GlucoBench: Curated List of Continuous Glucose Monitoring Datasets with Prediction Benchmarks

Renat Sergazinov [1,*]
ElizabethChun[1], Valeriya Rogovchenko [1], Nathaniel Fernandes [2], Nicholas Kasman [2], Irina Gaynanova [1*]
[1] Department of Statistics, Texas A&M University
[2] Department of Electrical and Computer Engineering, Texas A&M University

***Abstract*—The rising rates of diabetes necessitate innovative methods for its management. Continuous glucose monitors (CGM) are small medical devices that measure blood glucose levels at regular intervals providing insights into daily patterns of glucose variation. Forecasting of glucose trajectories based on CGM data holds the potential to substantially improve diabetes management, by both refining artificial pancreas systems and enabling individuals to make adjustments based on predictions to maintain optimal glycemic range. Despite numerous methods proposed for CGMbased glucose trajectory prediction, these methods are typically evaluated on small, private datasets, impeding reproducibility, further research, and practical adoption. The absence of standardized prediction tasks and systematic comparisons between methods has led to uncoordinated research efforts, obstructing the identification of optimal tools for tackling specific challenges. As a result, only a limited number of prediction methods have been implemented in clinical practice. To address these challenges, we present a comprehensive resource that provides (1) a consolidated repository of curated publicly available CGM datasets to foster reproducibility and accessibility; (2) a standardized task list to unify research objectives and facilitate coordinated efforts; (3) a set of benchmark models with established baseline performance, enabling the research community to objectively gauge new methods' efficacy; and (4) a detailed analysis of performance-influencing factors for model development. We anticipate these resources to propel collaborative research endeavors in the critical domain of CGM-based glucose predictions. Our code is available online at github.com/IrinaStatsLab/GlucoBench.**

1 Introduction

According to the International Diabetes Federation, 463 million adults worldwide have diabetes with 34.2 million people affected in the United States alone (IDF, 2021). Diabetes is a leading cause of heart disease (Nanayakkara et al., 2021), blindness (Wykoff et al., 2021), and kidney disease (Alicic et al. 2017). Glucose management is a critical component of diabetes care, however achieving target glucose levels is difficult due to multiple factors that affect glucose fluctuations, e.g., diet, exercise, stress, medications, and individual physiological variations.

Continuous glucose monitors (CGM) are medical devices that measure blood glucose levels at frequent intervals, often with a granularity of approximately one minute. CGMs have great potential to improve diabetes management by furnishing real-time feedback to patients and by enabling an autonomous artificial pancreas (AP) system when paired with an insulin pump (Contreras & Vehi, 2018, Kim & Yoon, 2020). Figure 1 illustrates an example of a CGM-human feedback loop in a recommender setting. The full realization of CGM potential, however, requires accurate glucose prediction models. Although numerous prediction models (Fox et al., 2018, Armandpour et al., 2021, Sergazinov et al., 2023) have been proposed, only simple physiological (Bergman et al., 1979, Hovorka et al., 2004) or statistical (Oviedo et al., 2017; Mirshekarian et al., 2019; Xie & Wang, 2020) models are utilized within current CGM and AP software. The absence of systematic model evaluation protocols and established benchmark datasets hinder the analysis of more complex models' risks and benefits, leading to their limited practical adoption (Mirshekarian et al., 2019).

In response, we present a curated list of five public CGM datasets and a systematic protocol for models' evaluation and benchmarking. The selected datasets have varying sizes and demographic characteristics, while the developed systematic data-preprocessing pipeline facilitates the inclusion of additional datasets. We propose two tasks: (1) enhancing the predictive accuracy; (2) improving the uncertainty quantification (distributional fit) associated with predictions. In line with previous works (Mirshekarian et al., 2019; Xie & Wang, 2020), we measure the performance on the first task with mean squared error (MSE) and mean absolute error (MAE), and on the second task with likelihood and expected calibration error (ECE) (Kuleshov et al., 2018). For each task, we train and evaluate a set of baseline models. From data-driven models, we select linear regression and ARIMA to represent shallow baselines, and Transformer (Vaswani et al., 2017), NHiTS (Challu et al., 2023), TFT (Lim et al. 2021), and Gluformer (Sergazinov et al., 2023) to represent deep learning baselines. We select the Latent ODE (Rubanova et al., 2019) to represent a hybrid data-driven / physiological model.

