Project Milestone - STATS 315B
Using Machine Learning to Predict HbA1c and Analyze Glucose Management Strategies in Pediatric Type 1 Diabetes

Nicolas Fertout, Thomas Bougnon
Department of Statistics, Department of Aeronautics and Astronautics
Stanford University
nfertout@stanford.edu, thomasbn@stanford.edu

Abstract

We compared State-of-the-art machine learning and Deep Learning techniques, including OLS and Lasso, Random Forest, XGBoost and 1D-CNN to predict HbA1c levels using CGM and patient data. Of the Linear Models, Lasso achieved the best performance, with an average error (MAE) of 0.34. Using more complex ML algorithms further improved results. XGBoost surpassed all other models in terms of performance, with 99.10% of predictions within 1 percent point and 83.26% within 0.5 percent points of the true HbA1c value. However, Deep Learning models under-performed, due to a small sample size and a lack of structure in the data. Overall, these findings demonstrate the effectiveness of ML methods in predicting HbA1c measurements using CGM and patient data.

1 Introduction

Type-1 diabetes is a prevalent chronic disease affecting millions of individuals worldwide, primarily diagnosed in childhood or adolescence. It is characterized by the immune system mistakenly attacking and destroying the insulin-producing cells in the pancreas, resulting is the inability to properly regulate blood glucose levels. Effective management of diabetes is crucial to prevent complications and improve patient outcomes. Hemoglobin A1c (HbA1c) is a widely used measure that reflects average blood glucose levels over a specific period, providing valuable information about a patient's glycemic control. Accurate prediction of HbA1c levels can help healthcare providers tailor treatment plans, monitor progress, and make informed decisions regarding type-1 diabetes management.

This project takes place in the context of a study conducted by the Stanford Diabetes Research Center, specifically related to type-1 diabetes, a type of diabetes primarily diagnosed in childhood or adolescence and characterized by the immune system mistakenly attacking and destroying the insulin-producing cells in the pancreas. The aim of this study is to utilize advanced machine learning techniques to predict hemoglobin A1c (HbA1c) levels using continuous glucose monitoring (CGM) data and patient-specific information.

CGM devices were worn by patients, collecting real-time glucose data at 5-minute intervals for a duration of one year. Simultaneously, multiple hemoglobin A1c (HbA1c) measurements were taken to evaluate average blood sugar levels over a longer period. HbA1c serves as an essential indicator of glycemic control and provides insights into the effectiveness of diabetes management strategies.

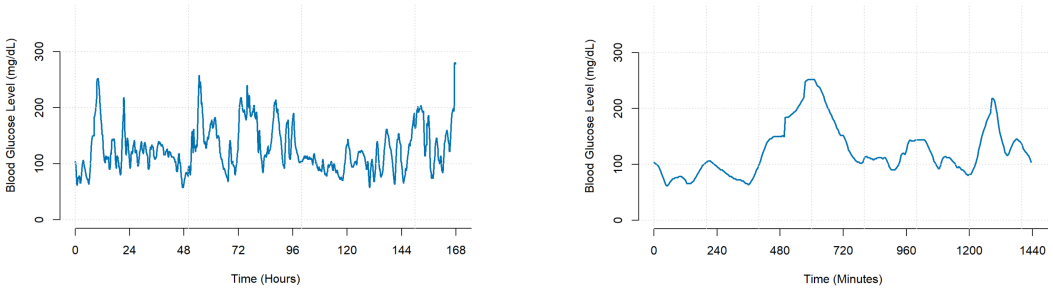
The primary objective of this Machine Learning (ML) project is to leverage supervised and unsupervised to replicate and extend a similar study by Grossman et al. [2021] to explore the associations

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

between HbA1c and CGM data, as well as to demonstrate the benefits of employing ML models in this context.

2 Dataset

To conduct our analysis, we required a comprehensive dataset with ample HbA1c measurements, CGM data, and patient characteristics (such as weight, age or ethnicity). To meet these requirements, we selected publicly available data from the research paper titled "Continuous glucose monitoring and intensive treatment of type 1 diabetes". This dataset included five HbA1c measurements per patient, taken at the beginning of monitoring and every 13 weeks thereafter over a one-year monitoring period, along with the corresponding CGM data. In Figure 1 we plot an example of a CGM time series. Notice the lack of structure in the data – very noisy, with no clear periodicity.



