The Challenge of Predicting Future Blood Glucose for Patients with Type I Diabetes

by

Neil C. Borle

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

Patients with Type I Diabetes (T1D) must take insulin injections to prevent the serious long term effects of hyperglycemia – high blood glucose (BG). These patients must also be careful not to inject too much insulin because this could induce hypoglycemia (low BG), which can be fatal. Patients therefore follow a "regimen" that, based on various measures, determines how much insulin to inject at certain times. Current methods for managing this disease require the manual adjustment of a patient's regimen over time based on the disease's behavior (recorded in the patient's diabetes diary). This is both time consuming and error-prone. If we can accurately predict a patient's future BG values from their current features (*e.g.*, predicting today's lunch BG value given today's diabetes diary entry for breakfast, including insulin injections), then it is relatively easy to produce an effective regimen. This study explores the challenges of BG modeling by applying a number of machine learning algorithms, as well as various data preprocessing variations (corresponding to 312 [learner, dataset] combinations), to a new T1D dataset that contains 30,221 entries from 51 different patients. Our most accurate predictor is a weighted ensemble of two Gaussian Process Regression (GPR) models where GPR#1 is learned using a patient's entire history (over all meals) and GPR#2 is learned using data from individual meals. This ensemble achieved an err_{L1} loss of 2.72 mmol/L. This was an unexpectedly poor result given that one can obtain an err_{L1} of 2.94 mmol/L using the naive approach of only predicting the patients average BG. These results suggest that accurate BG prediction models may not be obtainable from the diabetes diary data that is typically collected; additional data may be necessary to build fine-grained BG control systems that use BG prediction models.

Preface

The description of K-means clustering (Section 2.3.2) and silhouette plots as a means of determining the K-means parameter have been previously made public as a PeerJ Preprint. "Borle, Neil C., et al. 'Analyzing test driven development based on GitHub evidence.' PeerJ Preprints 4 (2016): e1920v3."

To my parents and my wife,
For all your love and support.

Todo aquel que piense que la vida es desigual, tiene que saber que no es así, que la vida es una hermosura, hay que vivirla.

– Celia Cruz, *Mi Vida Es Cantar*, 1998.

Acknowledgements

First, I would like to thank my supervisor, Dr. Russell Greiner, for his feedback and guidance throughout my degree. In particular, I appreciate all the effort he put into helping me to communicate more effectively in scientific writing. I would also like to thank Dr. Edmond A. Ryan for his feedback, for participating in our experiments and for helping me to understand a diabetologist's thought process and point of view. I would like to thank Dr. Osmar Zaïane for reviewing and providing feedback on this thesis.

I would like to thank all my friends and coworkers who have journeyed with me through my graduate school experience. From the Dr. Greiner Lab I would like to thank Mina Gheiratmand, Tanvir Sajed, Negar Hassanpour, Bhaskar Sen, Luke Nitish Kumar and Roberto Vega for all your conversations, insights and kindness. From Software Engineering, I would like to thank Meysam Feghhi and Stephen Romansky for our collaborations and conversations, as well as Dr. Abram Hindle for his advice and encouragement. Thank you also to the Natural Sciences and Engineering Research Council of Canada (NSERC) for the Canada Graduate Scholarships-Masters Program (CGS-M) scholarship.

Finally, I would like to thank my parents (Alvin and Sandy), my brother, Sean, and my wife, Stephanie, for all your love and support, and for making the completion of this thesis possible.

Table of Contents

1 Introduction **1**
- 1.1 Motivation and Context . . . 1
  - 1.1.1 Our Specific Task . . . 3
- 1.2 Contributions . . . 3
- 1.3 Outline . . . 4

2 Background **5**
- 2.1 Type I Diabetes . . . 5
- 2.2 Towards Automated Type I Diabetes Management . . . 6
  - 2.2.1 Limitations of Prior Diabetes Modeling Studies . . . 6
  - 2.2.2 Additional Background . . . 7
- 2.3 Machine Learning Algorithms . . . 8
  - 2.3.1 Supervised Learning Algorithms . . . 8
  - 2.3.2 Unsupervised Learning Algorithms . . . 13

3 Methods: Machine Learning Applied to Type I Diabetes **16**
- 3.1 Data . . . 16
- 3.2 Data Preprocessing . . . 19
- 3.3 Expert Feedback on Deciding When to Predict . . . 20
- 3.4 Feature Engineering . . . 22
- 3.5 Model Evaluation . . . 25
  - 3.5.1 Evaluating Model Quality . . . 25
  - 3.5.2 Cross Validation with Contiguous Segments . . . 25
  - 3.5.3 Comparing to a Naive Predictor . . . 26
  - 3.5.4 Comparison to an Expert . . . 27
- 3.6 Machine Learning Models . . . 28
  - 3.6.1 Standard Machine Learning Algorithms and Parameters 28
  - 3.6.2 Modeling with a confidence weighted GPR Ensemble, M^w_{gpr} 30
  - 3.6.3 Incorporating Other Patient's Data with Stacking . . . 30
- 3.7 Further Investigations . . . 31
  - 3.7.1 Incorporating Patients with Similar BG Variances . . . 32
  - 3.7.2 Meal Specific Prediction and a Less Naive M_{avg} . . . 32
  - 3.7.3 Predicting from Records without Injection or Ingestion 33
  - 3.7.4 Performance on a 3-way Classification Variant . . . 33

4 Experimental Results **35**
- 4.1 Cross Validation Results . . . 35
- 4.2 Results from our Further Investigations . . . 41
- 4.3 Comparison to an Expert . . . 43

5 Discussion and Conclusions **45**
5.1 Discussion 45
5.2 Conclusions 46
5.3 Directions for Future Work 47

Bibliography **49**

A Complete Description of Feature **53**

B Table of EP Record Proportions **55**

C Tables Corresponding To Heatmaps **56**

List of Tables

3.1 **Summary of Demographics** 17
3.2 **Demographics for Type I diabetes patients.** 18
3.3 **Description of Original Features, and some Computed Features, used in this Study** 21
3.4 **Example of Data, over a single day, from Patient 16** . . 22
3.5 **Datasets Generated from Different Preprocessing Steps.** 24
3.6 **Descriptions of the Different Learners Used.** 28

4.1 **Average Errors using M^w_{gpr} on Dataset D2.** 42
4.2 **Comparison of an Expert against our model across 6 patients (L_1 Error)** 44
4.3 **Comparison of an Expert against our model across 6 patients (Relative L_1 Error)** 44

A.1 **Description of Original and Processed Features used in this Study** 54
A.2 **Example of Processed Data, over a single day, from Patient 16 (Variant D1)** 54

B.1 **Proportion of Data Adhering to the Expert's Prediction Criteria** 55

C.1 **Losses Corresponding to Figure 4.1.** 56
C.2 **Percentage Improvements Corresponding to Figure 4.2.** 57
C.3 **Losses Corresponding to Figure 4.3.** 58
C.4 **Percentage Improvements Corresponding to Figure 4.4.** 59

List of Figures

3.1 **Spline of Insulin on Board over Time.** 19
3.2 **Records Meeting the Expert's Prediction Criteria.** Patients are sorted by descending numbers of records. See Appendix B for further details. 23
3.3 **Target err_{rLl} Loss Function.** This function emphasizes loss when the model fails to predict hypoglycemic events. Here, "True BG" refers to BG_{i+1} and "Predicted BG" refers to $\widehat{BG_{i+1}} = M(\, x_i,\, \Delta t_{i+1}\,)$. Note that the error is large when the true BG is very small, but the predicted $\widehat{BG}$ is relatively large. 26
3.4 **Illustration of 5-Fold CV with Contiguous Segments.** In each CV iteration training is done on the blue segments and testing on the green segment. 27
3.5 **How we Implement Stacking.** 31

4.1 **Average L1 Loss: Datasets vs. Models.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square represents the cross-validation L1 error, micro-averaged over patients. 35
4.2 **Percent Improvement in Average L1 Loss for Models vs. Baseline.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square corresponds with the percent change between the corresponding result in Figure 4.1 and the performance of M_{ave}. 36
4.3 **Average Relative L1 Loss: Datasets vs. Models.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square represents the cross-validation relative L1 error, micro-averaged over patients. 36
4.4 **Percent Improvement in Average Relative L1 Loss for Models vs. Baseline.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square corresponds with the percent change between the corresponding result in Figure 4.3 and the performance of M_{ave}. 37
4.5 **Model M^w_{gpr}: average err_{L1} as a function of BG variance, for all 51 patients.** 38
4.6 **Model M^w_{gpr}: average err_{L1} as a function of the # of diabetes diary entries for a patient, for all 51 patients.** 39
4.7 **Model M^w_{gpr}: Our GPR ensemble's predictions on data from patient 16.** 40

Chapter 1

Introduction

The goal of this work is to determine if the data produced by Type 1 diabetes (T1D) patients, as is typically collected in a diabetes diary, can be used with machine learning algorithms to produce accurate blood glucose prediction models.

1.1 Motivation and Context

Patients suffering from Type I diabetes (T1D) are unable to produce insulin, meaning their bodies cannot properly regulate their blood glucose (BG) [11] – *i.e.*, keep their BG between four to eight mmol/L [21]. As a result, T1D is a serious chronic condition that can lead to microvascular, macrovascular, neurolgical and metabolic complications [11, 21].

To manage their diabetes, patients give themselves periodic injections of insulin as directed by their health care team. Injecting too much insulin may induce hypoglycemia (BG less than four mmol/L), which can be dangerous and possibly cause a coma. However, patients should not attempt to avoid hypoglycemia by consistently injecting too little insulin; this will result in hyperglycemia (BG greater than eight mmol/L), which may give rise to chronic complications such as blindness, kidney failure, nerve damage and circulatory problems [11, 21]. A patient's BG at a given time will depend on many factors, such as past carbohydrate intake, the amount of bolus/basal insulin injected, exercise, and stress [21].

In general, diabetes patients try to properly maintain their BG in a normal

range. This is challenging because tight glycemic control using bolus insulin injections is associated with an increased risk of having hypoglycemic events [11]. This challenge has led to attempts to create closed-loop systems and the use of computational techniques that assist in controlling patients' BG levels [5]. An extreme example of this is the effort to create an "artificial pancreas" which explicitly integrates automatic monitoring with automatic administration of insulin [23]. Another perspective on fully automated diabetes management views the BG control problem as two sequential subproblems:

1. "modeling": learning an accurate BG *prediction model* that, for example, predicts the BG level at lunch given a description of the subject throughout breakfast (including her previous BG values, carbohydrate intake, etc., from earlier meals), as well as the amount of insulin injected at breakfast.

2. "controlling": given the current information (at breakfast), consider the effects of injecting various possible amounts of insulin – *e.g.*, {1 unit, 1.5 units, 2 units, ...} – for each, use the learned model to predict the BG value at lunch. One can then inject the amount that is predicted to lead to the best lunch time BG-value[1].

This paper focuses on the first subtask: developing a BG prediction system. We use machine learning techniques to learn models that can be used to estimate an individual's future BG using covariates that describe the current patient. This work is an extensive effort to build an accurate BG prediction model, which involved exploring 312 different model and preprocessing variant combinations. For the training of our models, we used a dataset consisting of 30 221 data points collected from 51 patients. Each data point included the information typically collected: the time of day, the patient's current BG, the carbohydrate about to be consumed and the anticipated exercise. Before experimentation, we posited that accurate blood glucose prediction models

[1]Of course, this assumes that the breakfast decision affects only lunch, then the lunch decision will only affect dinner, etc. – which does not consider the longer-range effects of actions; see Bastani [5].

could be developed using this type of data. However, through the course of the investigation it was found that this is likely not the case.

1.1.1 Our Specific Task

In general, a model M will predict the blood glucose $\widehat{BG_{i+1}} = M(\, x_i,\, \Delta t_{i+1}\,)$ at the next time point (Δt_{i+1} minutes into the future), based on information currently known about this patient[2]:

$$x_i \quad = \quad [time_i,\ BG_i,\ bolus_i,\ basal_i,\ ExV_i,\ PV_i,\ IOB_i,\ \ldots] \qquad (1.1)$$

(Think of predicting the blood glucose at 12pm lunch on Tuesday, given information collected up-until 8am breakfast on Tuesday. Note that this could be only the Tuesday breakfast information, or it could include other earlier information – *e.g.*, the ellipses in Equation 1.1 might contain information about events from yesterday, or last week). See also Table 3.3.

1.2 Contributions

Below we list the main contributions of this work:

1. To our knowledge, this study examines the largest multi-year dataset of diabetes diary records, collected from Type 1 diabetes patients, used for modeling future BG.

2. We provide a comprehensive study of this data, considering 312 combinations of learning algorithm and type of data to determine if machine learning can be used to create an accurate blood glucose prediction model.

3. Our results demonstrate that it is difficult for a machine learned model to perform better than a naive baseline model (in this case, predicting a patient's average BG).

[2]Here we abstract some issues; see Appendix A for details.

4. We compare our best model to a human expert and show that both perform similarly to one another on this *BG* prediction task, while both outperforming a naive baseline model.

1.3 Outline

The remaining chapters of this thesis are as follows: Chapter 2 discusses the previous *BG* modeling literature and describes the different machine learning algorithms used in this work. Chapter 3 describes the dataset that was used, how it was processed, and how different models were trained and evaluated on the data. Chapter 4 shows the results of the different experiments conducted in Chapter 3, and Chapter 5 discusses and summarizes these results. Appendix A provides a more complete view of features we obtain after data preprocessing. Appendix B lists the number of records for each patient that meet our EP criteria, described in Section 3.3. Finally, Appendix C provides the detailed results corresponding to the heatmaps shown in Chapter 4.

Chapter 2

Background

2.1 Type I Diabetes

Type 1 diabetes is a metabolic disorder where β cells, the insulin producing cells of the pancreas, have been destroyed and some insulin resistance is present [17]. Without β cells producing sufficient amounts of insulin, a patient's body is unable to properly regulate blood glucose. As previously stated, this leads to hyperglycemia which must be managed with regular injections of insulin. While this disease can have a late onset, it is typically found in younger individuals. These individuals are often genetically predisposed to having an autoimmune response (Type 1 A) that targets their β cells [17, 6]. Looking at levels of glycated hemoglobin (HbA_{1c}) can help determine if an individual is consistently hyperglycemic and therefore diabetic [17].

Patients with T1D will follow a regimen for monitoring and controlling their blood glucose. Typically, this will involve three daily injections of bolus (short acting) insulin, at least one daily injection of basal (long acting) insulin, and four daily measurements of blood glucose (one before each meal and one before bed) [17]. Ideally, patients will record all of their measurements and injections in a notebook (diabetes diary) thereby creating a recorded history of their blood glucose changes in response to factors such as bolus (rapid acting) insulin injections, basal (long acting) insulin injections, carbohydrate ingestion and physical activity. Maintaining this history allows medical professionals to determine the parameters governing how much insulin patients will inject in response to the performance of the regimen. However, adjustments made by

professionals may only happen a few times every year [5]. Note that strict glycemic control is achieved when patients reach a pre-meal blood glucose of four to seven mmol/L and a post-meal blood glucose of four to ten mmol/L [17].

2.2 Towards Automated Type I Diabetes Management

2.2.1 Limitations of Prior Diabetes Modeling Studies

One of the issues limiting Type 1 diabetes modeling research is the lack of available large datasets. Previous studies have been based on data from small numbers of subjects and/or data collected over a short time period. For example, several studies have been based on data from a single patient where records were only collected for fewer than 100 days [39, 21, 4, 44]. These studies are limited because the predictive quality of models trained on different patients varied greatly (shown later in Figure 4.5). Other studies included more patients (12–15) but only had 3–22 days worth of data [15, 2]. Another study used three patients with two years of data [24]. In these last three cases, datasets either had short histories for their patients, or they had a small number of patients in total. In contrast, our work uses a larger number of patients who had up to two years worth of data. Records were collected multiple times each day.

There exist large datasets of type 2 diabetes patients – *e.g.*, Quinn *et al.* [31] who measured glycated hemogloben changes in data collected data from 163 patients over the course of a year. However, studies that model type-2 diabetes [9, 38] should not be directly compared to those that model type-1 diabetes because there are significant differences between these diabetes types. In particular, there is less variance in the blood glucose readings over time for type-2 patients than there is for type-1 patients, making it easier to model.