Our work contributes a curated collection of diverse CGM datasets, formulation of two tasks focused on model accuracy and uncertainty quantification, an efficient benchmarking protocol, evaluation of a range of baseline models including shallow, deep and hybrid methods, and a detailed analysis of performance-influencing factors for model optimization.

*Address correspondence to: mrsergazinov@tamu.edu, irinag@tamu.edu

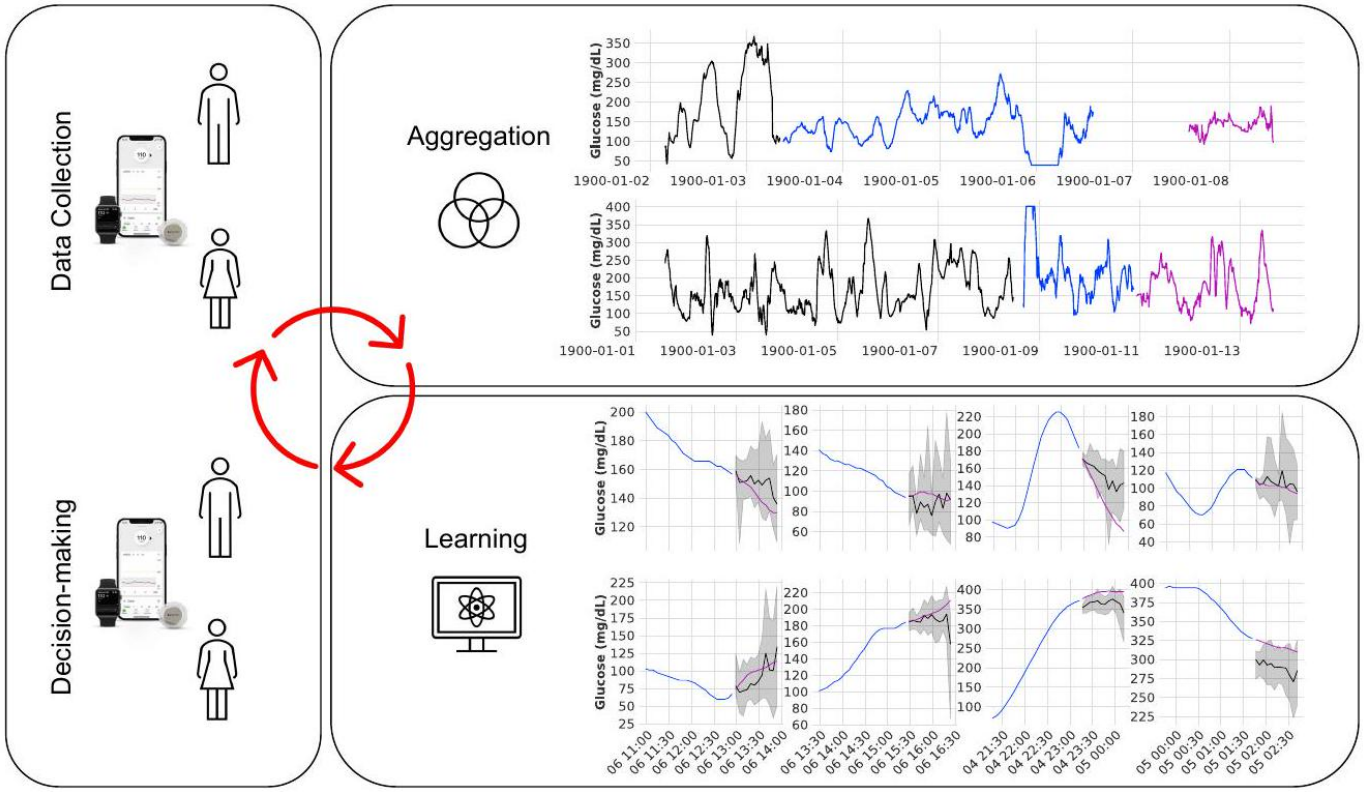

Figure 1: Sample of glucose curves captured by the Dexcom G4 Continuous Glucose Monitoring (CGM) system, with dates de-identified for privacy(Weinstock et al., 2016).