(a) CGM Time Series - One Week

(b) CGM Time Series - One Day

Figure 1: An example of a CGM Time Series over different time intervals.

In order to structure the dataset for analysis, each HbA1c measurement was represented as a separate observation, except for the initial measurement taken at the beginning of monitoring. For each row representing an HbA1c measurement, the dataset includes the corresponding glucose data collected during a period of two months leading up to the measurement. As some readings did not have enough associated CGM data, or other patients stopped the experiment earlier than expected, we ended up with a dataset consisting of 886 HbA1c measurements from 230 patients with type-1 diabetes.

We first structured our data as followed, to get a first dataset $X_{\text{time\_series}}$. Note that each patient ID is associated with several HbA1c measures.

Table 1: $X_{\text{time\_series}}$, dataset with time series.

Patient ID	Glucose Time Series	HbA1c Measurement
1	163, 197, 225...	8.3
1	120, 159, 181...	6.3
1	223, 203, 181...	9.4
1	142, 173, 201...	8.8
2	236, 218, 195...	7.1
...	...	...

Feature Engineering

Now, when it comes to non Deep Learning methods, a raw time series would not represent an appropriate input; most of the traditional ML algorithm require feature extraction at first. We aimed to both replicate and enhance the CGM statistics used in the original paper.

We replicated the following CGM statistics from Grossman et al.: Mean, Standard Deviation (SD), Coefficient of Variation (CV), and the percentage of time spent in various glycemic ranges: Hypo-

glycemia [<54 mg/dL], Clinical Hypoglycemia [54–69], Target Range [70–180], Conservative Target Range [70–140], Above Target Range [181–250], and Far Above Target Range [>250].

In addition to the replicated statistics, we introduced new CGM statistics: Minimum, Maximum, Median, Mean Absolute Deviation (MAD).
Furthermore, we computed statistics related to the Rate of Change (ROC) of CGM data, from which we derived the following metrics: Mean ROC, SD ROC, Minimum ROC, Maximum ROC, Median ROC, MAD ROC.

We end up with a second dataset, $X_{\text{statistics}}$, containing 30 features (see Table 2).

Table 2: $X_{\text{statistics}}$, dataset with CGM statistics.

Patient ID	Patient Characteristics	Glucose Characteristics	HbA1c
1	sex, age, weight...	CGM statistics (mean, SD, ...)	8.3
1	sex, age, weight...	CGM statistics (mean, SD, ...)	6.3
1	sex, age, weight...	CGM statistics (mean, SD, ...)	9.4
1	sex, age, weight...	CGM statistics (mean, SD, ...)	8.8
2	sex, age, weight...	CGM statistics (mean, SD, ...)	7.1
..	...	...	...

Figure 2 summarizes the feature engineering and data formatting process.