While we focus on predicting BG values *many hours later*, some studies instead attempt to predict the occurrence of hypoglycemic events and only within a short window (*e.g.*, 30 to 120 minutes) [8, 14, 15, 27, 30]. Even though this might help to protect patients from a very serious situation, it

is lacking in several ways. First, such fine-grain measurements are often not practically obtainable outside of a study setting and without using a continuous glucose monitoring (CGM) device that provides measurements every 5 minutes. Second, these short-term predictions are not adequate for spanning the time between meals. Third, the goal of building a diabetes control system is better served with a more expressive model, as opposed to one that can only provide binary classifications – hypoglycemic or not. Note that these models that only make binary predictions do not provide useful feedback for situations where patients are hyperglycemic. In our work, we try to model blood glucose dynamics (including both hyperglycemia and hypoglycemia) using only the standard records collected at meal times. While this makes our task more challenging, we do this because it involves only the data that medical professionals most often encounter in practice.

2.2.2 Additional Background

In this section we describe several techniques and approaches within the diabetes modeling literature that we used.

We considered neural network models, as did Pappada *et al.* [26]. Using a held-out patient from a dataset of 18 T1D patients, Pappada *et al.* was able to achieve an overall "score" of 0.067, 0.089, 0.117, 0.145, 0.166, 0.189 when using predictive windows of 50, 75, 100, 120, 150 and 180 minutes in the future. This score is a version of the rL1 measure (defined in Equation 3.2) that we used in this study. While Pappada *et al.* are able to achieve a score of 0.189 using 180 minute predictive windows, it is important to note that their result is based on the test data of a single patient (other patients may be more difficult to predict) and that the dataset used in our study required that predictions, on average, be made 310.6 minutes into the future (593 minutes on average for overnight predictions and 236 minutes on average otherwise). Since Pappada *et al.* showed that larger predictive windows decrease the accuracy of their models, we expect that our data should be more difficult to model well. Also, note that the participants from their study provided only 3 to 9 days of data with continuous glucose monitoring, whereas our data were collected over a

period of months to years.

Our work resembles previous works [41, 13] that use Gaussian Process Regression (GPR; see Section 2.3.1) as one approach for modeling diabetes. In particular, Duke [13] used GPR to learn models of individual patients that could be used to aid in cross-patient prediction. We similarly explore some transfer learning techniques with GPR, along with ensembles of learners and various other machine learning algorithms.

Prior works have also addressed the data and blood glucose modeling problem that we seek to address in this work [21, 4, 44]. These latter two evaluate their results using normalized blood glucose values; since they are not in units of mmol/L, they cannot be directly interpreted. This also means that we cannot compare our results to theirs. However, in our results, we do evaluate the performance of a model that is similar to the Gaussian Wavelet Neural Network used in Zainuddin *et al.* [44].

2.3 Machine Learning Algorithms

2.3.1 Supervised Learning Algorithms

This work exists within the standard supervised machine learning framework: We start with a labeled dataset associated with a single patient $D[\text{patient\#j}] = \{\, [[x_i, \Delta t_{i+1}],\ BG_{i+1}] \,\}_i$, using the x_i shown in Equation 1.1. Here, our task is to predict the blood glucose value BG_{i+i} after time Δt_{i+1} has elapsed given the features x_i. Note that we augment the features x_i, collected at the starting time i, with the time interval Δt_{i+1}. We do this because the prediction depends critically on both the x_i features and the time when the predictions are made. For example, given information about the patient at noon, her BG at 1pm will be different from her BG at 4pm, etc.

A machine learning algorithm L, in general, takes a dataset D of the form described above, and produces a predictor $M_{L,D}$; this predictor will then be applied to a new instance $[x_j, \Delta t_{j+1}]$ and produce an estimate of the value BG_{j+1}. Note that in this work, the majority of our learners train exclusively on a single patient's data.

Motivation for the Supervised Algorithms Selected

The main motivation for our selection of supervised algorithms was to have representative algorithms from a broad range of supervised learning categories.

We selected Ridge Regression because it is an example of a simple linear model with the most basic regularization. We consider this algorithm to be representative of linear models in terms of its performance. We also considered Random Forests Regression because it is a well-known ensemble method built upon decision trees. We assume it to be representative of rule based learning as well as ensemble methods. We chose the third algorithm, SVR, for multiple reasons. First, SVR can be kernelized, meaning it can learn non-linear relationships. Therefore, we consider SVR to be representative of kernel methods. Second, Support Vector Machine algorithms are very popular in the ML literature due to properties such as having a single local/global optimum. The fourth class of algorithms used was Neural Networks, including a Feed-Forward Network and a WNN. These were chosen because of their prevalence in the blood glucose modeling literature. In particular, we chose the WNN in order to evaluate the model used by Zainuddin *et al.* [44]. Finally, we chose Gaussian Process Regression because it provides a full posterior distribution for each prediction and it is representative of models that use stochastic processes.

K-Nearest Neighbors (KNN)

For a data point $x_i \in \mathbb{R}^n$ with an unknown continuous label $y_i \in \mathbb{R}$, regression with the K-Nearest Neighbors algorithm can be viewed as an averaging of the K nearest local $\{x_j, y_j\}$ near x_i where the value K (e.g. $K = 5$) determines the number of closest points to be used [22]. Closeness is determined by a distance metric $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ (*e.g.*, Euclidean distance).

Variants of this basic implementation use weights (w_j, given $\sum_{j \in \mathcal{N}_K} w_j = 1$ and $w_j \geq 0$) which can accentuate the contribution of closer points relative to farther points within the set of the K closest points ($\mathcal{N}_K$).

$$y_i = \sum_{j \in \mathcal{N}_K} w_j y_j \tag{2.1}$$

In this work we use the implementation provided by *scikit-learn* with the "uniform" and "distance" options[1].

Support Vector Regression (SVR)

The original SVR description (with slack variables ξ_i and ξ_i^*) by Vapnik *et al.* is an optimization of the form [37]:

$$\begin{aligned} min_{w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle w_i, x_i\rangle - b \leq \epsilon + \xi_i, \\ & \langle w_i, x_i\rangle + b - y_i \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0. \end{aligned} \tag{2.2}$$

where C is a tunable parameter that adjusts the accrued slack variable penalties. Here, the objective is to find a w and a b, and therefore a regression line such that all points lie within an ϵ margin or as close as possible to the ϵ margin boundary.

The previously described optimization can also be reformulated to include non-linear kernels such as the radial basis function kernel, so as to model non-linear relationships. To do this we first reformulate the optimization above as a Lagrange function (known as the primal) [37].

$$\begin{aligned} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ - \sum_{i=1}^{l}\alpha_i(\epsilon + \xi_i - y_i + \langle w_i, x_i\rangle + b) \\ - \sum_{i=1}^{l}\alpha_i^*(\epsilon + \xi_i^* + y_i - \langle w_i, x_i\rangle - b) \end{aligned} \tag{2.3}$$

where

$$\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$$

[1] `http://scikit-learn.org/stable/modules/neighbors.html`

The introduced variables α_i, α_i^*, η_i and η_i^* are the Lagrange multipliers. Once we have this primal Lagrange function, we can obtain the dual Lagrange function by taking the partial derivatives of the primal (w.r.t. w, b, ξ_i and ξ_i^*) and solving for the values of these variables that minimize the primal. These can then be substituted into the primal to obtain the dual [37].

$$\begin{aligned} \text{max} \quad & \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle - \epsilon\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \\ \text{subject to} \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0, \\ & \alpha_i, \alpha_i^* \in [0, C]. \end{aligned} \tag{2.4}$$

Finally, now that we have defined our problem in terms of inner products of the data ($\langle x_i, x_j\rangle$), we can learn non-linear relationships using transformed data ($\Phi(x_i)$). We do not need to explicitly transform the data because an appropriate kernel ($\kappa(x_i, x_j) = \langle\Phi(x_i), \Phi(x_j)\rangle$) can be applied to the inner product of the data instances. Note that in this work, we use both radial basis function (RBF) kernels as well as simple linear kernels.

Ridge Regression

Ridge regression is an extension of the ordinary least squares algorithm that includes a regularized term for the model weights. Therefore, the training problem for a dataset $\mathbf{X}$ with labels $\mathbf{y}$ becomes:

$$w^*, b^* = \underset{w,b}{\text{argmin}}\, ||\mathbf{y} - b\mathbb{1} - \mathbf{X}\mathbf{w}||^2 + \alpha||\mathbf{w}||^2 \tag{2.5}$$

where w^* is the learned set of weights for the linear model, b^* is the learned bias term and α is a hyper-parameter that adjusts the relative contribution of the weight magnitudes [33].

Artificial/Wavelet Neural Networks (ANNs/WNNs)

We primarily use a type of artificial neural network (ANN) that is known as a feed-forward neural network, which is an acyclic ANN with internal weights

that capture the mapping from inputs to desired outputs. These types of networks are typically composed of three distinct types of layers: an input layer, hidden layers and an output layer. At training time, a feature vector is provided to the input layers to produce a predicted value at the output layer, which is compared to a known true value. With this, the subsequent error gradient is used to adjust the internal weights [18].

The architecture we use includes one hidden layer, a single output node and rectified linear units as activation functions for each neuron. The implementation of this network is done through Keras[2].

Wavelet neural networks (WNNs) differ from other feed-forward neural networks in that they use wavelet activation functions that are derived from a mother wavelet function [44]. In a three-layer network, the value of an output neuron ($\hat{y}$) for the i^{th} data instance ($\mathbf{x}_i$) is $\hat{y}(\mathbf{x}_i) = \Sigma_{j=1}^{k} w_j \Psi_j(||d_j(\mathbf{x}_i - t_j)||) + b$, where b is the bias term, w_j is the weight for the j^{th} hidden neuron, Ψ is the mother wavelet function and Ψ_j is the activation wavelet for the j^{th} hidden neuron (which is parameterized with a dilation constant d_j and a translation constant t_j).

Zainuddin *et al.* suggest that WNNs are better suited for BG modeling, as compared to traditional neural networks, because the integration of wavelets allows the resulting model to better match the fluctuations present in BG time series data. In particular, they state that the shape of the Gaussian wavelet, having antisymmetry and a steep gradient at its center, correlates strongly with the irregular, saw-tooth shape of the data [44].

Random Forest (RF) Regression

In Random Forest Regression, an ensemble of regression trees (decision trees with real valued output) are used to collectively determine the label associated with input features, defined as the averaging of each tree's prediction. Here, bootstrapped samples of the data are used to build a set of random trees in which a random subset of features are used for each tree [7]. Importantly, Random Forests have the advantageous property of being more resistant to

[2] `https://keras.io/`

overfitting than individual decision trees.

Gaussian Process Regression

Gaussian Process Regression (GPR) is regression technique that uses Gaussian Processes. A Gaussian Process is defined as a set of random variables in which any finite subset of these variables has a joint Gaussian distribution [32]. Specifically, for the real process $f(\mathbf{x})$ with mean function $\mu(\mathbf{x}) = E[f(\mathbf{x})]$ and covariance function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = E[(f(\mathbf{x}_i) - \mu(\mathbf{x}_i))(f(\mathbf{x}_j) - \mu(\mathbf{x}_j))]$, where $\mathbf{x}, \mathbf{x}_i, \mathbf{x}_j \in X$ (X is our covariate space), we define a Gaussian process entirely in terms of its mean function and covariance function: $f(\mathbf{x}_i) \sim GP(\mu(\mathbf{x}_i), \kappa(\mathbf{x}_i, \mathbf{x}_j))$ [32].

For a new data instance $\mathbf{x}_* \in X$ we can calculate

$$\begin{aligned} f(\mathbf{x}_*) \,|\, \mathbf{x}_*, X_{tr}, \mathbf{y}_{tr} &\sim N(\mu_*, \sigma_*^2) \\ \mu_* &= \mathbf{k}_*^\top (K + \sigma^2 I)^{-1} \mathbf{y}_{tr} \\ \sigma_* &= \Phi(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (K + \sigma^2 I)^{-1} \mathbf{k}_* \end{aligned} \tag{2.6}$$

where $\Phi(\cdot, \cdot)$ is a kernel, X_{tr} is our training data, $\mathbf{y}_{tr}$ is our training labels (with n instances), $\mathbf{k}_* = [\Phi(\mathbf{x}_*, \mathbf{x}_1), \Phi(\mathbf{x}_*, \mathbf{x}_2), ..., \Phi(\mathbf{x}_*, \mathbf{x}_n)]^\top$ and $K_{ij} = \Phi(\mathbf{x}_i, \mathbf{x}_j)$ [42].

2.3.2 Unsupervised Learning Algorithms

In this section, we describe the two unsupervised learning algorithms that were used with supervised learning algorithms in order to generate some of the different *BG* prediction models. We use PCA as a feature preprocessing option to reduce the dimensionality, and K-means clustering as a method of grouping similar patients, to allow our algorithms to just learn a model for one patient, based only on similar data from similar patients.

Principal Component Analysis (PCA)

PCA is a linear transformation representing a change of basis such that data are projected onto a new orthonormal basis. These new basis vectors (components) are the directions of greatest variance in the data and are obtained

in order such that the direction of the first vector has the most variance, the direction of the second vector has the next greatest variance, etc. Generally, the purpose of this transformation is to reduce redundancy and noise in the data [36], which is achieved by isolating those principal components that capture the most variance.

PCA typically involves the following steps. For a 0 centered dataset $\mathbf{X}_{n\times m}$ with n random variables and m observations, the covariance matrix is calculated $\Sigma = \frac{1}{m}XX^T$, where the covariance of the i^{th} and j^{th} random variables is $\Sigma_{i,j} = E[(X_i - E[X_i])(X_j - E[X_j])]$. The eigenvalues ($\lambda$) and eigenvectors ($v$) of Σ, which satisfy $\Sigma v = \lambda v$, are then found using the equalities $det(\Sigma - \lambda I) = 0$ and $(\Sigma - \lambda I)v = 0$ respectively. Here, $det(\Sigma - \lambda I)$ produces the characteristic polynomial with eigenvalues as roots. Since the covariance matrix is symmetric, the resulting eigenvectors will form a new orthonormal basis for the data and the absolute magnitude of the eigenvalues will determine the most significant components.

K-means Clustering

One of the oldest problems in the field of computational geometry is that of partitioning d dimensional points in $\mathbb{R}^d$ into appropriate groups (clusters) where members of a cluster are related to one another [3]. To achieve this goal, we use the well-known K-means algorithm. The generic K-means variant can be described in four steps [3] with a given input parameter $K \in \mathbb{N}^+$. First, K initial points (centers) are arbitrarily selected in $\mathbb{R}^d$ space. Second, all points in $\mathbb{R}^d$ space are assigned to the closest center. Third, centers are recalculated to be the center of each cluster determined in the second step. Finally, steps two and three are repeated until there is no longer any change in the value of the centers calculated in step two. This algorithm finally returns the assignment, mapping each data point to a cluster.

To assess the quality of clusters generated from any clustering method such as K-means, we use a visualization technique known as the silhouette plot [34].

The silhouette plot uses

$$s(i) \quad = \quad \frac{b(i) - a(i)}{\max\{\, a(i),\, b(i)\,\}} \tag{2.7}$$

where $s(i)$ is the silhouette of the i^{th} data point, $a(i)$ is the average dissimilarity (based on a given distance metric) between the i^{th} data point and the other members of its cluster, and $b(i)$ is the minimum average dissimilarity between the i^{th} data point and the points of another cluster in the partitioned space. We can use equation 2.7 to obtain the average silhouette width from all the silhouette values for each cluster to determine the cluster's quality individually. Alternatively, we can find the average silhouette width across all clusters to determine the quality of a particular K partition of the data space using K-means clustering.

Chapter 3

Methods: Machine Learning Applied to Type I Diabetes

3.1 Data

This study used data from 51 Type I diabetes patients that was collected by using the "Intelligent Diabetes Management" (IDM) software [1](described in Ryan *et al.* [35]). This data included patients who participated in Ryan *et al.*'s study, as well as additional patients who began using the IDM software after the completion of the study (up until December 2016). The participants gave their informed written consent, and the Research Ethics Board of the University of Alberta approved the collection and analysis of the data. For further details regarding patient participation, see Ryan *et al.* [35]. However, some of the participants only used the system a few times. We therefore only included patients who made at least 100 diabetes diary entries with the system – *i.e.*, produced at least 100 "sufficient" records. This led to a dataset consisting of 16 pump users and 34 + 1[2] non-pump users. Table 3.1 provides summary statistics for our data and Table 3.2 presents the complete demographic information for all patients included in the study.

