electronics

MDPI

Article

Comparison of Machine Learning Models for Predicting Interstitial Glucose Using Smart Watch and Food Log

Haider Ali [1,2], Imran Khan Niazi [3,4,5], David White [1,2], Malik Naveed Akhter [5] and Samaneh Madanian [1,*]

[1] Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; haider.ali@aut.ac.nz (H.A.); david.white@aut.ac.nz (D.W.)
[2] Biodesign Lab, New Zealand College of Chiropractic, Auckland 1010, New Zealand
[3] Center of Chiropractic Research, New Zealand College of Chiropractic, Auckland 1010, New Zealand
[4] Center for Sensory-Motor Interaction, Department of Health Science and Technology, Aalborg University, 9220 Aalborg, Denmark
[5] Department of Clinical Sciences, Auckland University of Technology, Aukcland 1010, New Zealand; pjt6828@autuni.ac.nz
* Correspondence: sam.madanian@aut.ac.nz; Tel.: +64-(09)-921-9999 (ext. 6539)

Abstract: This study examines the performance of various machine learning (ML) models in predicting Interstitial Glucose (IG) levels using data from wrist-worn wearable sensors. The insights from these predictions can aid in understanding metabolic syndromes and disease states. A public dataset comprising information from the Empatica E4 smart watch, the Dexcom Continuous Glucose Monitor (CGM) measuring IG, and a food log was utilized. The raw data were processed into features, which were then used to train different ML models. This study evaluates the performance of decision tree (DT), support vector machine (SVM), Random Forest (RF), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), lasso cross-validation (LassoCV), Ridge, Elastic Net, and XGBoost models. For classification, IG labels were categorized into high, standard, and low, and the performance of the ML models was assessed using accuracy (40–78%), precision (41–78%), recall (39–77%), F1-score (0.31–0.77), and receiver operating characteristic (ROC) curves. Regression models predicting IG values were evaluated based on R-squared values (−7.84–0.84), mean absolute error (5.54–60.84 mg/dL), root mean square error (9.04–68.07 mg/dL), and visual methods like residual and QQ plots. To assess whether the differences between models were statistically significant, the Friedman test was carried out and was interpreted using the Nemenyi post hoc test. Tree-based models, particularly RF and DT, demonstrated superior accuracy for classification tasks in comparison to other models. For regression, the RF model achieved the lowest RMSE of 9.04 mg/dL with an R-squared value of 0.84, while the GNB model performed the worst, with an RMSE of 68.07 mg/dL. A SHAP analysis identified time from midnight as the most significant predictor. Partial dependence plots revealed complex feature interactions in the RF model, contrasting with the simpler interactions captured by LDA.

Keywords: wearable sensors; machine learning; interstitial glucose; Empatica E4

Citation: Ali, H.; Niazi, I.K.; White, D.; Akhter, M.N.; Madanian, S. Comparison of Machine Learning Models for Predicting Interstitial Glucose Using Smart Watch and Food Log. *Electronics* **2024**, *13*, 3192. https://doi.org/10.3390/electronics13163192

Academic Editor: Shing-Hong Liu

Received: 8 July 2024
Revised: 27 July 2024
Accepted: 8 August 2024
Published: 12 August 2024

1. Introduction

Diabetes is characterized by increased glucose levels. The incidence of it is increasing at a rapid rate. According to the World Health Organization, the number of people with diabetes worldwide rose from 108 million in 1980 to 422 million in 2014 [1], and this number is projected to reach 700 million by 2045 [2]. Prediabetes is a series of risk factors of diabetes defined using fasting glucose levels between 100 and 125 mg/dL [3]. Prediabetes affects approximately 34% of adults in the United States [3], with nearly 7.3 million undiagnosed cases [4,5]. However, 85% of individuals with prediabetes are unaware [6] that they have it [7]. Early intervention through lifestyle changes or medication can significantly reduce the risk of progression from prediabetes to diabetes by up to 58% [8]. Monitoring glucose

levels is thus helpful for managing and preventing metabolic diseases [9]. Classically, glucose levels are measured using a blood test that measures glycated hemoglobin levels (HbA1C). Fasting HbA1C levels measure glucose regulation for the past two to three months and do not measure fluctuations and short-term glucose spikes [10]. Monitoring short-term glucose variations is essential for adjusting medication, dietary habits, and physical activity to maintain optimal glucose regulation. To measure these short-term glucose spikes, continuous glucose markers (CGMs) are used. Glucose regulation markers such as time in range (TIR) can be measured using CGMs. CGMs are attached to the body with the help of a thread that penetrates the interstitial fluid. CGMs log Interstitial Glucose (IG) values every one to five minutes depending on the device. IG values are stored in them for up to 8 h. The stored IG values from CGMs can be downloaded with the help of Bluetooth technology [11]. CGMs require regular downloading of the data and are minimally invasive.