Table 1: Summary of the glucose prediction models by dataset and model type. We indicate "open" for datasets that are publicly available online, "simulation" for the ones based on simulated data, "proprietary" for the ones that cannot be released. We indicate deep learning models by "deep", non-deep learning models by "shallow", and physiological models by "physiological." We provide full table with references to all the works in Appendix A.

Type	Diabetes	# of datasets	# of deep	# of shallow	# of Physiological
Open	Type 1	9	13	3	2
Simulation	Type 1	12	3	3	6
Proprietary	Mixed	22	7	8	7

2 Related works

An extensive review of glucose prediction models and their utility is provided by Oviedo et al. (2017); Contreras & Vehi (2018); Kim & Yoon (2020); Kavakiotis et al. (2017). Following Oviedo et al. (2017), we categorize prediction models as physiological, data-driven, or hybrid. Physiological models rely on the mathematical formulation of the dynamics of insulin and glucose metabolism via differential equations (Man et al., 2014; Lehmann & Deutsch, 1991). A significant limitation of these models is the necessity to predefine numerous parameters. Data-driven models rely solely on the CGM data (and additional covariates when available) to characterize glucose trajectory without incorporating physiological dynamics. These models can be further subdivided into shallow (e.g. linear regression, ARIMA, random forest, etc.) and deep learning models (e.g. recurrent neural network models, Transformer, etc.). Data-driven models, despite their capacity to capture complex

patterns, may suffer from overfitting and lack interpretability. Hybrid models use physiological models as a preprocessing or data augmentation tool for data-driven models. Hybrid models enhance the flexibility of physiological models and facilitate the fitting process, albeit at the expense of diminished interpretability. Table 1 summarizes existing models and datasets, indicating model type.

Limitations. The present state of the field is characterized by several key constraints, including (1) an absence of well-defined benchmark datasets and studies, (2) a dearth of open-source code bases, and (3) omission of Type 2 diabetes from most open CGM studies. To address the second limitation, two benchmark studies have been undertaken to assess the predictive performance of various models (Mirshekarian et al., 2019; Xie & Wang, 2020). Nonetheless, these studies only evaluated the models on one dataset (Marling & Bunescu, 2020), comprising a limited sample of 5 patients with Type 1

diabetes, and failed to provide source code. We emphasize that, among the 45 methods identified in Table 1, a staggering 38 works do not offer publicly available implementations. For the limitation (3), it is important to recognize that Type 2 is more easily managed through lifestyle change and oral medications than Type 1 which requires lifelong insulin therapy.

3 DATA

3.1 Description

We have selected five publicly available CGM datasets: Broll et al. (2021); Colás et al. (2019); Dubosson et al. (2018); Hall et al. (2018); Weinstock et al. (2016).

To ensure data quality, we used the following set of criteria. First, we included a variety of dataset sizes and verified that each dataset has measurements on at least 5 subjects and that the collection includes a variety of sizes ranging from only five (Broll et al., 2021) to over 200 (Colás et al., 2019; Weinstock et al., 2016) patients. On the patient level, we ensured that each subject has non-missing CGM measurements for at least 16 consecutive hours. At the CGM curve level, we have verified that measurements fall within a clinically relevant range of $20\mathrm{mg/dL}$ to $400\mathrm{mg/dL}$, avoiding drastic fluctuations exceeding $40\mathrm{mg/dL}$ within a 5 -minute interval, and ensuring non-constant values.

Finally, we ensured that the collection covers distinct population groups representing subjects with Type 1 diabetes (Dubosson et al., 2018, Weinstock et al., 2016), Type 2 diabetes (Broll et al., 2021), or a mix of Type 2 and none (Colás et al., 2019; Hall et al., 2018). We expect that the difficulty of accurate predictions will depend on the population group: patients with Type 1 have significantly larger and more frequent glucose fluctuatons. Table 2 summarizes all five datasets with associated demographic information, where some subjects are removed due to data quality issues as a result of pre-processing (Section 3.2). We describe data availability in Appendix A.