This study also includes Patient #16 from Table 3.2, who has by far the most records of any patient in our dataset. Indeed, it is unusual for any patient to consistently produce diabetes entries over the course of many years. Because of her large number of records, we use part of her dataset as our

[1] `https://idm.ualberta.ca/`

[2] See Table 3.1: One patient did not indicate their pump status

Table 3.1: **Summary of Demographics**

	Patients	Age	Height	Weight	Sex	Pump Users
# of Values Recorded	51	40	10	42	40 (7 ♂ / 33 ♀)	16/51*
Average	N/A	42 ± 12	164 ± 9 cm	74 ± 14 kg	N/A	N/A

* One patient did not report using a pump and did not have any recorded pump values. This individual was therefore treated as a non-pump patient.

hyper-parameter tuning (validation) dataset, as well as for visualization (such as Figure 4.7 later).

Each record i corresponds to an entry in a patient's "diabetes diary", which includes the meal associated with the record $meal_i$, a time stamp ($date_i$ and $time_i$), the blood glucose value BG_i, the grams of carbohydrates consumed CHO_i, the units of bolus (rapid acting) insulin injected $bolus_i$, and basal (long acting) insulin injected $basal_i$. The patients also entered the anticipated level of exercise using the non-numeric values {"less than normal", "normal", "active", "very active"}. Following advice from our expert diabetologist[3], we converted these into numeric values for use by standard learning algorithm: 2, 4, 7 and 10.

As stated earlier, 16 of the patients in this study used insulin pumps. These pumps work by directly infusing insulin from a reservoir, via a catheter, into a patient's skin at a basal rate. Moreover, they are also used to inject larger amounts of bolus insulin, when a patient ingests carbohydrates (as a patient would with a syringe)[4]. If the patient was using an insulin pump, the basal pump value PV_i (in units/hour) was included into each record. The insulin pump settings work by partitioning the 24h clock into intervals, where a particular rate of insulin is set to be delivered during each interval. A PV_i for any specific record is then found by obtaining the corresponding rate of insulin delivery for the interval containing the record's time stamp.

We also computed two other features: Δt_i, which is the elapsed time since

[3]Dr. Edmond A. Ryan

[4]`http://www.diabetes.org/living-with-diabetes/treatment-and-care/medication/insulin/how-do-insulin-pumps-work.html`

Table 3.2: **Demographics for Type I diabetes patients.**

Patient	Age	Height (cm)	Weight (kg)	Gender	Pregnant	Pump	Record
1	N/A	N/A	N/A	N/A	N/A	0	203
2	N/A	N/A	N/A	N/A	N/A	1	168
3	N/A	N/A	N/A	N/A	N/A	0	1008
4	N/A	N/A	N/A	N/A	N/A	N/A	170
5	53	162.5	67.2	Female	0	0	340
6	36	183	93	Male	0	0	324
7	33	175.9	72.1	N/A	0	1	178
8	42	170.2	79.3	Female	0	0	241
9	41	160	71.2	Female	0	0	722
10	56	161	83.7	Female	0	1	794
11	53	161.5	70.6	Female	0	0	424
12	59	155.5	82.7	Female	0	1	1666
13	25	156.6	45.7	Female	0	0	180
14	42	155	74.2	Female	0	0	1217
15	24	N/A	62	Female	0	0	827
16	61	N/A	58	Female	0	1	4722
17	45	N/A	77	Female	0	1	791
18	39	N/A	47	Female	0	1	192
19	31	N/A	80	Female	0	0	151
20	50	N/A	80	Female	0	1	645
21	63	N/A	80	Male	0	1	893
22	66	N/A	59	Female	0	1	783
23	62	N/A	56	Female	0	1	114
24	26	N/A	80	Male	0	1	973
25	53	N/A	85	Female	0	0	626
26	49	N/A	68	Female	0	0	317
27	46	N/A	84	Female	0	0	385
28	40	N/A	55	Female	0	1	217
29	N/A	N/A	69	Female	0	0	120
30	36	N/A	62	Female	0	0	291
31	37	N/A	70	Male	0	0	189
32	41	N/A	75	Male	0	0	547
33	44	N/A	80	Female	0	1	331
34	48	N/A	84	Female	0	0	204
35	31	N/A	117	Female	0	0	376
36	40	N/A	95	Female	0	1	352
37	50	N/A	60	N/A	0	0	167
38	21	N/A	63	Female	0	0	537
39	20	N/A	86	Male	0	0	792
40	28	N/A	95	Female	0	0	522
41	34	N/A	66	Female	0	0	1273
42	44	N/A	100	Male	0	0	400
43	26	N/A	64	Female	0	0	312
44	59	N/A	63.6	Female	0	1	488
45	33	N/A	75.2	Female	1	0	318
46	N/A	N/A	N/A	N/A	N/A	0	603
47	N/A	N/A	N/A	N/A	N/A	0	275
48	N/A	N/A	N/A	N/A	N/A	0	177
49	N/A	N/A	N/A	N/A	N/A	0	145
50	N/A	N/A	73	Female	N/A	0	2088
51	N/A	N/A	N/A	N/A	N/A	0	443

Many entries are missing in this table as this data was collected voluntarily. Note patient 45 was pregnant at the time of this study.

the previous record[5] and "Insulin on Board" IOB_i, which captures the effect of any insulin remaining in a diabetic's system from previous injections [1]. This was based on the following pairs of elapsed time and percentage of post injection insulin remaining obtained from our diabetes expert: (1.66 hours, 78%), (2.5 hours, 48%), (3.33 hours, 27%), (4.15 hours, 12%), (5 hours, 3%). A simple spline was used to interpolate these values and is visualized in Figure 3.1. Table 3.3 formalizes these features and Table 3.4 provides example data.

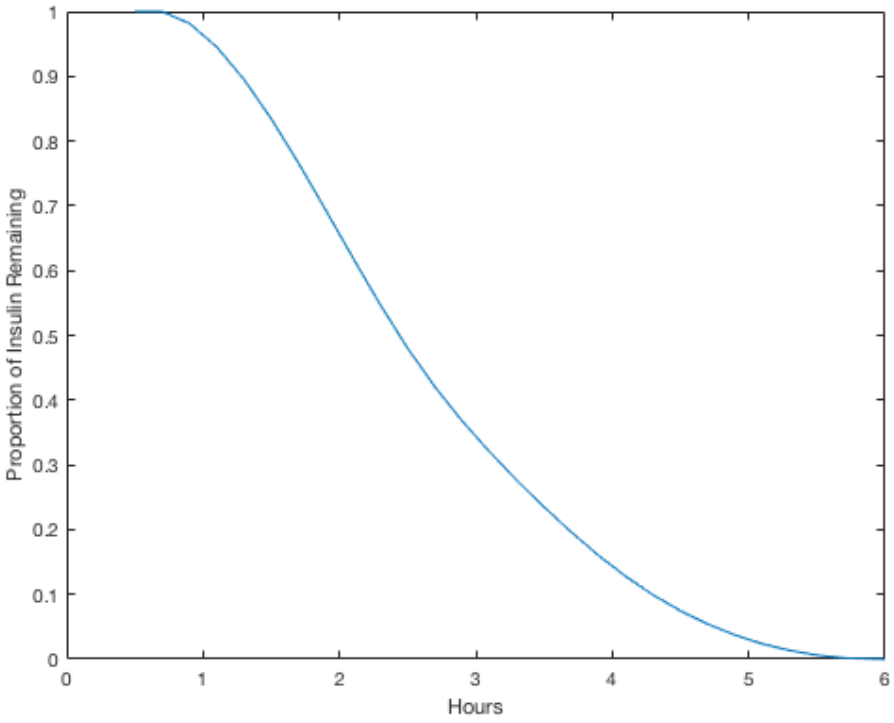

Figure 3.1: **Spline of Insulin on Board over Time.**

3.2 Data Preprocessing

The first step in processing the data was dealing with the missing or erroneous values. We discarded any record that did not have an associated BG value (572 records). This was necessary as we cannot evaluate a model on records

[5]Actually, Δt_i is based on previous *bolus* or CHO values; see Appendix A.

where we do not have a ground truth for BG. We also discarded any records that had missing dates (6 records) because these timestamps are integral to deriving features from the data. Second, we changed any BG value less than 1 mmol/L (8 records) to 1 mmol/L. We did this based on the assumption that the glucose meter reported an erroneous BG value because it is not likely that a patient would be healthy enough to report or survive such low values. We then log-transformed these blood glucose values for all our predictors, anticipating this log-linear model might have better performance. This means that after the model makes a prediction, we must use simple exponentiation to transform that prediction back into the original interpretable units.

To address missing bolus insulin and carbohydrate values CHO, we imputed average values into the missing entry (variants for this step are described in Section 3.4). This was done on a per-subject, per-meal basis – that is, we imputed an individual's average value for a particular meal. For example, say a specific patient injected, on average, 3 units of bolus insulin before breakfast. Whenever she does not enter the before-breakfast insulin, we replace that missing value with "3 units". When dealing with missing exercise values, we imputed the "normal" value instead of an average value as is done with insulin and carbohydrates. For missing basal insulin values, we always imputed a particular value – here 0 – that never appeared in any real situation. This allows a learner to distinguish when basal insulin was recorded and when it was not. After this preprocessing, we computed the auxiliary features (Δt_i, IOB_i) from the improved data. We describe the complete set of features in Table A.1, and we show example records as columns in Table A.2; see Appendix A.

3.3 Expert Feedback on Deciding When to Predict

As our data was collected voluntarily from patients at their own convenience, sampling intervals are not uniform in the data and data is not available for every meal. This becomes problematic when modeling the data because blood glucose values become more difficult to predict as more time is allowed to elapse

Table 3.3: **Description of Original Features, and some Computed Features, used in this Study**

$meal_i$	The time of day: { Before Breakfast, After Breakfast, Before lunch, After Lunch, Before Supper, After Supper, Before Bed, During the Night}
$date_i$	The date as year-month-day
$time_i$	The time as hour:minute:second
BG_i	The BG value at the current time (mmol/L)
CHO_i	The amount of carbohydrates ingested (grams)
$bolus_i$	The amount of insulin injected (units)
$basal_i$	The units of background insulin injected
ExV_i	Numeric encoding of exercise value: $\{2, 4, 7, 10\}$
PV_i	Pump Value: The rate at which the insulin pump is infusing (units/hour). This is always 0 if the patient does not have a pump
Δt_i	The elapsed time since last record
IOB_i	Insulin on Board: Estimated residual insulin from the previous injection (mmol/L)

Note this is a simplified set of features; see Table A.1 in the Appendix for the complete set of feature descriptions

between readings. To address this issue, we asked an expert diabetologist (E. A. Ryan) when an expert would feel comfortable making a prediction, given a patient's history. This discussion led to the following criteria – BG is "expert predictable" (EP) at a future time point when given the following:

1. The preceding record cannot be a hypoglycemic event[6].

2. The blood glucose reading must be present for the preceding meal. For example, to make a prediction about a patient's blood glucose value before lunch, a record detailing his/her previous breakfast must be available.

3. Six of the last eight days prior to a prediction must have records for the current meal time and the previous meal time. For example, to predict the blood glucose before lunch, six of the last eight days must have both "before lunch" and "after breakfast" entries, to help capture this "after breakfast to before lunch" transition pattern.

[6]Due to potential glucose counterregulation effects [16] and the uncertainty in BG that follows from a physiological response to hypoglycemia

Table 3.4: **Example of Data, over a single day, from Patient 16**

index i	27	28	29	30	31	32
$meal_i$	Before Breakfast	After Breakfast	Before Lunch	After Lunch	Before Dinner	After Dinner
$date_i$	2015-11-25	2015-11-25	2015-11-25	2015-11-25	2015-11-25	2015-11-25
$time_i$	08:36:00	10:19:00	12:19:00	15:35:00	18:42:00	20:11:00
BG_i	16.2	14.7	5.6	6.8	10.5	3.0
CHO_i	30.0	0	30.0	0	15.0	0
$bolus_i$	10.4	0	3.0	0	3.8	0
$basal_i$	0	0	0	0	0	0
ExV_i	4	4	4	4	4	4
PV_i	0.50	0.50	0.63	0.45	0.90	0.90
Δt_i	540	103	120	196	187	89
IOB_i	0.00	7.90	3.61	0.89	0.81	3.35

Note this is a simplified version of the data; Table A.2 in the Appendix provides the general, complete set of features.

Fig 3.2 shows the number of records from each patient that qualify as EP – the number of records for which our expert would feel comfortable making predictions.

Later, we trained and evaluated models using the entire dataset D as well as models that were trained on D and evaluated using only "EP" records, D^{EP}, that met the expert's criteria. Note that the results when testing on D^{EP} were worse when we trained on only D^{EP}. Also, the learner had access to both pump patients and non-pump patients for each test patient, regardless of whether or not that patient was a pump-patient (but note that this "pump" characteristic was a feature that the learner, and resulting classifier, could use).

3.4 Feature Engineering

Table 3.3 shows the basic features used to describe each event. Additionally, we also considered many other feature sets to see if any could lead to better performance. Some of the variants completed records that were missing entries for carbohydrates or bolus insulin, and some removed those deficit records. Others added in the day of the week as an integer feature or as a one-hot encoded feature[7], while others excluded basal insulin as a feature. A few variants included non-temporal patient characteristics: age, gender, height and

[7] `http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html`

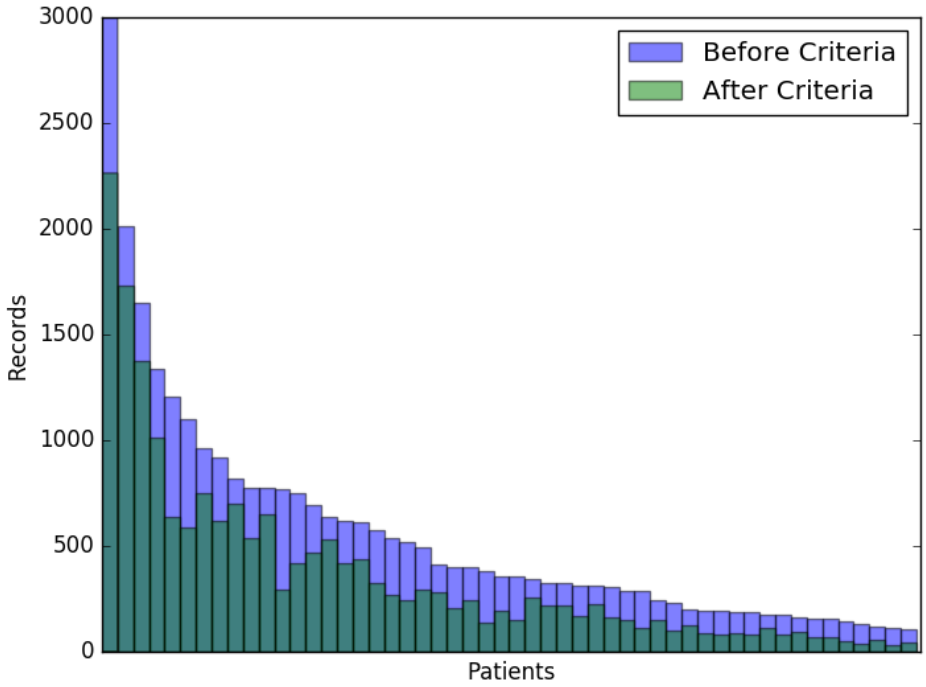

Figure 3.2: **Records Meeting the Expert's Prediction Criteria.** Patients are sorted by descending numbers of records. See Appendix B for further details.

weight. Some replaced the features with 4 principal components (obtained by principal component analysis (PCA) – see Section 2.3.2) according to what was done by Zainuddin *et al.* [44]. Finally, some variants applied the EP filter (from Section 3.3) to remove problematic records – *i.e.*, ones that (our expert believes) do not contain enough information to be reliably predicted. Note that this reduces the predictions that our system attempts. For example, we do not attempt to predict an after-breakfast BG value if there is no preceding before-breakfast reading. For each variant, we only considered the subset of the records that belonged to that variant, both for producing the model and also for estimating the quality of that model.

The "Kok Features" variant uses computed features similar to Kok [21], and subsequently used by Baghdadi *et al.* [4] and Zainuddin *et al.* [44]. Unlike Kok's data, however, we do not have stress level values in our data and were

therefore unable to incorporate that feature.

For any given dataset variant, some models have components that train on different subsets of the data. In addition, we also use models that include components that are trained on the data from all patients other than the patient under test, as well as models that involve sub-models that are each trained only on data from one meal type (*e.g.*, before breakfast).

Table 3.5 describes these 26 datasets, showing which set of modifications is applied to each.