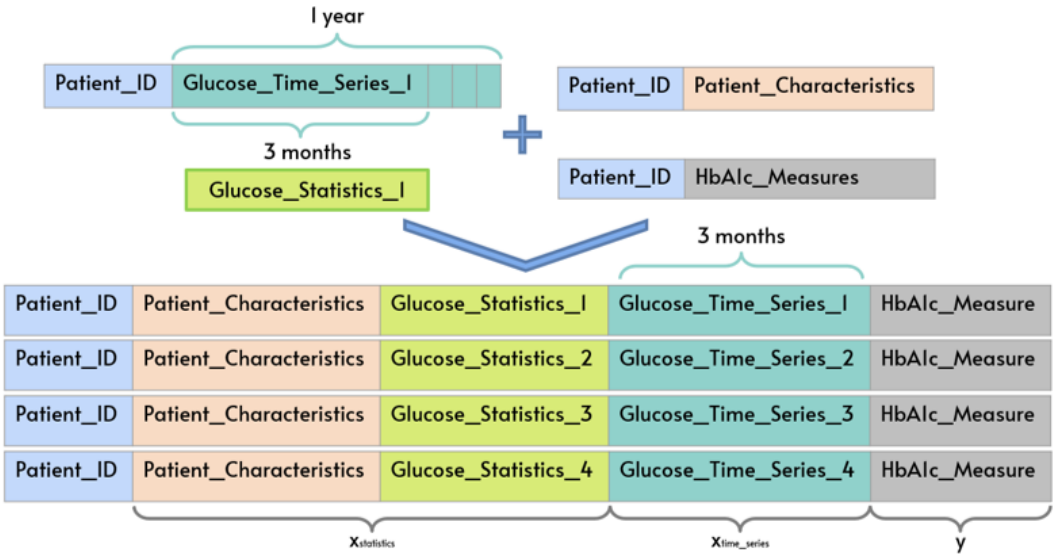


Figure 2: Dataset structure for one patient

In the Results section, we analyzed three datasets: one with only replicated statistics ($X_{\text{replicated\_statistics}}$), one with all CGM statistics ($X_{\text{statistics}}$) and one with time series data ($X_{\text{time\_series}}$). This allowed us to assess the performance of our models using different data configurations, and to see whether our new features led to improvements.

3 Methodology

3.1 Data split

Before running our models, we first need to define appropriate training and test set. For all three datasets defined below, we used a 75:25 ratio. This resulted in a training set of size 665 and a test set

of size 221. It is worth mentioning that the same data points were used as training and test set for every model.

3.2 Models

We implemented and evaluated three classes of ML models: Linear, Non-Linear and Deep Learning. The first two use both $X_{\text{replicated\_statistics}}$ and $X_{\text{statistics}}$, while the latter uses $X_{\text{time\_series}}$.

3.2.1 Linear models

To align with the methodology of Grossman et al., Linear Regression (OLS) was first implemented, providing a baseline model to assess achievable performance with minimal effort.
We then added regularization, with the use of Ridge [Hoerl and Kennard, 1970] and Lasso [Tibshirani, 1996]. Ridge introduces an L2 penalty to the Least Squares objective, creating bias to prevent overfitting. Lasso performs feature selection by introducing an L1 penalty, allowing us to prioritize important variables.

3.2.2 Non-linear models

In order to capture more complex relationships, we then implemented some non-linear models. The simplest one, Decision Tree (DT) offers interpretable insights through each splits. Then, Random Forest (RF) [Breiman, 2001] takes this concept a step further by constructing an ensemble of decision trees, and incorporating decorrelation techniques, using a different subset of the training data and a random subset of features each time.
Some Boosting methods were also considered. AdaBoost [Schapire, 1999] sequentially trains weak learners and assign higher weights to samples with higher regression error. Gradient Boosting, sequentially trains weak trees with each subsequent tree aiming to correct the mistakes made by the previous ones. Finally, XGBoost (eXtreme Gradient Boosting) [Chen and Guestrin, 2016] provides an optimized implementation of Gradient Boosting, to enhance model performance through parallel computing and regularization techniques.

3.2.3 Deep Learning models

So far, all our models considered only the hand-crafted CGM statistics. We thus tried to implement models that will take advantage of the temporal structure of our time series. As illustrated in Figure 3, we hope that using Deep Learning will automate the feature extraction process.

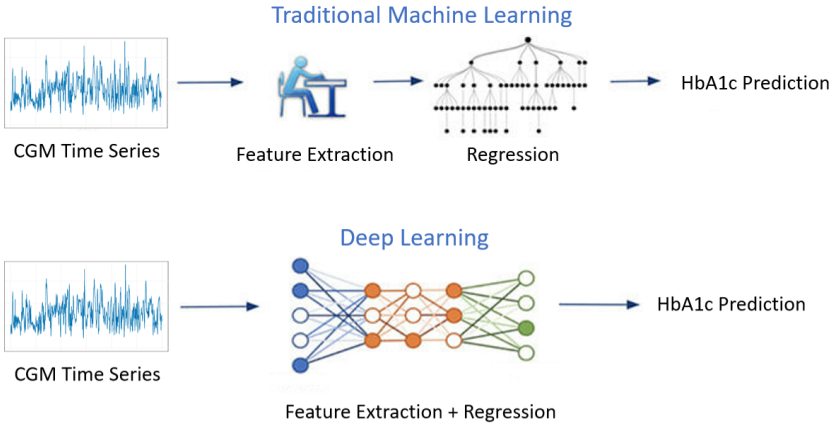


Figure 3: Deep Learning automatically performs features extraction.