In comparison to CGMs, smart watches are noninvasive and self-updating. Therefore, there is an increased interest in using smart watches for predicting IG levels. An example of this growing interest is the curation of various datasets [12] that include smart watch data paired with glucose labels [13]. Smart watches are equipped with sensors capable of tracking various physiological parameters. Smart watch sensors include heart rate, an accelerometer, and skin conductance, etc. The smart watch sensor values can be used to engineer predictors of IG [14].

In addition to enhancing individual glucose management, predicting IG levels from smart watch sensors can also contribute to population-level health insights and disease management strategies. Aggregated data from smart watches and CGMs can provide valuable epidemiological information about glucose trends, the prevalence of metabolic conditions, and the impact of lifestyle factors on glycemic control.

Machine learning (ML) algorithms are used to predict IG markers from smart watches due to their ability to extract complex patterns and relationships [15]. ML models can adapt and improve over time by continuously learning from new data, making them well suited for personalized glucose monitoring and management [16]. Studies have demonstrated the effectiveness of ML algorithms in predicting glucose levels from wearable sensor data, achieving high levels of accuracy and precision [1,13,15,17]. Many of these models add food log data, in addition to smart watch data.

In this study, we categorize continuous glucose monitoring (CGM) values into high, low, and normal labels as described in [13,17,18]. Unlike traditional classifications of hyper- and hypoglycemia, which are tailored for diabetic patients, this approach considers individualized glucose fluctuations. These designations are dynamic and personalized, reflecting an individual's unique glycemic baseline and accounting for circadian and intra-/inter-day variability.

Related Work

Earlier works utilized support vector machines (SVMs) and decision trees for predicting IG values and categories using smart watch data. For example, ref. [18] produced 69 features predictive of glucose, defined the classification problem, and used decision trees to perform a classification with a root mean squared error (RMSE) equal to 21.22 ± 4.14 mg/dL. Similar works used depth vision guiding for recognizing human activity that can be used as input to glucose-monitoring models [19,20]. However, this requires additional sensors. Another work [13] recently designed four additional features using smart watch data but only performed a classification of CGM values into normal, high, and low and found support vector machine (SVM) to have an accuracy of 69% and decision tree (DT) to be 72.38% accurate. Another work utilizes extreme gradient boosting (XGBoost) models to classify the IG values of each participant with minimum accuracy = 60% and maximum accuracy = 86% [15]. While these works used smart watches and CGM data to train ML models to predict IG markers (classes and values), it would be useful to compare different ML models for both the classification and regression problem using the same

performance metrics. While these works report a hyperparameter tuning process for the models, there is a need for hyperparameter tuning for the best performing models. To compare the performance of the ML models, model explanations and visual techniques can inform why certain models such as tree models outperform other models [18]. The comparisons of earlier works is given in Table 1.

Table 1. The performance of different models in related works.

Study	Type	Models	Performance
[18]	Classification/Regression	DT	MSE = 21.22 ± 4.14 mg/dL
[13]	Classification	DT/SVM	Accuracy SVM (69%), DT (72.38%)
[15]	Classification	XGBoost	Accuracy (60–86%) *
[21]	Regression	Gradient Boosting	MSE = 23.40 mg/dL
[22]	Classification	DT	AUROC = 0.76 ± 0.07
[23]	Regression	RF	RMSE = 26.83 mg/dL
[14]	Classification	SVM	Accuracy = 72.6 ± 2.4%

* Individual specific models.

In summary, this work makes these novel contributions in comparison to other works:

- C1: Compare performance of ML models in predicting IG values using smart watch data as input and compared using Friedman test with Neymani post hoc analysis;
- C2: Compare performance of ML models in classifying IG values into high, low, and normal classes using smart watch data as input and compared using Friedman test with Neymani post hoc analysis;
- C3: Explain why different model types, such as tree-based models (RF, DT, and XGBoost), outperform SVM and GNB using partial dependence plots and Cook's plots.