Covariates. In addition to CGM data, each dataset has covariates (features), which we categorize based on their temporal structure and input type. The temporal structure distinguishes covariates as static (e.g. gender), dynamic known (e.g. hour, minute), and dynamic unknown (e.g. heart beat, blood pressure). Furthermore, input types define covariates as either real-valued (e.g. age) or ordinal (e.g. education level) and categorical or unordered (e.g. gender) variables. We illustrate different types of temporal variables in Figure 2. We summarize covariate types for each dataset in Appendix A.

3.2 Pre-Processing

We pre-process the raw CGM data via interpolation and segmentation, encoding categorical features, data partitioning, scaling, and division into input-output pairs for training a predictive model.

Interpolation and segmentation. To put glucose values on a uniform temporal grid, we identify gaps in each subject's trajectory due to missing measurements. When the gap length is less than a predetermined threshold (Table 3), we impute the missing values by linear interpolation. When the gap length exceeds the threshold, we break the trajectory into several continuous segments. Green squares in Figure 3 indicate gaps that will be interpolated, whereas red arrows indicate larger gaps where the data will be broken into separate segments. In Dubosson et al. (2018) dataset, we also interpolate missing values in dynamic covariates (e.g., heart rate). Thus, for each dataset we obtain a list of CGM sequences $\mathcal{D} = \left\{\mathbf{x}_j^{(i)}\right\}_{i,j}$ with i indexing the patients and j the continuous segments. Each segment $\mathbf{x}_j^{(i)}$ has length $L_j^{(i)} > L_{\text{min}}$, where L_{min} is the pre-specified minimal value (Table 3).

Covariates Encoding. While many of the covariates are real-valued, e.g., age, some covariates are categorical, e.g., sex. In particular, Weinstock et al. (2016) dataset has 36 categorical covariates with an average of 10 levels per covariate. While one-hot encoding is a popular approach for modeling categorical covariates, it will lead to 360 feature columns on Weinstock et al. (2016), making it undesirable from model training time and memory perspectives. Instead, we use label encoding by converting each categorical level into a numerical value. Given R covariates, we include them in the dataset as $\mathcal{D} = \left\{\mathbf{x}_j^{(i)}, \mathbf{c}_{1,j}^{(i)}, \dots \mathbf{c}_{R,j}^{(i)}\right\}_{i,j}$ where $\mathbf{c}_{r,j}^{(i)} \in \mathbb{R}$ for static and $\mathbf{c}_{r,j}^{(i)} \in \mathbb{R}^{L_j^{(i)}}$ for dynamic.

Data splitting. Each dataset is split into train, validation, and in-distribution (ID) test sets using 90% of subjects. For each subject, the sets follow chronological time order as shown in Figure 3, with validation and ID test sets always being of a fixed length of 16 hours each (192 measurements). The data from the remaining 10% of subjects is used to form an out-of-distribution (OD) test set to assess the generalization abilities of predictive models as in Section 5.2. Thus, $\mathcal{D} = \mathcal{D}_{tr} \cup \mathcal{D}_{val} \cup \mathcal{D}_{id} \cup \mathcal{D}_{od}$.

Scaling. We use min-max scaling to standardize the measurement range for both glucose values and covariates. The minimum and maximum values are computed per dataset using $\mathcal{D}_{tr}$, and then the same values are used to rescale $\mathcal{D}_{val}, \mathcal{D}_{id}$, and $\mathcal{D}_{od}$.