Table 3.5: **Datasets Generated from Different Preprocessing Steps.**

	# of Subjects	Records Predicted	EP Rules	DOW Features	Basal Feature	Patient Specific Features	Kok Features	PCA Transform	Missing Carbs	Missing Bolus
D0	46	7981	1	1	1	0	0	0	Throwout	Impute Mean
D1	51	16378	1	1	1	0	0	0	Impute Mean	Impute Mean
D2	51	16378	1	7	1	1	0	0	Impute Mean	Impute Mean
D3	51	16378	1	1	1	0	0	0	Impute 0	Impute Mean
D4	51	16378	1	7	1	0	0	0	Impute Mean	Impute Mean
D5	51	16378	1	1	1	0	0	0	Impute Mean	Impute 0
D6	42	6437	1	1	1	0	0	0	Throwout	Throwout
D7	51	16378	1	1	1	0	0	0	Impute 0	Impute 0
D8	51	16378	1	0	0	0	0	0	Impute Mean	Impute Mean
D9	48	10725	1	1	1	0	0	0	Impute Mean	Throwout
D10	39	7332	1	0	0	0	1	0	N/A	N/A
D11	51	16378	1	0	0	0	0	1	Impute Mean	Impute Mean
D12	39	7332	1	0	0	0	1	1	N/A	N/A
D13	46	17100	0	1	1	0	0	0	Throwout	Impute Mean
D14	51	25445	0	1	1	0	0	0	Impute Mean	Impute Mean
D15	51	25445	0	7	1	1	0	0	Impute Mean	Impute Mean
D16	51	25445	0	1	1	0	0	0	Impute 0	Impute Mean
D17	51	25445	0	7	1	0	0	0	Impute Mean	Impute Mean
D18	51	25445	0	1	1	0	0	0	Impute Mean	Impute 0
D19	42	14747	0	1	1	0	0	0	Throwout	Throwout
D20	51	25445	0	1	1	0	0	0	Impute 0	Impute 0
D21	51	25445	0	0	0	0	0	0	Impute Mean	Impute Mean
D22	48	19494	0	1	1	0	0	0	Impute Mean	Throwout
D23	39	11963	0	0	0	0	1	0	N/A	N/A
D24	51	25445	0	0	0	0	0	1	Impute Mean	Impute Mean
D25	39	11963	0	0	0	0	1	1	N/A	N/A

Here, "Basal feature" and "Patient Specific Features" are features that were included (1) or excluded (0) from datasets. "DOW Features" indicates if the day of the week was not included (0), included (1), or included as a one hot encoded feature (7). "PCA Transform" indicates whether the data was reduced to 4 principle components. "Kok Features" means that the data was preprocessed to replicate (as best as possible) the features used in Kok's MSc thesis [21]. In the final two columns, the value "Throwout" means that these records were removed from the dataset. The "# of Subjects" column shows that some datasets did not include all (51) patients. In these cases, patients were excluded because they had too few records (under 100) after the preprocessing steps were applied to their data. The table is partitioned vertically so that datasets with the EP rules (D0 – D12) precede their corresponding datasets without EP rules (D13 – D25).

3.5 Model Evaluation

3.5.1 Evaluating Model Quality

We consider two evaluation functions: Given a model $M(\cdot)$ and a dataset $D = \{x_i\}_i$ where each x_i provides the "temporal information" shown in Equation 1.1, we consider both the "L_1-loss" (err_{L1}) and "relative L_1-loss" (err_{rLl})

$$\text{err}_{L1}(\, M, D\,) \;=\; \frac{1}{|D|-1}\sum_{i=1}^{|D|-1} \mid M(x_i,\, \Delta t_{i+1}) - BG_{i+1} \mid \tag{3.1}$$

$$\text{err}_{rL1}(\, M, D\,) \;=\; \frac{1}{|D|-1}\sum_{i=1}^{|D|-1} \frac{\mid M(x_i,\, \Delta t_{i+1}) - BG_{i+1} \mid}{BG_{i+1}} \tag{3.2}$$

where BG_{i+1} is the blood glucose associated with the next time point, occurring Δt_{i+1} minutes later. While err_{L1} is standard loss function, we show why it can be problematic in the following example: The [predicted, true] pair $[M(x_1,\, \Delta t_2),\; BG_2] \;=\; [5,\, 3]$ and $[M(x_3,\, \Delta t_4),\; BG_4] \;=\; [12,\, 10]$ both have an err_{L1} of 2 – *i.e.*, $|M(x_1, \Delta t_2) - BG_2| = 2 = |M(x_3, \Delta t_4) - BG_4|$ – but the first discrepancy is potentially much more dangerous, in terms of patient health, than the second. The err_{rLl} function, however, would correctly impose a larger penalty to the first $\frac{|M(x_1,\Delta t_2)-BG_2|}{BG_2} = \frac{2}{3}$ versus the second $\frac{|M(x_3,\Delta t_4)-BG_4|}{BG_4} = \frac{2}{12}$. See Figure 3.3 for a visualization of the err_{rLl} function.

3.5.2 Cross Validation with Contiguous Segments

Each of our learners will take the entire dataset and produce a model. The next challenge is evaluating this learned model. To evaluate the predictive quality of each learner, we use 10-fold cross validation (CV), with respect to each patient. We first partition time series history of a patient, denoted X_i, into ten contiguous segments $X_i = \bigcup_{j=1..10} X_i^j$. We then use nine of the ten segments for training in each CV round and use the remaining one segment for testing – so the first split would be $X_{i,tr}^1 = \bigcup_{j=2..10} X_i^j$ and $X_{i,te}^1 = X_i^1$. Note that while the testing partition always consists of contiguous data, the training partition will not always be completely contiguous. Figure 3.4 provides a visualization of what it means to partition time series data into contiguous segments for the

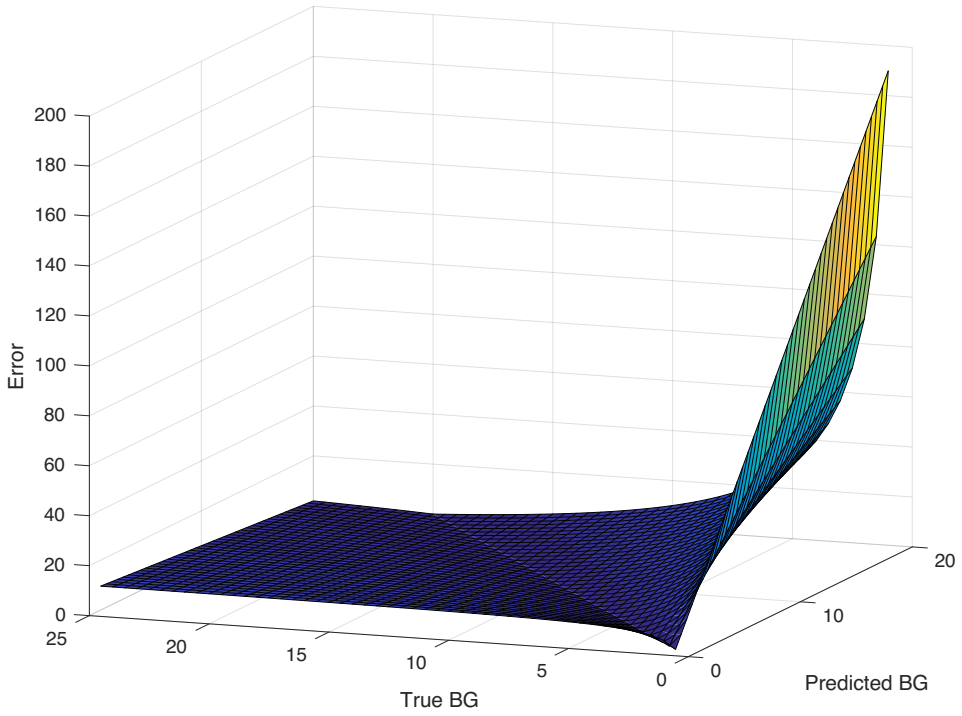

Figure 3.3: **Target err_{rLl} Loss Function.** This function emphasizes loss when the model fails to predict hypoglycemic events. Here, "True BG" refers to BG_{i+1} and "Predicted BG" refers to $\widehat{BG_{i+1}} = M(x_i, \Delta t_{i+1})$. Note that the error is large when the true BG is very small, but the predicted $\widehat{BG}$ is relatively large.

purposes of cross validation – for simplicity, here we show "5-fold CV" rather than 10.

3.5.3 Comparing to a Naive Predictor

To evaluate the quality of the models used in this work and to establish a baseline for comparison, we created a naive model (M_{avg}) that, for each patient, simply predicted that patient's average BG value for their diabetes history – that is, the naive model predicted the same average value (for that patient), independent of any other information about that patient[8]. More concretely, given a patient's data $D = \{[\ldots, BG_i, \ldots]\}_i$, partitioned into a training set D^{train} and a test set D^{test}, the model calculates the average blood glucose for

[8]This would be like a weatherman just predicting that the temperature tomorrow will be 3.6^oC every day – independent of the season, or today's temperature, or any other climatic features `http://www.edmonton.climatemps.com/temperatures.php`.

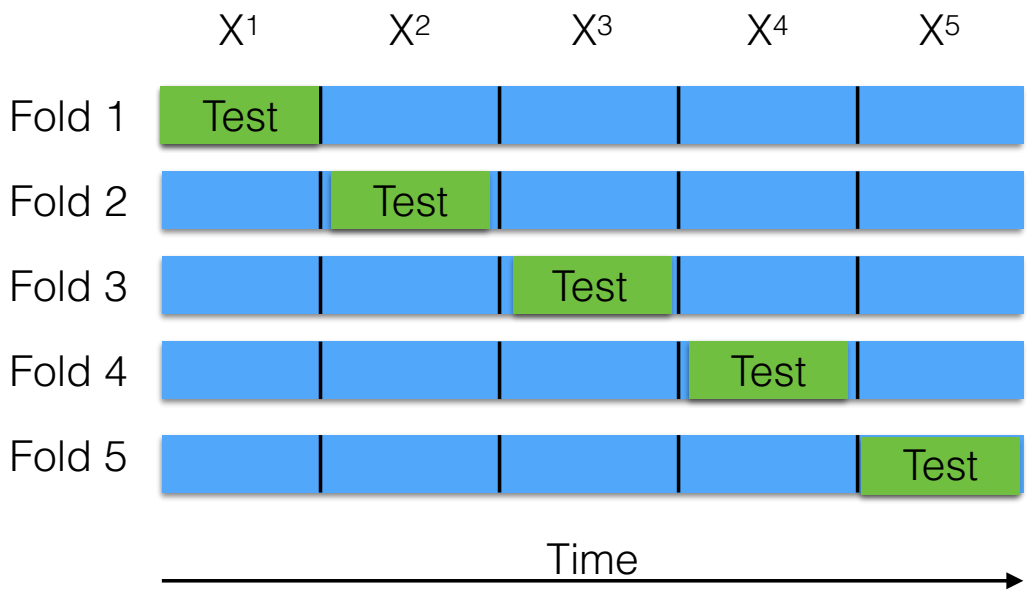

Figure 3.4: **Illustration of 5-Fold CV with Contiguous Segments.** In each CV iteration training is done on the blue segments and testing on the green segment.

the entire training set

$$BG_{avg}(D^{train}) \quad = \quad \frac{1}{|D^{train}|} \sum_{[...,BG_i,...] \in D^{train}} BG_i \tag{3.3}$$

including readings for all meals and all days. Then for all instances x_j in the associated test set D^{test}, this model sets $M_{avg}(\, x_j,\, \Delta t_{j+1}\,) \;=\; BG_{avg}(D^{train})$; it can then be used for evaluation, using Equations 3.1 and 3.2. So, for patient 16, because her BG_{avg} value for her first training set was 8.4, this trivial model predicts that her blood glucose value will be 8.4 for each meal in the associated test set.

3.5.4 Comparison to an Expert

To evaluate how well our model compares to an expert diabetes physician in terms of predicting blood glucose, we selected seven patients at random from our dataset. For each of these patients, we partitioned his/her data into ten contiguous segments (for cross validation) and sampled one record from each segment for each patient – leading to 70 records in total. We then removed

those records that our expert indicated did not adhere to the "EP" criteria (see Section 3.3). This left the 46 records from six remaining subjects that are used in the comparison. To make a prediction for any given BG value from one segment, our learning algorithm trained on the other nine segments that did not contain that BG value, whereas the expert studied the entire available patient history that preceded the BG value.

We compared each of our models to the expert, using both evaluation measures defined in Section 3.5.1: Equations 3.1 and 3.2.

To reduce bias, we constructed the experiment so that the patients and the data points were sampled randomly and the sampling procedure was not disclosed to the expert.

3.6 Machine Learning Models

3.6.1 Standard Machine Learning Algorithms and Parameters

Table 3.6: **Descriptions of the Different Learners Used.**

name	Symbol	Algorithm	Confidence Weighting	Stacking
gpr_be	M^{w}_{gpr}	GPR	1	0
gpr_be_AllPat_AllMeals	M^{ws}_{gpr}	GPR	1	1
gpr_IndPat_AllMeals	M_{gpr}	GPR	0	0
gpr_AllPat_AllMeals	M^{s}_{gpr}	GPR	0	1
svr1	M_{svr}	SVR (RBF Kernel)	0	0
svr1_lin	M^{lin}_{svr}	SVR (Linear Kernel)	0	0
svr1_allpats	M^{s}_{svr}	SVR (RBF Kernel)	0	1
rf4	M_{rf}	Random Forest	0	0
KNN10U	M_{knn}	KNN	0	0
ridge	M_{ridge}	Ridge Regression	0	0
wnn	M_{wnn}	Wavelet Neural Network	0	0
NN	M_{nn}	Feed-Forward NN	0	0
naive	M_{avg}	BG History Average	0	0

Confidence Weighting is explained in Section 3.6.2 and Stacking is explained in Section 3.6.3.

This work considers a range of possible learning algorithms run on each of the various different types of feature preprocessing variants that we have described previously, in Section 3.4. Again, refer to Section 2.3 for the descriptions of all the standard learning algorithms that we use: K-Nearest Neighbors

(KNN), Support Vector Regression (SVR), Artificial Neural Networks (ANN), Wavelet Neural Networks (WNN), Ridge Regression (RR), Random Forest Regression (RFR) and Gaussian Process Regression (GPR). Note that hyper-parameter tuning for the GPR model (nugget = 0.25), KNN model (K = 10, weighting = uniform), RF model (maximum depth = 4), and our neural network model (batch size = 20, epochs = 1000) was done using patient 16's first 3260 diabetes diary entries from dataset D21. These 3260 records were then excluded from our testing data in order to avoid overfitting. All other unspecified parameters were defaults – *e.g.*, the SVR model (C = 1), SVR with RBF kernel model (C = 1, $\sigma^2 = \frac{1}{\#\text{ of features}}$) and Ridge regression model ($\alpha = 1$) are the defaults provided by scikit-learn [29]. The ANN architecture included one output neuron with linear activation and two hidden layers of $3 \times$ (# of features) with rectified linear activation. The WNN architecture included one output neuron with linear activation and one hidden layer of (# of features) neurons with Gaussian wavelet activations ($\Psi(x) = -xe^{-\frac{1}{2}x^2}$). Most of these models were implemented with the help of `scikit-learn` [29], except for the `ANN` that was implemented using Keras [10] and the `WNN` that was implemented in part with `scikit-neuralnetwork`[9]

In this work we also combine base learners to develop more complex learners. The following section (Section 3.6.2) describes our GPR ensemble approach. We also considered another approach, which incorporates the information from the other patients – *e.g.*, use patients #1 to #50 to help train a model for patient #51. This "Stacking" approach first trains a model on the auxiliary patients, then runs this model on the test patient's data to produce a new feature for each record – *i.e.*, a 14th feature to augment the 13 features shown in Table A.2. Section 3.6.3 further describes our "stacking" approach and Figure 3.5 shows the entire stacking process. In total, we used 13 different models in this study, which are listed in Table 3.6.