To effectively capture short term dependencies in sequential data, we used 1-D Convolutional Neural Network (CNN), which uses a mix of convolutional, pooling and fully connected layers. Then, we used Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN) known for capturing long term dependencies.

Trying to benefit from the above models and their ability to comprehend the underlying structure of time series data, we also used Transfer Learning approaches, extracting the representation of CGM data through the weights of the last layer of 1D CNN, and feeding them into a RF (along with patient characteristics). Similarly, we also trained an Auto-Encoder to get a latent representation of our CGM time series, and then use it as input in a RF (combined with patient characteristics).

3.3 Evaluation metrics

In order to validate and test our models, and in line with Grossman et al., let us define useful metrics. Let Y_i and $\hat{Y}_i$ be respectively the true and predicted HbA1c value of the i^{th} observation, and let N be the size of the set we are considering.

- Mean Absolute Error (MAE) $= \frac{1}{N}\sum_{i=1}^{N}|\hat{Y}_i - Y_i|$
- Within 1 Accuracy $= \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(|\hat{Y}_i - Y_i| \leq 1)$
- Within 0.5 Accuracy $= \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(|\hat{Y}_i - Y_i| \leq 0.5)$

3.4 Hyperparameter tuning and Cross-Validation

Using the above evaluation metrics, we then used k-fold Cross-Validation (CV) on our training set to tune our hyperparameters (k varying between 5 and 10 depending on the model used). For Ridge and Lasso, we used the built-in function `cv.glmnet` from the `glmnet` package to get the optimal choice of λ. As seen on Figure 4, the optimal Lasso uses only 12/30 parameters.

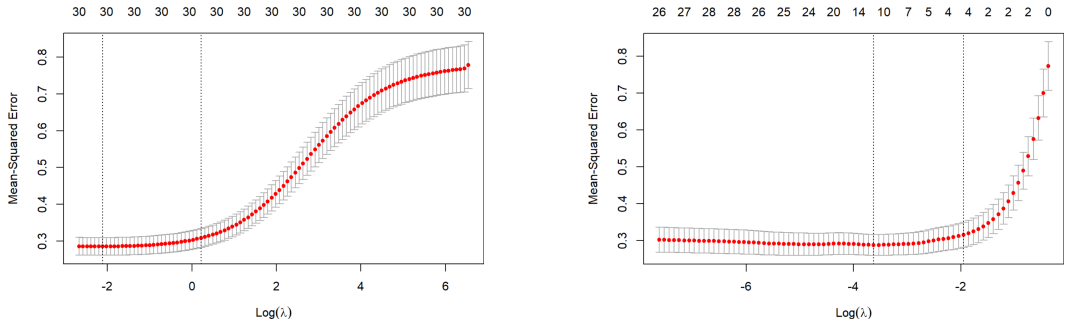


Figure 4: Cross-Validation Plots for Ridge (left) and Lasso (right)

For other models, the `GridSearchCV` function from `sklearn.model_selection` allowed us to use k-fold CV on several hyperparameters, such as Max Tree Depth or Minimum Samples per Leaf. For instance, for RF those optimal parameters were 7 and 29 respectively.

For Deep Learning models, multiple architectures were implemented, along with varying performance improving layers such as Batch Normalization (BN) or Dropout. For 1-D CNN, the best-performing architecture (which we will use in the Result section) is displayed in Figure 5 below.