Input-output pairs. Let $\mathbf{x}_{j,k:k+L}^{(i)}$ be a length L contiguous slice of a segment from index k to $k+L$. We define an input-output pair as $\left\{\mathbf{x}_{j,k:k+L}^{(i)}, \mathbf{y}_{j,k+L+1:k+L+T}^{(i)}\right\}$, where $\mathbf{y}_{j,k+L+1:k+L+T}^{(i)} = \mathbf{x}_{j,k+L+1:k+L+T}^{(i)}$ and T is the length of prediction interval. Our choices of T, L and k are as follows. In line with the previous works (Oviedo et al., 2017) we focus on the 1 -hour ahead forecasting ($T = 12$ for 5 minute frequency). We treat L as a hyper-parameter for model optimization since different models have different capabilities in capturing long-term dependencies. We sample k without replacement from among the index set of the segment during training, similar to Oreshkin et al. (2020); Challu et al. (2023), and treat the total number of samples as a model hyper-parameter. We found the sampling approach to be preferable over the use of a sliding window with a unit stride (Herzen et al., 2022), as the latter is computationally prohibitive on larger training datasets and leads to high between-sample correlation,

Table 2: Demographic information (average) for each dataset before (Raw) and after pre-processing (Processed). CGM indicates the device type; all devices have 5 minute measurment frequency.

Dataset	Diabetes	CGM	# of Subjects		Age		Sex (M/F)	
	Overall	Overall	Raw	Processed	Raw	Processed	Raw	Processed
Broll et al. (2021)	Type 2	Dexcom G4	5	5	NA	NA	NA	NA
Colas et al. (2019	Mixed	MiniMed iPro	208	201	59	59	103 / 104	100 / 100
Dubosson et al. (2018)	Type 1	MiniMed iPro2	9	7	NA	NA	6 / 3	NA
Hall et al. (2018,	Mixed	Dexcom G4	57	56	48	48	25 / 32	NA
Weinstock et al. (2016)	Type 1	Dexcom G4	200	192	68	NA	106 / 94	101 / 91

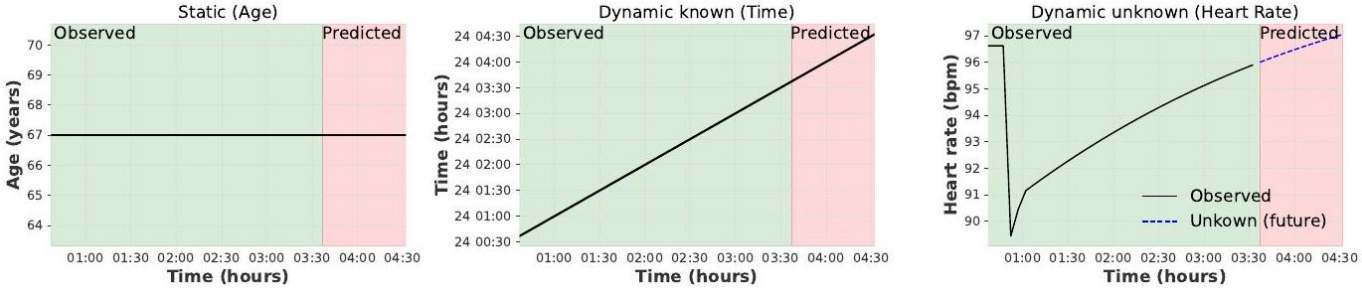

Figure 2: An illustration of static (Age), dynamic known (Date), and dynamic unknown (Heart Rate) covariate categories based on data from Hall et al. (2018) and Dubosson et al. (2018).

Table 3: Interpolation parameters for datasets.

Parameters	Broll	Colas	Dubosson	Hall	Weinstock
Gap threshold (minutes)	45	45	30	30	45
Minimum length (hours)	20	16	20	16	20

slowing convergence in optimization. We still use the sliding window when evaluating the model on the test set.

4 BENCHMARKS

4.1 TASKS AND METRICS

Task 1: Predictive Accuracy. Given the model prediction $\hat{\mathbf{y}}_{j,k+L:k+L+T}$, we measure accuracy on the test set using root mean squared error (RMSE) and mean absolute error (MAE):

$$RMSE_{i,j,k} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y^{(i)}_{j,k+L+t} - \hat{y}^{(i)}_{j,k+L+t}\right)^2}; \quad MAE_{i,j,k} = \frac{1}{T}\sum_{t=1}^{T}\left|y^{(i)}_{j,k+L+t} - \hat{y}^{(i)}_{j,k+L+t}\right|.$$

Since the distribution of MAE and RMSE across samples is right-skewed, we use the median of the errors as has been done in Sergazinov et al. (2023); Armandpour et al. (2021).