[9] `http://scikit-neuralnetwork.readthedocs.io/en/latest/index.html`

3.6.2 Modeling with a confidence weighted GPR Ensemble, M^w_{gpr}

For each patient, our "GPR ensemble" model first creates two different GPR models, then combines them into a single model called M^w_{gpr}. The first of these two models, GPR_p, learns from the entirety of a patient's training data. The second model, $\{GRP_m\}$, is actually a collection of GPR models – one for each possible meal category m (corresponding to $meal_i$ in Table 3.3). Each of these GPR_m models is trained using only the occurrences of that particular meal category in the patient's training data (*e.g.*, all occurrences of "Before Lunch"). Once we have obtained GPR_p and the set of GRP_m models and wish to make a prediction for instance x_i, we produce a weighted prediction of the form

$$\hat{BG}_{i+1} \quad = \quad \frac{1}{\alpha+\beta}\,[\alpha\; GPR_p(\,[x_i, \Delta t_{i+1}]\,) \;+\; \beta\; GPR_m(\,[x_i, \Delta t_{i+1}]\,)\;] \quad (3.4)$$

where $\alpha = \frac{1}{\sigma_{p_i}}$ and $\beta = \frac{1}{\sigma_{mi}}$, and where σ_{p_i} and σ_{mi} are respectively the standard deviations of the posterior Gaussian distributions $\mathcal{N}(\,GPR_p(\,[x_i, \Delta t_{i+1}]\,),\, \sigma^2_{p_i}\,)$ and $\mathcal{N}(\,GPR_m(\,[x_i, \Delta t_{i+1}]\,),\, \sigma^2_{mi}\,)$ at the point x_i.

3.6.3 Incorporating Other Patient's Data with Stacking

Up to this point, all the previous learning algorithms described have been trained specifically on a single patient and then evaluated on new data from that same patient. To incorporate the available data from the other patients in the dataset, we use stacking (originally introduced as "stacked generalization" [43]), which involves training higher levels of learners on a combination of meta-data/data with the predictions of other basic learners [25]. In our case, this involves training a stacking learner on all the other patients to produce a stacking model that is then used to make BG predictions for each record associated with the patient under test (see Figure 3.5 for a description of the stacking process that we use). These BG predictions then become a new feature for the patient under test, and CV with this patient proceeds as described before. Specifically, we use SVR as our stacking learner and we use

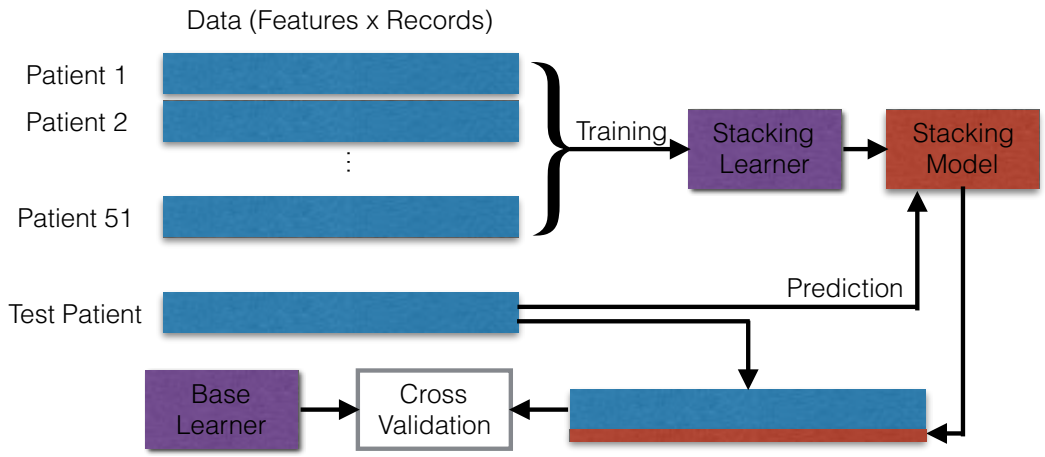

Figure 3.5: **How we Implement Stacking.**

GPR, confidence weighted GPR and SVR as our base learners. These produce the models M^s_{gpr}, M^{ws}_{gpr}, and M^s_{svr} respectively.

The rationale behind this stacking approach is motivated by the fact that our data is self reported, so some patients (such as patient 16) have a large number of records, while other patients have relatively few records. Consider the datasets (features and outcome) for two patients, $D_1 = \{(\mathbf{x}^1_i, y^1_i)|i = 1..n\}$ and $D_2 = \{(\mathbf{x}^2_j, y^2_j)|j = 1..m\}$ where $m \ll n$. Let $f_1 : \mathbf{X} \rightarrow Y$ and $f_2 : \mathbf{X} \rightarrow Y$ be the true underlying glucose functions for these patients. Both functions are sufficiently complex that a learning algorithm cannot learn a good approximation $\hat{f}_2$ when only learning with D_2, but a learning algorithm can learn a good approximation $\hat{f}_1$ when only learning with D_1. If f_1 and f_2 are closely related in some sense (*e.g.*, affinely related $f_2(\mathbf{x}) = a\, f_1(\mathbf{x}) + b$) then $\hat{f}_2$ can be learned using D_1 and D_2 by first learning $\hat{f}_1$. We can also view this stacking approach as representing knowledge that has been acquired from previous patients, to help better understand the current patient.

3.7 Further Investigations

After obtaining the results from the different models and dataset variants tested, we continued investigations to answer additional questions. For the investigations in this section, we used dataset D1 to continue testing new models. We chose dataset D1 specifically because, of the datasets that make

use of the data from all 51 patients, it is the dataset where models have the best performance on average. Section 4.2 reports the results of these two explorations.

3.7.1 Incorporating Patients with Similar BG Variances

Most of the earlier approaches use only data about the target patient – *i.e.*, using only information about the i^{th} patient when building a model for predicting that patient's BG. Here, we consider a way to use information from other patients when building a model. In particular, to learn a model for the i^{th} patient, we use information from other patients whose time series variance is similar to the i^{th} patient's. This is based on the assumption that BG variance is indicative of a patient's tolerance to insulin, or simply that patients might be in a similar stage of their disease.

Here, we first compute the variance of each patient's BG, then apply KMeans clustering to cluster patients, based only on this single feature. The KMeans parameter, $K \in \{3, 4, .., n\}$ for the n patients in our dataset, is selected using the maximum silhouette width value. Note that the value of this parameter and the resulting BG variance clusters may change on each iteration of cross validation. Once we have obtained the patients with similar BG variance to a target patient, we then incorporate these other patients in one of two ways: by including their data onto the target patient's training dataset, or by using their data to build a stacking model as described in Section 3.6.3. The results for this approach can be found later in Section 4.2.

3.7.2 Meal Specific Prediction and a Less Naive M_{avg}

In this investigation we quantify how much M^w_{gpr} improves predictive quality as compared to one of its components, $\{GRP_m\}$ (from Section 3.6.2). For simplicity, we refer to the model composed of $\{GRP_m\}$ as M^m_{gpr}. We also compare M^w_{gpr}, M^m_{gpr} and M_{avg} to a less naive version of our naive predictor (referred to as M^m_{avg}), which is a set of naive predictors $\{M_{avg,m}\}$. Here, each $M_{avg,m}$ is the naive predictor after training on the subset of D^{train} that is specific to a particular meal m. This means that, unlike M_{avg}, M^m_{avg} considers the meal category

of a new record when making a prediction for that record. Concretely, given a new record with meal category m, M^m_{avg} predicts the average value of training data that has the meal category m. Equation 3.5 shows M^m_{avg}'s prediction for a new sample with meal category m.

$$BG^m_{avg}(D^{train}) \quad = \quad \frac{\sum_{[...,BG_i,m_i,...]\in D^{train}} BG_i \mathbb{1}\{m_i = m\}}{\sum_{[...,m_i,...]\in D^{train}} \mathbb{1}\{m_i = m\}} \tag{3.5}$$

3.7.3 Predicting from Records without Injection or Ingestion

As was previously described, we set up our problem as predicting BG_{i+1} that is Δt_i minutes after the current record. We also noted that this was a simplification (explained in detail in Appendix 3.4) and that Δt_i is actually with respect to records that either contain carbohydrate ingestion values or bolus injection values.

Here, we consider a slightly easier BG modeling problem where we base the prediction on the immediately earlier record, removing the restriction that this earlier record must contain either carbohydrate ingestion values or bolus injection values. The purpose of this investigation is to see whether the performance of our M^w_{gpr} model is affected in order to quantify the performance impact suffered by a model that must operate under the previously described restriction. We can then determine how much more difficult the BG modeling problem is when you only forecast BG predictions from events where the patient has taken a BG modifying action. The results for this approach can be found later in Section 4.2.

3.7.4 Performance on a 3-way Classification Variant

This experiment was conducted to see how well a learned classifier would perform given that it only needed to predict general blood glucose categories instead of the complete spectrum of BG values. In this case, we consider a 3-way classification of BG (as low, normal or high) instead of the previous regression problem. To set up this scenario, the function in Equation 3.6 was

applied to all the labels in the dataset.

$$f(x) \quad = \quad \begin{cases} 1, & \text{if } x > 8 \\ -1, & \text{if } x < 4 \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

After transforming the labels, we applied the Random Forest learning algorithm to the data to produce a classifier. We selected this learner for this experiment because it performed well on the previous regression task (second to the GPR models) and because its components (decision trees) can be easily applied to multi-label classification problems. The maximum tree depth (maximum depth = 6) was selected using the same subset of patient#16's data that was used for selecting the hyper-parameters for the other models.

Chapter 4

Experimental Results

4.1 Cross Validation Results

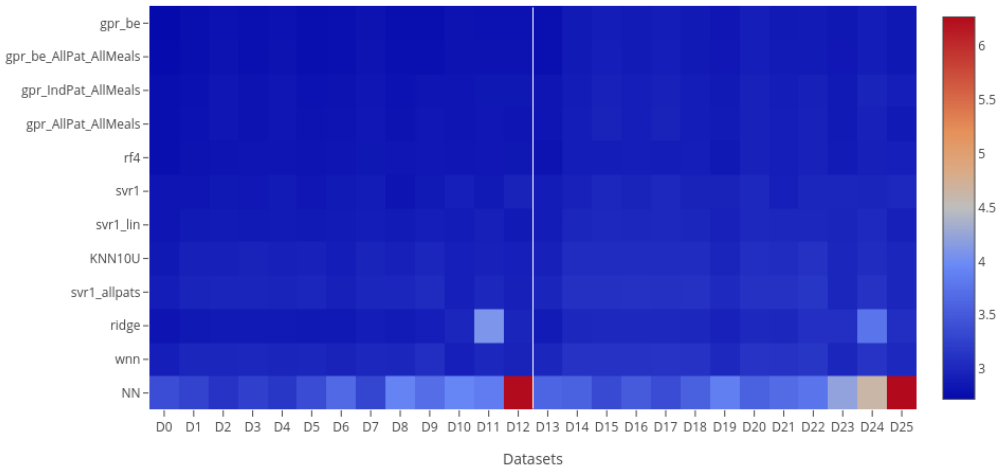

Figure 4.1: **Average L1 Loss: Datasets vs. Models.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square represents the cross-validation L1 error, micro-averaged over patients.

For each of the 51 patients (described in Section 3.1), we perform 10-fold CV using 12×26 different learner/dataset-variant combinations to determine their effectiveness and how well they compare to the baseline model, M_{ave} from Section 3.5.3. The complete results from these experiments are shown in Fig 4.1, Fig 4.2, Fig 4.3, and Fig 4.4 as heat maps. The corresponding tables for these heat maps can be found in Appendix C. These heat maps show both

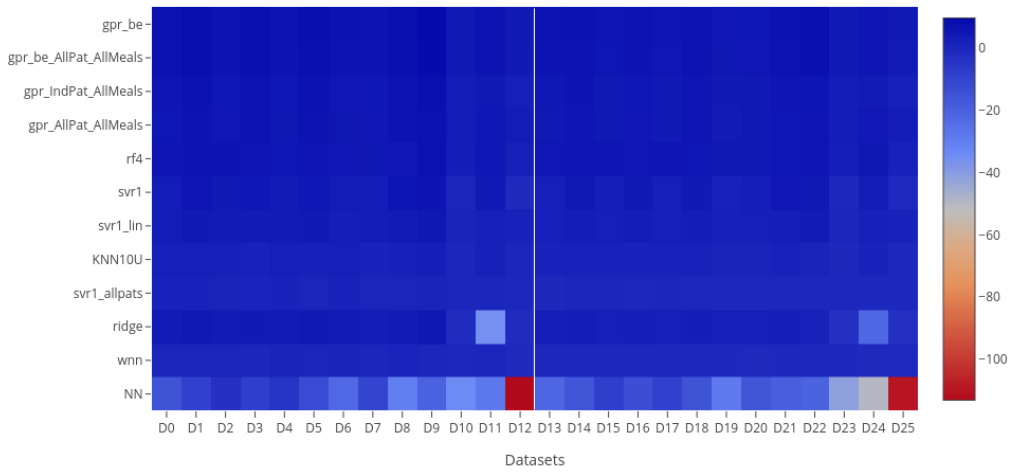

Figure 4.2: **Percent Improvement in Average L1 Loss for Models vs. Baseline.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square corresponds with the percent change between the corresponding result in Figure 4.1 and the performance of M_{ave}.

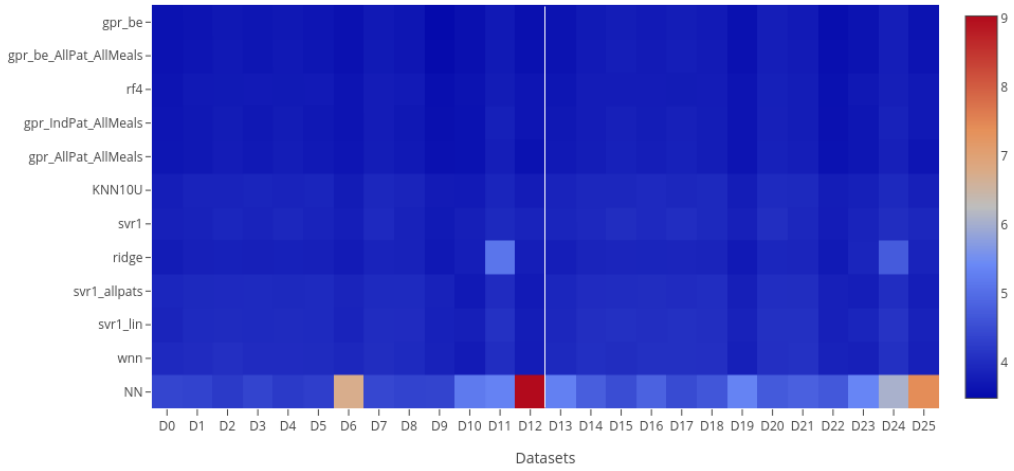

Figure 4.3: **Average Relative L1 Loss: Datasets vs. Models.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square represents the cross-validation relative L1 error, micro-averaged over patients.

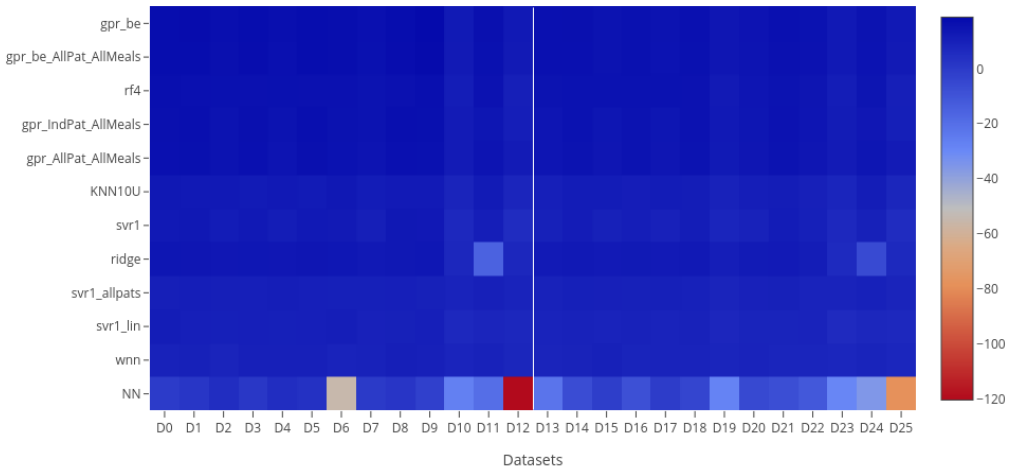

Figure 4.4: **Percent Improvement in Average Relative L1 Loss for Models vs. Baseline.** Datasets using the expert criteria are on the left half of the bisecting white line. Each square corresponds with the percent change between the corresponding result in Figure 4.3 and the performance of M_{ave}.

the performance of models on different datasets (in terms of err_{L1} and err_{rLl}), as well as the improved performance relative to M_{ave}.

The heat maps are organized so that the left half of each heat map contains the datasets that adhere to our EP rules, while the right half contains those datasets that do not. Models (on the y axis) are sorted in terms of their average err_{L1} error (over all 26 datasets), so that the model with the best average err_{L1} error across all datasets appears at the top of these figures. Further, datasets D0 to D12 and D13 to D25 (in each half of these heat maps) are sorted horizontally in increasing order of average err_{L1} error across all models, with the left most dataset in each half having the smallest error. Note that for Figure 4.1 the best model/dataset combination is found in the top left corner.