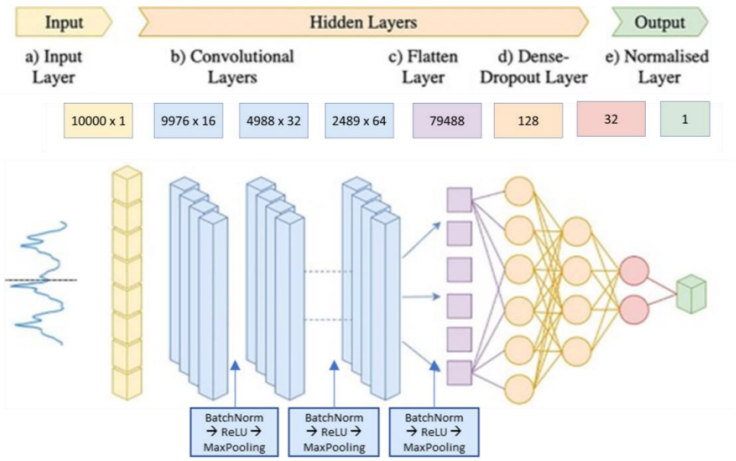


Figure 5: Best 1-D CNN Architecture

4 Results

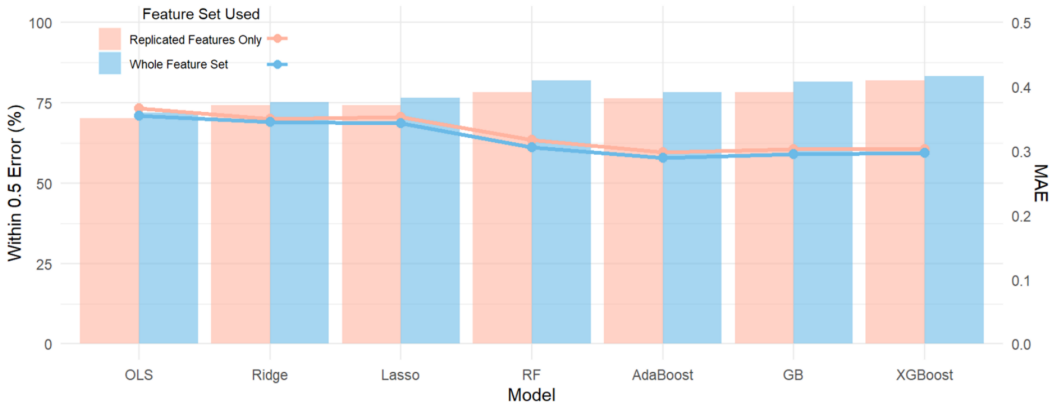


Figure 6: The effect of additional features on Non Deep Learning models performance

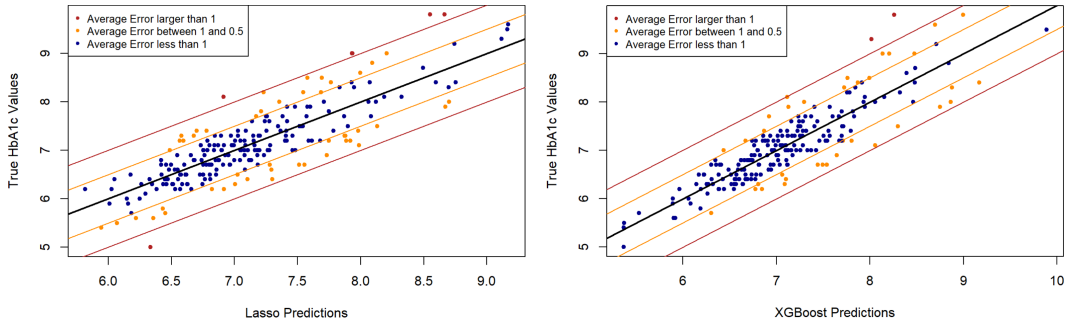


Figure 7: True HbA1c against Prediction made by Lasso (left) and XGBoost (right), with Within 1 and 0.5 Intervals.

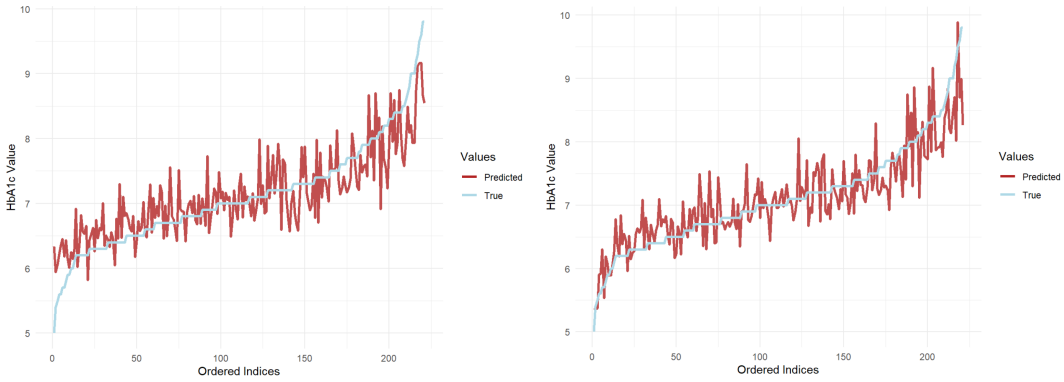


Figure 8: True Ordered HbA1c against Prediction made by Lasso (left) and XGBoost (right)

Table 3: Performance metrics using $X_{\text{statistics}}$ – *Italic*: Best Linear; **Bold**: Best Non-Linear.