Task 2: Uncertainty Quantification. To measure the quality of uncertainty quantification, we use two metrics: log-likelihood on test data and calibration. For models that estimate a parametric predictive distribution over the future values, $\hat{P}^{(i)}_{j,k+L+1:k+L+T} : \mathbb{R}^T \to [0,1]$, we evaluate log-likelihood as

$$\log L_{i,j,k} = \log \hat{P}^{(i)}_{j,k+L+1:k+L+T}\left(\mathbf{y}^{(i)}_{j,k+L+1:k+L+T}\right),$$

where the parameters of the distribution are learned from training data, and the likelihood is evaluated on test data. Higher values indicate a better fit to the observed distribution. For both parametric and non-parametric models (such as quantile-based methods), we use regression calibration metric (Kuleshov et al., 2018). The original metric is limited only to the univariate distributions. To address the issue, we report an average calibration across marginal estimates for each time $t = 1, \ldots, T$. To compute marginal calibration at time t, we (1) pick M target confidence levels $0 < p_1 < \cdots < p_M < 1$; (2) estimate realized confidence level $\hat{p}_m$ using N test input-output pairs as

$$\hat{p}_m = \frac{\left|\left\{y^{(i)}_{j,k+L+t} \mid \hat{F}^{(i)}_{j,k+L+t}\left(y^{(i)}_{j,k+L+t}\right) \le p_m\right\}\right|}{N}$$

