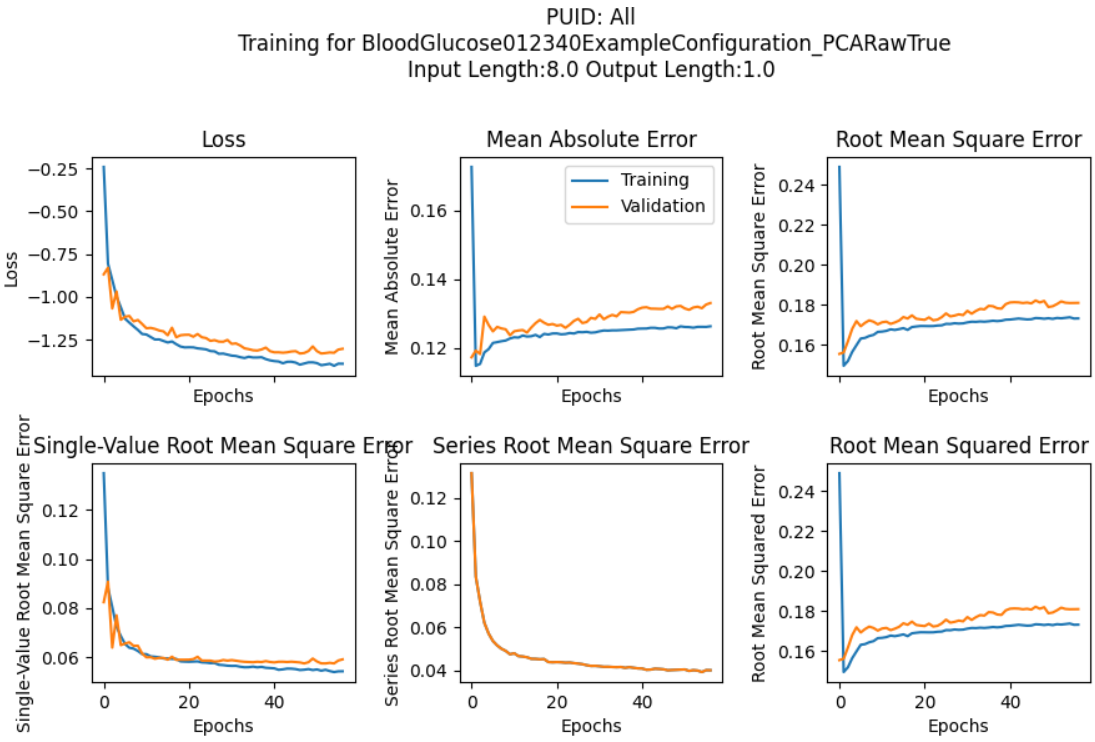


Figure 25. Example of Training and Validation Loss Plot.