Model Category	Model	MAE	Within 1 Accuracy	Within 0.5 Accuracy
NA	Null	0.613	82.81%	51.58%
Linear Models	OLS	0.355	97.29%	71.95%
	Ridge	0.345	*98.19%*	75.11%
	Lasso: Optimal λ	*0.343*	97.74%	*75.57%*
	Lasso: One-SE λ	0.359	97.74%	73.76%
Non-Linear Models	Decision Tree	0.375	96.83%	69.68%
	Random Forest	0.306	98.64%	81.90%
	AdaBoost	**0.289**	98.64%	78.28%
	Gradient Boosting	0.295	96.83%	81.45%
	XGBoost	0.297	**99.10%**	**83.26%**

Table 4: Performance metrics of Deep Learning models using $X_{\text{time\_series}}$.

Model	Dataset	MAE
Null Model	CGM and HbA1c	0.613
1D-CNN	CGM and HbA1c	0.552
LSTM	CGM and HbA1c	0.591
Transfer Learning (with RF)	CGM and HbA1c	0.856
Auto-Encoder (with RF)	CGM and HbA1c	0.910
Null Model	ECG	1.492
1D-CNN	ECG (train: 7500)	0.121
LSTM	ECG (train: 7500)	0.157
1D-CNN	ECG (train: 665)	0.235
LSTM	ECG (train: 665)	0.322

5 Discussion

While analysing the results, it is important to keep in mind the metrics from the Null (intercept-only) model, which assumes a constant prediction for all test samples, and serves as a reference on our dataset.

As a first, general insight, our additional features (i.e. the one present in $X_{\text{statistics}} - X_{\text{replicated\_statistics}}$) helped improve the predictive power of every model. Indeed, as we can see on Figure 6, using the whole feature set led to overall higher Within 0.5 Accuracy and lower MAE. From now on, we will consider

Zooming into Linear Models (see Table 3), both Lasso and Ridge performed well. The latter obtained the highest Within 1 Accuracy, while the former obtained the highest Within 0.5 Accuracy and lowest MAE, among linear models. Clearly, adding regularization to OLS improved results.
It is worth mentioning that the Optimal Lasso uses 12/30 features, while a λ one Standard Error (SE) away from the optimal λ gave close performance metrics while using only 4 variables:
Mean Glucose, Proportion in Target Range, Proportion Far Above Target Range, Median Glucose.

Those findings are confirmed with the help of the Decision Tree (DT). Although the DT clearly under-performed, the associated tree provided valuable insights regarding the most useful features for splitting patients into heterogeneous HbA1c groups: Proportion in Target Range, Mean Glucose and Proportion in Conservative Target Range. These were all part of the original statistics we replicated, demonstrating their suitability for this task.

Now, the best performing model seems to be XGBoost. Even though it was beaten by AdaBoost in terms of MAE (0.297 for XGBoost against 0.289 for AdaBoost), XGBoost surpassed all other models in terms of both Within 1 and Within 0.5 Accuracy (99.10% and 83.26% respectively).In particular, it outperformed RF, which was the best performing model in the original paper.
These accuracies can be visualized in Figure 7, in which we see, for instance, that only 2 data points were off by more than 1, as opposed to 5 for Lasso.
Furthermore, Figure 8 conveys another insight: while some variations remain, XGBoost seems well suited (and in particular, better than Lasso) for predicting extreme values, both low and high.