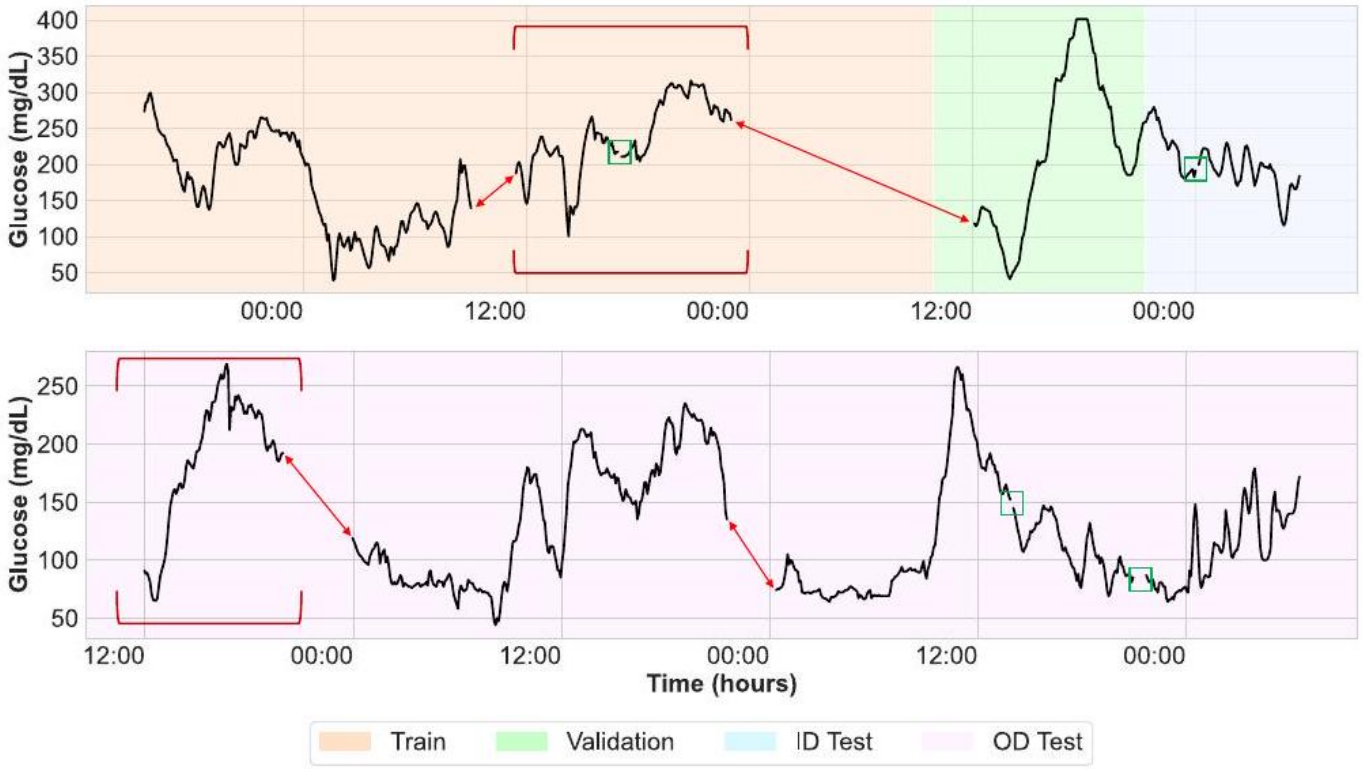

Figure 3: Example data processing on Weinstock et al. (2016). The red arrows denote segmentation, green blocks interpolate values, and red brackets indicate dropped segments due to length.

and (3) compute calibration across all M levels as

$$Cal_t = \sum_{1}^{M} (p_m - \hat{p}_m)^2$$

The smaller the calibration value, the better the match between the estimated and true levels.

4.2 Models

To benchmark the performance on the two tasks, we compare the following models. ARIMA is a classical time-series model, which has been previously used for glucose predictions (Otoom et al., 2015; Yang et al., 2019). Linear regression is a simple baseline with a separate model for each time step $t = 1, \ldots, T$. XGBoost (Chen & Guestrin, 2016) is gradient-boosted tree method, with a separate model for each time step t to support multi-output regression. Transformer represents a standard encoder-decoder auto-regressive Transformer implementation (Vaswani et al., 2017). Temporal Fusion Transformer (TFT) is a quantile-based model that uses RNN with attention. TFT is the only model that offers out-of-the-box support for static, dynamic known, and dynamic unknown covariates. NHiTS uses neural hierarchical interpolation for time series, focusing on the frequency domain (Challu et al., 2023). Latent ODE uses a recurrent neural network (RNN) to encode the sequence to a latent representation (Rubanova et al., 2019). The dynamics in the latent space are captured with another RNN with hidden state transitions modeled as an ODE. Finally, a generative model maps the latent space back to the original space. Gluformer is a probabilistic Transformer model that models forecasts using a mixture distribution (Sergazinov et al., 2023). For ARIMA, we use the code from (Garza et al., 2022) which implements the algorithm from (Hyndman & Khandakar, 2008). For linear regression, XGBoost, TFT, and NHiTS, we use the open-source DARTS library (Herzen et al., 2022). For Latent ODE and Gluformer, we use the implementation in PyTorch (Rubanova et al., 2019; Sergazinov et al., 2023). We report the compute resources in Appendix C.

4.3 Testing protocols

In devising the experiments, we pursue the principles of reproducibility and fairness to all methods.
Reproducibility. As the performance results are data split dependent, we train and evaluate each model using the same two random splits. Additionally, all stochastically-trained models (tree-based and deep learning) are initialized 10 times on each training set with different random seeds. Thus, each stochastically-trained model is re-trained and re-evaluated 20 times, and each deterministicallytrained model 2 times, with the final performance score taken as an average across evaluations. We report standard error of each metric across the re-runs in Appendix B.

Fairness. To promote fairness and limit the scope of comparisons, we focus on out-of-the-box model performance when establishing baselines. Thus, we do not consider additional

model-specific tuning that could lead to performance improvements, e.g., pre-training, additional loss functions, data augmentation, distillation, learning rate warm-up, learning rate decay, etc. However, since model hyper-parameters can significantly affect performance, we automate the selection of these parameters. For ARIMA, we use the native automatic hyper-parameter selection algorithm provided in (Hyndman & Khandakar, 2008). For all other models, we use Optuna (Akiba et al., 2019) to run Bayesian optimization with a fixed budget of 50 iterations. We provide a discussion on the selected optimal model hyperparameters for each dataset in the supplement (Appendix C).