For each learner and dataset variant pair, we compute the err_{L1} error as a micro-average over all the records of each patient. This differs from a macro-average, which would take the simple unweighted average of the performance with respect to the patients. Micro-averaging means that patients with more

records implicitly get proportionally more weight. With these results, we can identify the pair with the lowest average err_{L1}, over all 51 patients. These studies found that M^w_{gpr} had the lowest err_{L1} on average across all of the 26 different preprocessing variants of the data. On dataset D0 (the preprocessing variant with the lowest average err_{L1} across all models), M_{ave}'s average err_{L1} was 2.94 mmol/L, while M^w_{gpr}'s average err_{L1} was 2.72 mmol/L – *i.e.*, our model saw an improvement of 7.5% relative to the baseline.

To help understand why the improvement is not greater, Figure 4.7 shows the predictions of M^w_{gpr} for the processed entries from patient#16 that were used for selecting hyperparameters. Here, we can see that the model is unable to account for the high amount of variance present in the BG records for this patient.

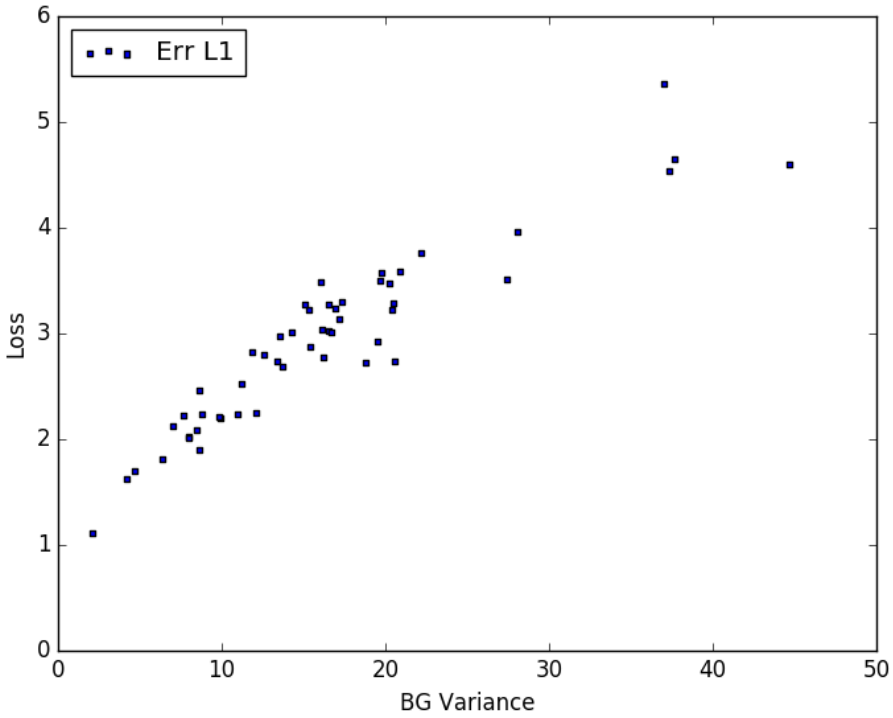

Figure 4.5: **Model M^w_{gpr}: average err_{L1} as a function of BG variance, for all 51 patients.**

Figure 4.5 plots the variance in each patient's BG history and the corresponding patient's err_{L1} loss (Equations 3.1) that M^w_{gpr} was able to achieve.

This figure shows that the variance of a patient's blood glucose was highly correlated with the err_{L1} test loss (0.93 Pearson Correlation).

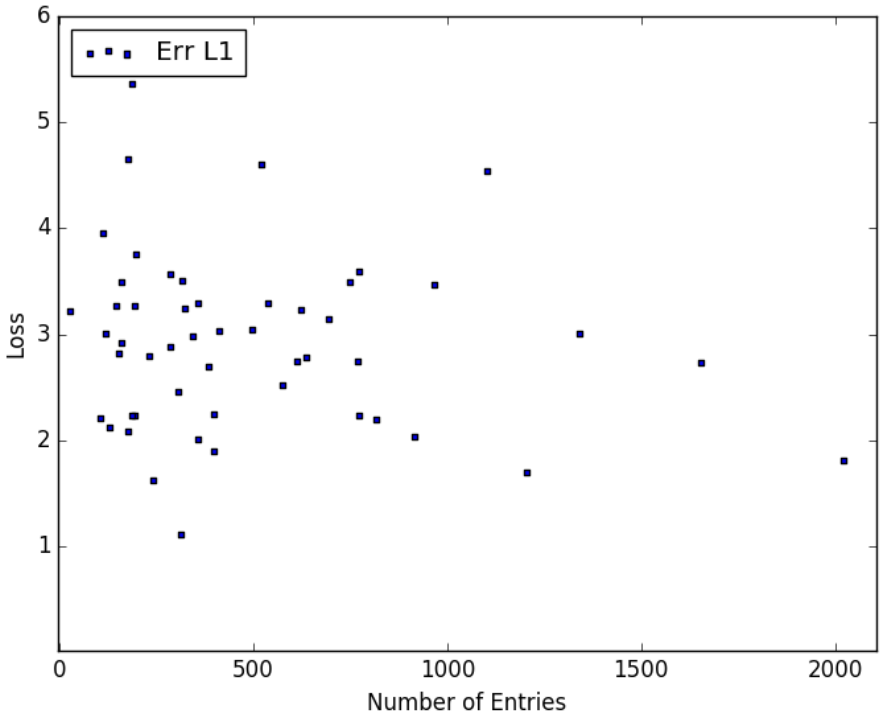

Figure 4.6: **Model M^w_{gpr}: average err_{L1} as a function of the # of diabetes diary entries for a patient, for all 51 patients.**

Figure 4.6 is a scatter plot of err_{L1} loss as a function of the number of data points that were available for each patient in the dataset. Figure 4.6 shows that there seems to be no relationship between how well the model performs on any particular patient and how many data points were collected from that patient, in terms of err_{L1} test loss (-0.14 Pearson Correlation).

We then considered the err_{rLl} loss and found that M^w_{gpr}'s err_{rLl} loss on D0 was 0.360. We also saw that M^w_{gpr} achieved the best err_{rLl}, although this was on a different dataset-variant. The best [model, dataset-variant] pair was M^w_{gpr} on dataset D9, which achieved an average err_{rLl} error of 0.348; this was an improvement of 18.9% relative to the M_{avg} baseline of 0.429. Note that M^w_{gpr} achieved an err_{L1} of 2.77 mmol/L dataset D9.

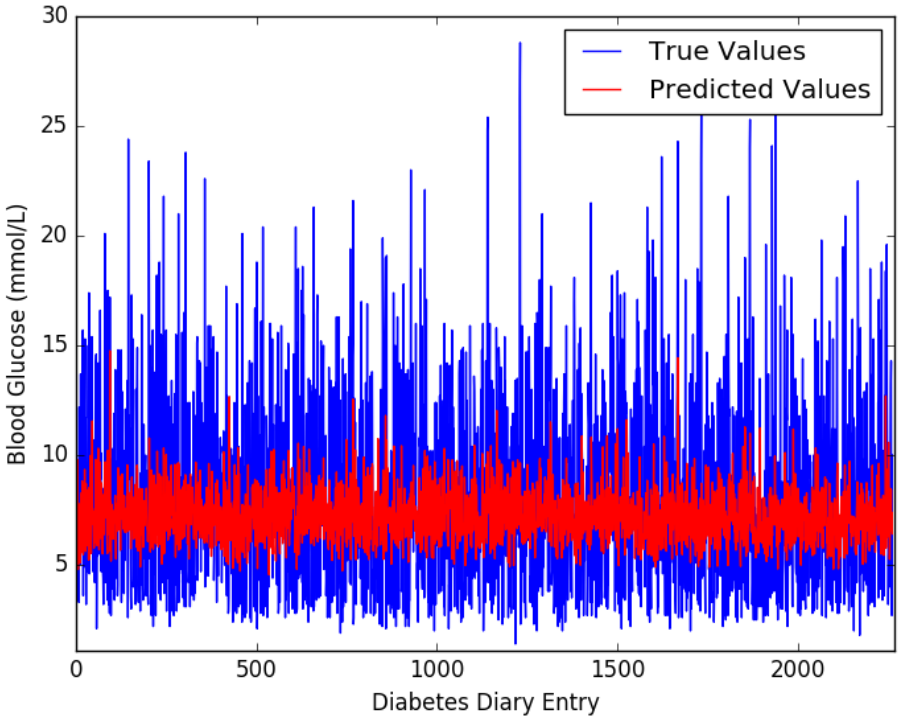

Figure 4.7: **Model M^w_{gpr}: Our GPR ensemble's predictions on data from patient 16.**

4.2 Results from our Further Investigations

Here, we report on the additional experiments described in Section 3.7. For Section 3.7.1, we tried two different approaches for incorporating the records from patients who had similar BG variance. For the first approach, incorporating the auxiliary patient's data by adding it to the training set on each CV iteration (using M^w_{gpr}) gave us an err_{L1} of 2.80 mmol/L. The second approach uses the auxiliary patient's data to train a stacking model (M^{ws}_{gpr}). It then produces a new feature which is appended to the training and test set data. This second approach gave us an err_{L1} of 2.77 mmol/L.

From these results we see that incorporating the information of other patients with similar BG variances does not improve the model. Again, this supports the idea that training data from other patients will not improve any model for a particular patient.

In our next experiment, from Section 3.7.2, we compared four models, M^w_{gpr}, M^m_{gpr}, M^m_{avg}, and M_{avg}. On dataset D1, we found that M^w_{gpr} achieved an err_{L1} of 2.77 mmol/L, M^m_{gpr} achieved an err_{L1} of 2.78 mmol/L, M^m_{avg} achieved an err_{L1} of 2.93 mmol/L and M_{avg} achieved an err_{L1} of 3.01 mmol/L. These results show that M^w_{gpr} only marginally outperforms M^m_{gpr} on this dataset, indicating that meal specific learning, by itself, already captures the majority of the available signal in the data. Additionally, we see that M^m_{avg} outperforms M_{avg}, which suggests that if we had used M^m_{avg} as our baseline in the original 312 experiments, our models would have given even smaller percentage improvements relative to M^m_{avg}.

In our third experiment, from Section 3.7.3, we studied the impact of relaxing the restriction that the previous record to the one for which we are making a prediction must have recorded carbohydrate ingestion or insulin injection. We found that model M^w_{gpr}, when applied to dataset D1 without the restriction, achieved an err_{L1} of 2.76 mmol/L. This was the same result within two decimal places the restriction (before rounding). These results show that for our dataset, training algorithms applied to data with and without this restriction produce equivalently accurate models in either case.

Table 4.1: **Average Errors using M^w_{gpr} on Dataset D2.**

Patient	Records	BG σ^2	Test err_{L1}	Train err_{L1}	M_{ave} err_{L1}
1	81	7.67	2.22	1.42	2.15
2	38	6.99	2.06	1.12	2.08
3	750	20.23	3.41	2.89	3.63
4	96	19.53	2.79	2.03	3.27
5	219	15.06	3.13	2.6	3.05
6	224	2.13	1.11	0.81	1.13
7	83	8.49	2.09	1.5	2.19
8	99	12.56	2.86	2.18	2.78
9	469	17.21	3.13	2.7	3.31
10	538	20.89	3.55	2.87	3.81
11	191	20.49	3.32	2.73	3.67
12	1373	18.8	2.73	2.56	3.39
13	110	37.67	4.72	2.91	5.0
14	588	37.35	4.5	4.18	4.92
15	416	19.71	3.52	3.0	3.64
16	1012	16.71	3.01	2.63	3.22
17	647	8.81	2.25	1.89	2.32
18	86	10.99	2.28	1.91	2.33
19	51	15.09	3.27	2.06	2.95
20	436	13.41	2.72	2.43	2.87
21	699	9.91	2.19	1.91	2.49
22	529	16.19	2.8	2.55	3.1
23	42	9.8	2.17	1.47	2.15
24	621	7.97	2.04	1.79	2.17
25	418	20.39	3.21	2.6	3.66
26	151	19.73	3.44	2.49	3.64
27	138	13.72	2.67	2.47	2.66
28	89	16.55	3.29	1.98	3.47
29	54	13.6	2.75	1.79	2.84
30	115	15.38	2.83	2.1	3.38
31	79	37.0	5.46	3.36	5.68
32	270	17.33	3.32	2.93	3.41
33	220	16.9	3.21	2.32	3.36
34	123	22.2	3.68	2.76	3.76
35	149	7.97	1.99	1.6	2.02
36	255	13.54	3.03	2.44	3.04
37	68	16.02	3.52	2.28	3.33
38	291	16.09	3.03	2.51	3.26
39	296	20.51	2.76	2.82	3.77
40	246	44.65	4.58	4.3	5.17
41	639	4.66	1.71	1.52	1.75
42	245	12.12	2.3	2.14	2.82
43	164	8.61	2.42	1.68	2.39
44	280	16.51	2.96	2.36	3.32
45	171	27.42	3.58	2.82	3.62
46	325	11.21	2.49	2.25	2.76
47	153	4.22	1.61	1.11	1.64
48	70	11.84	2.84	1.72	2.74
49	30	28.06	3.95	2.58	3.65
50	1737	6.38	1.8	1.65	1.96
51	204	8.02	1.92	1.66	2.11

Here, Records indicates how many records were retained after pre-processing.
BG σ^2 indicates the variance of a patient's blood glucose history.
Test/Train/M_{avg} err_{L1} is the CV error for our model on the test data, the training data, and the baseline model on the test data respectively.

To show how well the M^{ws}_{gpr} does on the original D1 dataset, we give the complete breakdown of per patient performance in Table 4.1.

In our final addition experiment (Section 3.7.4), we consider a 3-way classification version of our original regression problem. We found that a random forest classifier, learned and evaluated through cross validation, was able to achieve an accuracy of 59.3% on micro-average across our 51 test patients. In comparison, the common baseline accuracy of selecting the majority class for each prediction was able to achieve an accuracy of 56.3%. Here we can see that the random forest classifier was able to achieve very slight gains over the majority class predictor.

4.3 Comparison to an Expert

As was described in Section 3.5.4, our expert provided his prediction for 46 records from six patients. For these records we also calculated the err_{L1} ($err_{L1} = 2.88$ mmol/L) of our best model M^{w}_{gpr} and found that it outperforms both the baseline model, M_{avg}, ($err_{L1} = 3.82$ mmol/L, p-value = 0.0005[1]) and the expert ($err_{L1} = 3.19$ mmol/L, p-value = 0.4), although the latter is not statistically significant.

We also looked at the err_{rLl}, which emphasizes the penalty associated with mis-predicting hypoglycemic values, and found that the expert had a lower err_{rLl} ($err_{rLl} = 0.444$, p-value = 0.5) as compared to M^{w}_{gpr} ($err_{rLl} = 0.499$) and M_{avg} ($err_{rLl} = 0.752$, p-value = 0.0004). This is in contrast to the err_{L1} results and may be because our model was not optimized to learn with this particular loss, but that diabetologists are likely trained with a bias towards preventing hypoglycemic events.

For further details on both of these sets of comparison results, see Table 4.2 and Table 4.3.

We also timed our expert to see how long he required to make these predictions. We found that our expert required an average of 77 seconds for each prediction, whereas our model required an average of 0.15 seconds. Note that

[1]All of these p-values are based on paired t-tests, with respect to the M^{w}_{gpr} model.

Table 4.2: **Comparison of an Expert against our model across 6 patients (L_1 Error)**

Patient	# of points	M_{avg} Average Error	Expert Average Error	M^w_{gpr} Average Error
10	7	4.87	2.16	2.75
13	9	3.95	4.78	3.57
45	8	3.25	2.96	2.39
16	9	3.90	3.89	2.62
12	7	2.86	2.26	2.65
11	6	4.13	2.37	3.28
Overall Average:		3.82	3.19	2.88

Table 4.3: **Comparison of an Expert against our model across 6 patients (Relative L_1 Error)**

Patient	# of points	M_{avg} Average Error	Expert Average Error	M^w_{gpr} Average Error
10	7	1.059	0.278	0.588
13	9	0.737	0.756	0.629
45	8	0.335	0.312	0.237
16	9	0.620	0.453	0.308
12	7	0.699	0.466	0.529
11	6	1.229	0.306	0.798
Overall Average:		0.752	0.444	0.499

our model's average time also includes the total training time required for the model.