When it comes to Deep Learning, all approaches were disappointing. The best model, 1-D CNN, barely overcame the Intercept Only model (MAE: 0.552, see Table 4). To validate our models and hypothesize on the cause of these performance, we ran the exact same models (with same parameters and architecture) on a simulated ECG dataset set to predict Heart Rate. Using a sample size of 7,500 for the training set (2,500 for the test set) led to strong results (MAE of 0.121 for 1-D CNN and 0.157 for LSTM). Moreover, with an ECG training set of the same size as our CGM training set (i.e. 665), our models still performed very well (0.235 and 0.322 resp.).
Thus, given those results, the poor performance of Deep Learning methods seems to be caused by a too small sample size, but most importantly by a lack of structure in the CGM data, as our model worked well on the very structured ECG data and managed to still get reasonable predictions for smaller sample sizes.

In conclusion, this project demonstrated the effectiveness of machine learning techniques in predicting HbA1c levels in pediatric type-1 diabetes. Incorporating additional statistics and fine-tuning models resulted in significant improvements. The superior performance of XGBoost and Adaboost highlights their potential as valuable tools for accurate and reliable predictions of HbA1c levels, improving diabetes management and patient outcomes.

6 Next Steps

Given the relatively small size of our dataset, it would be advisable to evaluate our models on a new, holdout dataset, to ensure that our results generalize well.

Despite the challenges associated with deep learning approaches, it may be worth considering certain denoising methods as an alternative approach to enhance their effectiveness.

To go further, we aim to identify the main differences in patient glucose management based on when/if the patient started using an insulin pump and a closed-loop system.

Finally, we also intend to learn indicators that can predict when to recommend support from the care team or when the care team should be alerted about a patient's condition. Comparing the results of those indicators with the GRI (Glycemia Risk index) (see DC et al. [2022]) to see if this new metric can accurately predict patient condition.

References

Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785.

Klonoff DC, Wang J, Rodbard D, Kohn MA, Li C, Liepmann D, Kerr D, Ahn D, Peters AL, Umpierrez GE, Seley JJ, Xu NY, Nguyen KT, Simonson G, Agus MSD, Al-Sofiani ME, Armaiz-Pena G, Bailey TS, Basu A, Battelino T, Bekele SY, Benhamou PY, Bequette BW, Blevins T, Breton MD, Castle JR, Chase JG, Chen KY, Choudhary P, Clements MA, Close KL, Cook CB, Danne T, Doyle FJ 3rd, Drincic A, Dungan KM, Edelman SV, Ejskjaer N, Espinoza JC, Fleming GA, Forlenza GP, Freckmann G, Galindo RJ, Gomez AM, Gutow HA, Heinemann L, Hirsch IB, Hoang TD, Hovorka R, Jendle JH, Ji L, Joshi SR, Joubert M, Koliwad SK, Lal RA, Lansang MC, Lee WA, Leelarathna L, Leiter LA, Lind M, Litchman ML, Mader JK, Mahoney KM, Mankovsky B, Masharani U, Mathioudakis NN, Mayorov A, Messler J, Miller JD, Mohan V, Nichols JH, Nørgaard K, O'Neal DN, Pasquel FJ, Philis-Tsimikas A, Pieber T, Phillip M, Polonsky WH, Pop-Busui R, Rayman G, Rhee EJ, Russell SJ, Shah VN, Sherr JL, Sode K, Spanakis EK, Wake DJ, Waki K, Wallia A, Weinberg ME, Wolpert H, Wright EE, Zilbermint M, and Kovatchev B. A glycemia risk index (gri) of hypoglycemia and hyperglycemia for continuous glucose monitoring validated by clinician ratings. *Journal of Diabetes Science and Technology*, 2022. doi: 10.1177/19322968221085273. URL https://pubmed.ncbi.nlm.nih.gov/35348391.

Joshua Grossman, Andrew Ward, Jamie L. Crandell, Priya Prahalad, David M. Maahs, and David Scheinker. Improved individual and population-level hba1c estimation using cgm data and patient characteristics. *Journal of Diabetes and its Complications*, 35(8):107950, 2021. ISSN 1056-8727. doi: https://doi.org/10.1016/j.jdiacomp.2021.107950. URL https://www.sciencedirect.com/science/article/pii/S1056872721001379.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. ISSN 00401706. URL http://www.jstor.org/stable/1267351.

Robert E. Schapire. A brief introduction to boosting. *IJCAI International Joint Conference on Artificial Intelligence*, 2:1401–1406, 1999. ISSN 1045-0823.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x.