Chapter 5

Discussion and Conclusions

5.1 Discussion

Our results show that our best learning algorithm is more accurate than a naive baseline – but only slightly – and that it can only achieve an average err_{L1}-loss of approximately 2.72 mmol/L. This loss means that, on average, if the patient's blood glucose was normal (*e.g.*, 6 mmol/L), the learned model may incorrectly identify the patient as either hypoglycemic (as $6 - 2.72 < 4$ mmol/L) or hyperglycemic ($6 + 2.72 > 8$ mmol/L). Together with the strong relationship between glucose variance and prediction error, this highlights how challenging it is to create models that produce fine-grained blood glucose predictions when only using diabetes diary entries – *i.e.*, using only the information that is commonly available to medical practitioners.

Having tried 312 different combinations of learners and dataset variants, and observing minimal differences in their performance, it seems unlikely (although not impossible) that further performance gains can be achieved without overfitting new models to the data. Of course, avoiding the overfitting of models to data is necessary when one wishes to make generalized claims about model performance. One noteworthy finding of these results is that efforts to combine patient data to build enhanced predictors were unsuccessful. This is interesting because it indicates that simply including more patients in the study is not likely to improve model performance. Moreover, since there does not seem to be a strong relationship between the number of data points that a patient has recorded and the performance of a model on that patient,

collecting more of this type of data (sampled before and after meals) for each individual likely will not improve model performance.

There are many possible reasons why modeling T1D glucose levels based on diabetes diary data is so challenging. In particular, it may be the case that inaccuracies and omissions of variables in data prevent the model from producing accurate predictions. These could possibly include: not knowing the site where the bolus insulin was injected, how much scar tissue was present at the injection site, skin temperature, how accurately the carbohydrate value was recorded, the accuracy of the recorded insulin dose, the levels of different hormones, whether or not the patient was menstruating, stress levels, accuracy of recording exertion, insulin age, amount of blood flow at the injection site and possibly many others factors. Given our belief that training more accurate models will require additional relevant variables, future research might incorporate more confounding variables, such as injection location [20], glucagon levels [40] and/or meal protein content [28]. However, it is not clear which, if any, of such variables are sufficient to explain the response, nor whether they can be practically captured in a clinical setting.

5.2 Conclusions

This work explored the challenge of accurately predicting future blood glucose values in Type I diabetes patients, based on a model learned using machine learning algorithms. Our extensive explorations – involving 12 different learning algorithms, and 26 different encodings of the data (312 combinations) found that, on average, the model with the lowest expected err_{L1} was a confidence weighted Gaussian process regression model (M^w_{gpr}). Using 10-fold cross validation on 30 221 blood glucose records from 51 patients, our M^w_{gpr} model performed 7.5% better than a naive "mean predicting" model (M_{avg}). We also found that this model's predictions (insignificantly) outperformed an expert diabetologist in terms of a simple unbiased loss function, but that the expert performed (insignificantly) better when the evaluation was biased toward predicting hypoglycemic events.

These results showed that our model could achieve an expected absolute error of 2.72 mmol/L, which is disconcertingly large given that this is based on the type of data that is frequently collected and used for clinical practice (records are collected at meal times by the patients themselves). These results strongly suggest that the standard data collected by T1D patients, while apparently appropriate for clinical treatment of T1D, is not sufficient for accurately predicting blood glucose levels. We conjecture that using patient data that is sampled more frequently and that includes additional features would improve both the ability of professionals and machine learning practitioners to more accurately predict patients' blood glucose levels, but there is a practical trade-off between patient convenience and highly detailed record keeping.

5.3 Directions for Future Work

While we have shown that it is difficult to build accurate BG models using diabetes diary entries, there is still opportunity for further research. In particular, while we showed that training on a specific patient's data produced better models than training on all patients' data, it is possible that an alternative method for training models on multiple patients' data could prove more effective than our approach. Techniques from the literature on domain adaptation and transfer learning (such as Bi-shifting Autoencoders [19]) could potentially be applied to address this problem. Another avenue of future work could be to build models using multiple sources of data. For example, patients could wear devices that constantly monitor their activity levels, blood glucose levels and vital signs. Patients could also take pictures of each of their meals, and machine learned models could be applied to these photographs to more accurately predict the amount of carbohydrate, protein, and lipid present in the meal. In addition, improved methods for measuring BG values (*e.g.*, Flash Glucose Monitoring [12]) could produce data that would be more easily modeled. Finally, cost-sensitive learning could be applied to this data in order to learn models that are more sensitive to hypoglycemic events. This type of learning would then produce models that would preform better when evaluated

with the err_{rLl} loss.

While our study has not revealed an easy route for applying machine learning algorithms to predict future blood glucose levels in T1D patients, our work provides a systematic investigation of a large data set, employing 312 data/learning algorithm combinations, upon which future research can build.

Bibliography

[1] Ahmad M Al-Taee, Majid A Al-Taee, Waleed Al-Nuaimy, Zahra J Muhsin, and Hamzah AlZu'bi. Smart bolus estimation taking into account the amount of insulin on board. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 1051–1056. IEEE, 2015.

[2] Steen Andreassen, Jonathan J Benn, Roman Hovorka, Kristian G Olesen, and Ewart R Carson. A probabilistic approach to glucose prediction and insulin dose adjustment: description of metabolic model and pilot evaluation study. *Computer methods and programs in biomedicine*, 41(3-4):153–165, 1994.

[3] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[4] Golnaz Baghdadi and Ali Motie Nasrabadi. Controlling blood glucose levels in diabetics by neural network predictor. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 3216–3219. IEEE, 2007.

[5] Meysam Bastani. Model-free intelligent diabetes management using machine learning. Master's thesis, Department of Computing Science, University of Alberta, 2014.

[6] Jeffrey A Bluestone, Kevan Herold, and George Eisenbarth. Genetics, pathogenesis and clinical interventions in type 1 diabetes. *Nature*, 464(7293):1293, 2010.

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] Razvan Bunescu, Nigel Struble, Cindy Marling, Jay Shubrook, and Frank Schwartz. Blood glucose level prediction using physiological models and support vector regression. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 1, pages 135–140. IEEE, 2013.

[9] Salim Chemlal, Sheri Colberg, Marta Satin-Smith, Eric Gyuricsko, Tom Hubbard, Mark W Scerbo, and Frederic D McKenzie. Blood glucose individualized prediction for type 2 diabetes using iphone application. In *Bioengineering Conference (NEBEC), 2011 IEEE 37th Annual Northeast*, pages 1–2. IEEE, 2011.

[10] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[11] Denis Daneman. Type 1 diabetes. *The Lancet*, 367(9513):847–858, 2006.

[12] Anna R Dover, Roland H Stimson, Nicola N Zammitt, and Fraser W Gibb. Flash glucose monitoring improves outcomes in a type 1 diabetes clinic. *Journal of diabetes science and technology*, page 1932296816661560, 2016.

[13] David L Duke. *Intelligent diabetes assistant: A telemedicine system for modeling and managing blood glucose*. Carnegie Mellon University, 2010.

[14] Meriyan Eren-Oruklu, Ali Cinar, and Lauretta Quinn. Hypoglycemia prediction with subject-specific recursive time-series models, 2010.

[15] Eleni I Georga, Vasilios C Protopappas, Demosthenes Polyzos, and Dimitrios I Fotiadis. Online prediction of glucose concentration in type 1 diabetes using extreme learning machines. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 3262–3265. IEEE, 2015.

[16] John E Gerich. Glucose counterregulation and its impact on diabetes mellitus. *Diabetes*, 37(12):1608–1617, 1988.

[17] Richard IG Holt, Clive Cockram, Allan Flyvbjerg, and Barry J Goldstein. *Textbook of diabetes*. John Wiley & Sons, 2017.

[18] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[19] Meina Kan, Shiguang Shan, and Xilin Chen. Bi-shifting auto-encoder for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3846–3854, 2015.

[20] Veikko A Koivisto, Philip Felig, et al. Alterations in insulin absorption and in blood glucose control associated with varying insulin injection sites in diabetic patients. *Ann Intern Med*, 92(1):59–61, 1980.

[21] Peter Kok. Predicting blood glucose levels of diabetics using artificial neural networks. *Research Assignment for Master of Science, Delft University of Technology*, 2004.

[22] Oliver Kramer. Dimensionality reduction by unsupervised k-nearest neighbor regression. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, volume 1, pages 275–278. IEEE, 2011.

[23] Katrin Lunze, Tarunraj Singh, Marian Walter, Mathias D Brendel, and Steffen Leonhardt. Blood glucose control algorithms for type 1 diabetic patients: A methodological review. *Biomedical Signal Processing and Control*, 8(2):107–119, 2013.

[24] Paolo Magni and Riccardo Bellazzi. A stochastic model to assess the variability of blood glucose time series in diabetic patients self-monitoring. *IEEE Transactions on biomedical engineering*, 53(6):977–985, 2006.

[25] Zoltan-Csaba Marton, Florian Seidel, Ferenc Balint-Benczedi, and Michael Beetz. Ensembles of strong learners for multi-cue classification. *Pattern Recognition Letters*, 34(7):754–761, 2013.

[26] Scott M Pappada, Brent D Cameron, and Paul M Rosman. Development of a neural network for prediction of glucose concentration in type 1 diabetes patients. *Journal of diabetes science and technology*, 2(5):792–801, 2008.

[27] Scott M Pappada, Brent D Cameron, Paul M Rosman, Raymond E Bourey, Thomas J Papadimos, William Olorunto, and Marilyn J Borst. Neural network-based real-time prediction of glucose in patients with insulin-dependent diabetes. *Diabetes technology & therapeutics*, 13(2):135–141, 2011.

[28] MA Paterson, CEM Smart, PE Lopez, P Howley, P McElduff, J Attia, C Morbey, and BR King. Increasing the protein quantity in a meal results in dose-dependent effects on postprandial glucose levels in individuals with type 1 diabetes mellitus. *Diabetic Medicine*, 34(6):851–854, 2017.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[30] Kevin Plis, Razvan Bunescu, Cindy Marling, Jay Shubrook, and Frank Schwartz. A machine learning approach to predicting blood glucose levels for diabetes management. *Modern Artificial Intelligence for Health Analytics. Papers from the AAAI-14*, 2014.

[31] Charlene C Quinn, Michelle D Shardell, Michael L Terrin, Erik A Barr, Shoshana H Ballew, and Ann L Gruber-Baldini. Cluster-randomized trial of a mobile phone personalized behavioral intervention for blood glucose control. *Diabetes care*, 34(9):1934–1942, 2011.

[32] C Rasmussen and C Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. MIT Press, 2005.

[33] Ryan M Rifkin and Ross A Lippert. Notes on regularized least squares. Technical report mit-csailtr-2007-025, Computer Science and Artificial Intelligence Laboratory, MIT, 2007.

[34] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[35] Edmond A Ryan, Joanna Holland, Eleni Stroulia, Blerina Bazelli, Stephanie A Babwik, Haipeng Li, Peter Senior, and Russ Greiner. Improved a1c levels in type 1 diabetes with smartphone app use. *Canadian journal of diabetes*, 41(1):33–40, 2017.

[36] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[37] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[38] Bharath Sudharsan, Malinda Peeples, and Mansur Shomali. Hypoglycemia prediction using machine learning models for patients with type 2 diabetes. *Journal of diabetes science and technology*, 9(1):86–90, 2014.

[39] Volker Tresp, Thomas Briegel, and John Moody. Neural-network models for the blood glucose metabolism of a diabetic. *IEEE Transactions on Neural networks*, 10(5):1204–1213, 1999.

[40] Roger H Unger and Alan D Cherrington. Glucagonocentric restructuring of diabetes: a pathophysiologic and therapeutic makeover. *The Journal of clinical investigation*, 122(1):4, 2012.

[41] John Joseph Valletta, Andrew J Chipperfield, and Christopher D Byrne. Gaussian process modelling of blood glucose response to free-living physical activity data in people with type 1 diabetes. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 4913–4916. IEEE, 2009.

[42] Junfeng Wen and Negar Hassanpour Russell Greiner. Weighted gaussian process for estimating treatment effect.

[43] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[44] Zarita Zainuddin, Ong Pauline, and Cemal Ardil. A neural network approach in predicting the blood glucose level for diabetic patients. *International Journal of Computational Intelligence*, 5:72–79, 2009.

Appendix A

Complete Description of Feature

As mentioned earlier, the features that we described in Section 3.2 are simplified so that they can be easily understood. We now describe the complete set of features (shown in Table A.1) that we created after imputing missing values and removing unusable records.

The previously described Δt_i and BG_i features are each decomposed into two different features. Δt_i is separated into Δt_i^{bolus} and Δt_i^{CHO}, and BG_i is separated BG_i^{bolus} and BG_i^{CHO}. These correspond to the time since most recent (previous) bolus injection Δt_i^{bolus} (when $bolus_{i-}$ bolus units were injected and the blood glucose was BG_i^{bolus}), and the time since the most recent previous carbohydrate consumption Δt_i^{CHO} (when the subject consumed CHO_{i-} and their blood glucose was BG_i^{CHO}). This distinction is made because while many events correspond to both injecting a quantity of insulin and also consuming some carbohydrates, this is not always true.

Table A.1: **Description of Original and Processed Features used in this Study**

$meal_i$	The time of day: { Before Breakfast, After Breakfast, Before lunch, After Lunch, Before Supper, After Supper, Before Bed, During the Night}
DOW_i	The day of the week
ExV_i	Numeric encoding of exercise value: $\{2, 4, 7, 10\}$
PV_i	Pump Value: The rate at which the insulin pump is infusing (units/hour). This is always 0 if the patient does not have a pump
$basal_i$	The units of background insulin injected
BG_i	The BG value at the current time (mmol/L)
IOB_i	Insulin on Board: Estimated residual insulin from the previous injection (mmol/L)
CHO_{i-}	The previous most recent amount of carbohydrates ingested (grams)
$bolus_{i-}$	The previous most recent amount of insulin injected (units)
BG_{i-}^{CHO}	The BG value at the time that CHO_{i-} was ingested (mmol/L)
BG_{i-}^{bolus}	The BG value at the time that $bolus_{i-}$ was injected (mmol/L)
Δt_i^{CHO}	The time between CHO_{i-} and BG_{i+1} (min)
Δt_i^{bolus}	The time between $bolus_{i-}$ and BG_{i+1} (min)

Table A.2: **Example of Processed Data, over a single day, from Patient 16 (Variant D1)**

$meal_i$	Before Breakfast	After Breakfast	Before Lunch	After Lunch	Before Dinner	After Dinner
DOW_i	Tuesday	Tuesday	Tuesday	Tuesday	Tuesday	Tuesday
ExV_i	4	4	4	4	4	4
PV_i	0.50	0.50	0.63	0.45	0.90	0.90
$basal_i$	0	0	0	0	0	0
BG_i	16.2	14.7	5.6	6.8	10.5	3.0
IOB_i	0.00	7.90	3.61	0.89	0.81	3.35
CHO_{i-}	17.5	30.0	20.5	30.0	18.5	15.0
$bolus_{i-}$	1.93	10.40	2.44	3.00	2.54	3.80
BG_{i-}^{CHO}	10.3	16.2	14.7	5.6	6.8	10.5
BG_{i-}^{bolus}	10.3	16.2	14.7	5.6	6.8	10.5
Δt_i^{CHO}	540	103	120	196	187	89
Δt_i^{bolus}	540	103	120	196	187	89

Note that occasionally BG_i^{CHO} will differ from BG_i^{bolus} and Δt_i^{CHO} will differ from Δt_i^{bolus}. In these cases carbohydrates and insulin were not taken at the same time.

Appendix B

Table of EP Record Proportions

Table B.1: **Proportion of Data Adhering to the Expert's Prediction Criteria**

ID (Processed Records, Expert Criteria Records, Proportion Kept)		
1 (192, 81, 0.422)	18 (188, 86, 0.457)	36 (344, 255, 0.741)
2 (129, 38, 0.295)	19 (146, 51, 0.349)	37 (159, 68, 0.428)
3 (966, 750, 0.776)	20 (611, 436, 0.714)	38 (495, 291, 0.588)
4 (160, 96, 0.6)	21 (818, 699, 0.855)	39 (770, 296, 0.384)
5 (325, 219, 0.674)	22 (637, 529, 0.83)	40 (519, 246, 0.474)
6 (314, 224, 0.713)	23 (106, 42, 0.396)	41 (1205, 639, 0.53)
7 (175, 83, 0.474)	24 (916, 621, 0.678)	42 (399, 245, 0.614)
8 (232, 99, 0.427)	25 (621, 418, 0.673)	43 (306, 164, 0.536)
9 (695, 469, 0.675)	26 (287, 151, 0.526)	44 (410, 280, 0.683)
10 (773, 538, 0.696)	27 (384, 138, 0.359)	45 (315, 171, 0.543)
11 (356, 191, 0.537)	28 (195, 89, 0.456)	46 (576, 325, 0.564)
12 (1653, 1373, 0.831)	29 (118, 54, 0.458)	47 (242, 153, 0.632)
13 (176, 110, 0.625)	30 (287, 115, 0.401)	48 (154, 70, 0.455)
14 (1102, 588, 0.534)	31 (186, 79, 0.425)	49 (113, 30, 0.265)
15 (747, 416, 0.557)	32 (536, 270, 0.504)	50 (2023, 1737, 0.859)
16^1 (2998, 2264, 0.755)	33 (322, 220, 0.683)	51 (397, 204, 0.514)
16^2 (1340, 1012, 0.755)	34 (197, 123, 0.624)	
17 (772, 647, 0.838)	35 (356, 149, 0.419)	

On average (macro-average), 57.3% of data points for each patient met the expert's criteria. Patient 50 had the smallest percentage (26.5%) of records meeting the criteria and patient 51 had the largest percentage (85.9%) of records meeting the criteria. Note that patient 16 appears twice because part of that patient's data was used for the validation of the hyperparameters (16^1) and the rest was used as part of the test data (16^2)

Appendix C

Tables Corresponding To Heatmaps

Table C.1: **Losses Corresponding to Figure 4.1.**

	M^{w}_{gpr}	M^{ws}_{gpr}	M_{gpr}	M^{s}_{gpr}	M_{rf}	M_{svr}	M^{lin}_{svr}	M_{knn}	M^{s}_{svr}	M_{ridge}	M_{wnn}	M_{nn}
D0	2.72	2.73	2.77	2.77	2.77	2.83	2.83	2.87	2.91	2.82	2.92	3.38
D1	2.77	2.77	2.79	2.8	2.81	2.83	2.88	2.94	2.97	2.87	2.99	3.28
D2	2.8	2.81	2.84	2.84	2.82	2.87	2.89	2.94	2.98	2.88	2.99	3.12
D3	2.78	2.78	2.81	2.81	2.83	2.85	2.88	2.96	2.98	2.88	3.0	3.25
D4	2.8	2.81	2.84	2.84	2.83	2.89	2.89	2.94	2.97	2.88	2.98	3.16
D5	2.77	2.77	2.8	2.81	2.82	2.84	2.88	2.95	2.99	2.87	2.99	3.37
D6	2.78	2.78	2.81	2.82	2.83	2.88	2.89	2.91	2.94	2.87	2.96	3.65
D7	2.81	2.82	2.84	2.85	2.86	2.9	2.91	2.97	2.99	2.91	3.0	3.3
D8	2.77	2.78	2.8	2.81	2.84	2.82	2.89	2.94	2.99	2.89	2.99	3.9
D9	2.77	2.78	2.83	2.85	2.85	2.88	2.92	2.99	3.04	2.92	3.07	3.7
D10	2.81	2.81	2.83	2.83	2.84	2.93	2.9	2.93	2.93	2.99	2.93	3.93
D11	2.8	2.81	2.86	2.85	2.85	2.88	2.94	2.95	3.0	4.08	3.0	3.84
D12	2.8	2.81	2.86	2.83	2.86	2.96	2.88	2.93	2.94	2.98	2.96	6.26
D13	2.78	2.78	2.83	2.83	2.81	2.9	2.9	2.94	2.98	2.89	2.99	3.61
D14	2.88	2.88	2.9	2.91	2.91	2.95	2.99	3.05	3.09	3.0	3.11	3.58
D15	2.91	2.92	2.95	2.96	2.91	3.0	3.01	3.05	3.09	3.01	3.11	3.35
D16	2.89	2.89	2.92	2.92	2.92	2.97	3.0	3.05	3.1	3.01	3.11	3.51
D17	2.91	2.92	2.95	2.96	2.91	3.01	3.01	3.05	3.09	3.01	3.12	3.37
D18	2.88	2.88	2.91	2.91	2.92	2.96	2.99	3.05	3.1	3.0	3.11	3.56
D19	2.84	2.85	2.88	2.89	2.87	2.96	2.95	2.98	3.03	2.95	3.02	3.86
D20	2.92	2.92	2.95	2.95	2.95	3.01	3.01	3.07	3.1	3.02	3.12	3.58
D21	2.88	2.89	2.91	2.92	2.92	2.93	3.0	3.05	3.1	3.0	3.11	3.68
D22	2.89	2.89	2.94	2.95	2.95	2.99	3.0	3.1	3.15	3.08	3.15	3.77
D23	2.86	2.85	2.88	2.88	2.89	2.99	2.98	2.99	2.99	3.08	2.99	4.21
D24	2.91	2.91	2.97	2.95	2.94	2.98	3.03	3.05	3.11	3.77	3.12	4.64
D25	2.86	2.86	2.92	2.88	2.93	3.01	2.94	2.99	2.99	3.07	3.01	6.27

See Table 3.5 for descriptions of the different datasets and Table 3.6 for descriptions of the models.

Table C.2: **Percentage Improvements Corresponding to Figure 4.2.**

	M^{w}_{gpr}	M^{ws}_{gpr}	M_{gpr}	M^{s}_{gpr}	M_{rf}	M_{svr}	M^{lin}_{svr}	M_{knn}	M^{s}_{svr}	M_{ridge}	M_{wnn}	M_{nn}
D0	7.49	7.37	6.0	5.8	5.99	3.86	3.83	2.47	1.24	4.08	0.73	-14.94
D1	8.13	7.94	7.21	6.92	6.49	6.06	4.37	2.42	1.28	4.58	0.71	-8.96
D2	6.91	6.77	5.8	5.58	6.27	4.52	3.94	2.4	1.1	4.19	0.67	-3.5
D3	7.72	7.66	6.75	6.68	6.04	5.35	4.19	1.72	0.85	4.4	0.18	-8.1
D4	6.91	6.67	5.8	5.5	5.9	4.13	3.94	2.4	1.23	4.19	0.93	-4.85
D5	7.96	7.81	6.96	6.8	6.15	5.49	4.42	2.16	0.8	4.58	0.61	-12.07
D6	6.93	6.85	5.74	5.52	5.27	3.69	3.1	2.69	1.47	3.98	0.99	-22.09
D7	6.58	6.4	5.49	5.27	5.04	3.52	3.25	1.35	0.7	3.48	0.24	-9.73
D8	7.87	7.71	6.87	6.63	5.71	6.23	4.05	2.21	0.78	4.02	0.85	-29.67
D9	9.8	9.59	7.92	7.51	7.5	6.31	5.09	2.84	1.03	5.2	0.37	-20.17
D10	4.4	4.37	3.6	3.61	3.24	0.3	1.13	0.34	0.18	-1.79	0.04	-33.91
D11	6.89	6.7	4.88	5.33	5.39	4.45	2.26	2.13	0.46	-35.64	0.23	-27.42
D12	4.46	4.25	2.4	3.73	2.64	-0.78	1.85	0.29	-0.07	-1.64	-0.7	-113.11
D13	6.64	6.58	4.98	4.85	5.64	2.45	2.55	1.41	-0.23	2.84	-0.44	-21.35
D14	7.11	6.91	6.26	5.99	5.91	4.84	3.36	1.64	0.33	3.21	-0.47	-15.6
D15	5.94	5.83	4.75	4.56	6.11	3.12	2.67	1.65	0.28	2.73	-0.35	-8.24
D16	6.77	6.72	5.82	5.75	5.66	4.24	3.25	1.34	-0.02	2.93	-0.53	-13.23
D17	5.94	5.75	4.75	4.49	6.04	2.66	2.67	1.65	0.34	2.73	-0.61	-8.85
D18	6.92	6.84	6.04	5.93	5.82	4.49	3.47	1.54	-0.12	3.2	-0.55	-14.83
D19	5.62	5.53	4.31	4.16	4.63	1.91	1.99	1.02	-0.55	2.27	-0.24	-28.2
D20	5.83	5.7	4.83	4.68	4.74	2.92	2.67	0.93	-0.2	2.43	-0.68	-15.72
D21	6.98	6.74	6.11	5.82	5.76	5.25	3.08	1.56	0.0	2.99	-0.52	-18.66
D22	7.85	7.74	6.24	5.97	6.01	4.7	4.17	1.0	-0.49	1.64	-0.46	-20.29
D23	4.26	4.28	3.29	3.27	3.0	-0.27	0.21	-0.22	-0.16	-3.4	-0.32	-41.2
D24	6.07	5.94	4.15	4.89	5.05	3.62	2.12	1.48	-0.28	-21.77	-0.76	-49.69
D25	4.25	4.07	2.0	3.39	1.84	-1.06	1.26	-0.28	-0.41	-2.94	-0.96	-110.24

See Table 3.5 for descriptions of the different datasets and Table 3.6 for descriptions of the models.

Table C.3: **Losses Corresponding to Figure 4.3.**

	M^{w}_{gpr}	M^{ws}_{gpr}	M_{gpr}	M^{s}_{gpr}	M_{rf}	M_{svr}	M^{lin}_{svr}	M_{knn}	M^{s}_{svr}	M_{ridge}	M_{wnn}	M_{nn}
D0	3.6	3.61	3.64	3.65	3.66	3.79	3.82	3.75	3.91	3.88	3.96	4.39
D1	3.64	3.65	3.71	3.69	3.7	3.86	3.84	3.82	3.95	3.97	3.99	4.35
D2	3.7	3.71	3.72	3.75	3.76	3.86	3.91	3.83	3.96	3.98	4.05	4.21
D3	3.66	3.66	3.73	3.7	3.71	3.89	3.87	3.82	3.97	3.97	3.99	4.37
D4	3.7	3.71	3.72	3.75	3.77	3.86	3.92	3.83	3.95	3.98	3.99	4.21
D5	3.65	3.66	3.73	3.7	3.71	3.89	3.87	3.82	3.97	3.97	3.98	4.28
D6	3.59	3.59	3.64	3.64	3.65	3.75	3.78	3.74	3.88	3.86	3.93	6.73
D7	3.71	3.72	3.76	3.76	3.77	3.92	3.96	3.86	3.97	4.0	4.01	4.42
D8	3.66	3.67	3.73	3.7	3.71	3.88	3.84	3.84	3.97	3.99	3.97	4.35
D9	3.48	3.49	3.56	3.57	3.59	3.74	3.72	3.69	3.85	3.83	3.84	4.36
D10	3.57	3.57	3.62	3.6	3.6	3.73	3.8	3.79	3.71	3.8	3.74	5.19
D11	3.71	3.72	3.75	3.81	3.78	3.89	3.95	5.11	3.99	4.07	4.02	5.32
D12	3.57	3.58	3.65	3.65	3.6	3.76	3.86	3.78	3.73	3.77	3.76	9.03
D13	3.62	3.62	3.66	3.7	3.71	3.87	3.89	3.79	3.91	3.93	3.95	5.29
D14	3.72	3.73	3.76	3.76	3.77	3.93	3.94	3.88	3.99	4.03	4.04	4.76
D15	3.77	3.78	3.76	3.82	3.83	3.93	4.01	3.9	4.0	4.06	4.01	4.5
D16	3.73	3.74	3.76	3.77	3.78	3.96	3.96	3.89	4.01	4.03	4.06	4.82
D17	3.77	3.78	3.75	3.82	3.83	3.93	4.02	3.9	3.99	4.06	4.06	4.47
D18	3.73	3.74	3.76	3.77	3.77	3.95	3.95	3.88	4.02	4.03	4.05	4.64
D19	3.58	3.59	3.64	3.63	3.64	3.77	3.82	3.71	3.8	3.84	3.81	5.33
D20	3.78	3.79	3.81	3.82	3.83	3.97	4.03	3.91	4.02	4.06	4.05	4.7
D21	3.73	3.74	3.77	3.77	3.78	3.95	3.93	3.9	4.01	4.06	4.08	4.8
D22	3.54	3.55	3.58	3.59	3.6	3.77	3.76	3.72	3.82	3.82	3.84	4.67
D23	3.62	3.62	3.69	3.66	3.66	3.81	3.86	3.89	3.78	3.89	3.82	5.36
D24	3.78	3.78	3.8	3.85	3.82	3.95	4.01	4.72	4.02	4.11	4.06	6.05
D25	3.62	3.63	3.71	3.71	3.65	3.82	3.92	3.87	3.79	3.85	3.82	7.42

See Table 3.5 for descriptions of the different datasets and Table 3.6 for descriptions of the models. Here all values are multiplied by 10 so that they are the same order of magnitude as in Table C.1.

Table C.4: **Percentage Improvements Corresponding to Figure 4.4.**

	M^{w}_{gpr}	M^{ws}_{gpr}	M_{gpr}	M^{s}_{gpr}	M_{rf}	M_{svr}	M^{lin}_{svr}	M_{knn}	M^{s}_{svr}	M_{ridge}	M_{wnn}	M_{nn}
D0	17.88	17.78	16.99	16.68	16.49	13.55	12.95	14.43	10.84	11.52	9.69	-0.07
D1	18.03	17.8	16.42	17.07	16.72	13.07	13.6	14.12	11.19	10.73	10.18	2.08
D2	16.75	16.61	16.33	15.54	15.32	13.09	11.97	13.78	10.83	10.5	8.9	5.33
D3	17.66	17.58	16.1	16.66	16.58	12.36	12.93	13.95	10.69	10.7	10.26	1.65
D4	16.75	16.48	16.28	15.54	15.2	13.09	11.76	13.78	11.05	10.5	10.32	5.33
D5	17.86	17.67	16.08	16.81	16.6	12.5	13.0	14.1	10.67	10.7	10.38	3.73
D6	17.32	17.24	16.11	15.98	15.76	13.6	12.78	13.88	10.5	11.01	9.37	-55.07
D7	16.42	16.22	15.37	15.31	15.07	11.9	10.8	13.25	10.57	9.9	9.79	0.6
D8	17.71	17.52	16.18	16.81	16.57	12.63	13.63	13.61	10.65	10.2	10.76	2.18
D9	18.86	18.58	16.85	16.73	16.29	12.62	13.3	13.94	10.16	10.65	10.38	-1.82
D10	12.88	12.82	11.64	12.18	12.19	8.92	7.33	7.63	9.36	7.18	8.73	-26.52
D11	16.59	16.35	15.61	14.4	15.07	12.45	11.24	-14.89	10.34	8.41	9.63	-19.68
D12	12.97	12.74	10.85	11.0	12.21	8.37	5.92	7.68	9.1	7.91	8.27	-120.36
D13	16.46	16.43	15.45	14.62	14.49	10.68	10.22	12.57	9.73	9.32	8.87	-21.97
D14	16.61	16.39	15.81	15.86	15.57	11.87	11.8	13.04	10.57	9.66	9.44	-6.64
D15	15.44	15.34	15.78	14.35	14.18	11.9	10.16	12.6	10.38	9.1	10.15	-0.91
D16	16.31	16.25	15.79	15.47	15.4	11.38	11.19	12.91	10.16	9.63	9.01	-7.94
D17	15.44	15.25	15.87	14.35	14.1	11.9	9.87	12.6	10.51	9.11	9.11	-0.21
D18	16.4	16.31	15.69	15.62	15.5	11.4	11.5	13.05	10.03	9.65	9.23	-3.86
D19	14.17	14.09	12.81	13.0	12.87	9.68	8.5	11.07	8.96	7.99	8.64	-27.62
D20	15.24	15.1	14.55	14.34	14.19	11.01	9.65	12.36	9.93	8.97	9.21	-5.31
D21	16.4	16.16	15.58	15.64	15.36	11.58	11.97	12.54	10.21	9.02	8.69	-7.41
D22	15.78	15.63	14.74	14.54	14.26	10.33	10.42	11.44	9.13	9.13	8.59	-11.07
D23	12.96	12.99	11.43	12.14	12.15	8.53	7.19	6.6	9.18	6.5	8.11	-28.8
D24	15.42	15.27	14.78	13.78	14.33	11.43	10.09	-5.81	9.94	7.83	8.99	-35.46
D25	12.97	12.76	10.79	10.91	12.21	8.21	5.82	7.07	8.94	7.43	8.11	-78.33

See Table 3.5 for descriptions of the different datasets and Table 3.6 for descriptions of the models.