In this context, our study compares several ML models, including DT, SVM, RF, Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), lasso cross-validation (LassoCV), Ridge, Elastic Net, and XGBoost. For the classification task, accuracy, precision, recall, F1-score, and ROC are used to compare models. Additionally, regression models predicting IG values are evaluated using R-squared values, mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and mean squared logarithmic error (MSLE). The hyperparameters of the best performing models are optimized using Bayesian Optimization with Optuna [24]. This work also explains why different models perform better than others, using partial dependence plots (PDP) to show feature interactions. This work also shows the robustness of RF models to influential outliers and the existence of such outliers using a Cook's plot.

The paper is organized as follows: Section 2 Materials and Methods: A detailed description of the dataset, highlighting its key characteristics and pertinent attributes, and an explanation of the data preprocessing, feature extraction techniques, and machine learning models used in this study. Section 3 Results: Presentation and comparison of the outcomes from the regression and classification analyses and their limitations. Sections 4 and 5 Discussion and Conclusion: Synthesis of the findings, discussion of their implications. The structure of paper is represented in Figure 1.

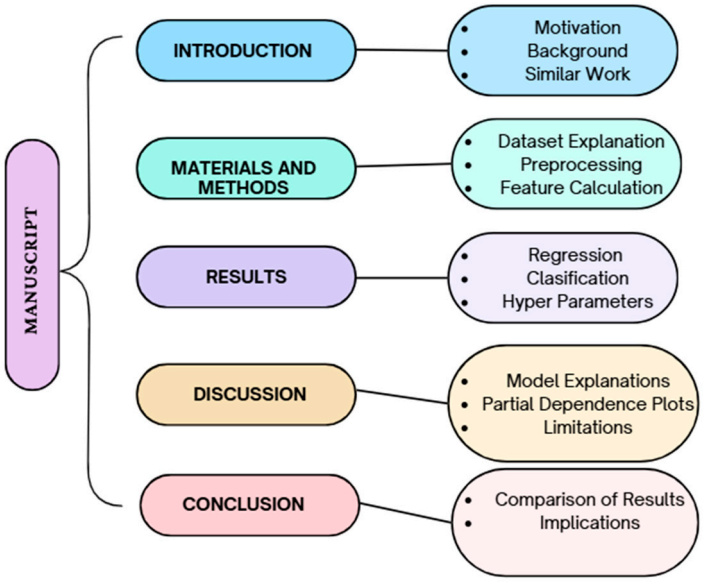


Figure 1. Structure of the manuscript.

2. Materials and Methods

In this study, we utilized a public dataset [12]. These data comprise a cohort of participants aged 35–65 years, inclusive, with elevated blood glucose levels falling within the range of normal to prediabetic. The dataset consists of 9 female participants and 7 male participants. The demographics are given in Supplementary Table S1. In this dataset, participants were required to wear a Dexcom G6 CGM and an Empatica E4 wristband for a duration of 8–10 days, during which physiological measurements such as heart rate, electrodermal activity, skin temperature, and tri-axial accelerometry were recorded.

Participants were also provided standardized breakfast meals every other day, and a food log was maintained. Date shifting was performed on the collected data to ensure participant de-identification.

The dataset includes a total of 16 participants. The files contain timestamped data. The ACC file provides accelerometer data for the X, Y, and Z orientations, while the BVP file records blood volume pulse measurements. The Dexcom, EDA, TEMP, IBI, and HR files contain data of IG values, electrodermal activity, skin temperature, inter-beat interval, and heart rate values, respectively. The food log file documents the food items consumed by each participant, including details such as date, time, logged food, amount, calories, total carbohydrates, dietary fiber, sugar, protein, and total fat content. Demographics, including gender and HbA1C values for each participant, are also provided. The PPG is sampled at 64 Hz, giving the HR and BVP every second for IBI computation. The EDA and skin temperature are sampled at 4 Hz, and the accelerometry at 32 Hz. CGM records a value of IG every 5 min.

For preprocessing, the HR and IBI data were filtered with a Chebyshev II order-4 filter with a stopband attenuation of 20 dB and a passband of 0.5–5 Hz, as described in [25]. For the removal of noise, a Gaussian low-pass filter was used with a sigma value of 400 ms [26]. We then segmented the filtered data into 5 min windows and aligned them with the Dexcom sensor values. Features were extracted from each window as described in [18]. These features can be broadly categorized into circadian features, statistical features of the sensor values, EDA features, and food features. A five-minute window was used for the calculation of these features, as the IG ground truth is available every five minutes. For the classification of IG labels, daily averages and standard deviations of CGM values were calculated. CGM values that are higher than the mean + standard deviation are considered high; conversely, values smaller than mean–standard deviation are considered low, whereas all the other values are considered normal, as described in [18].

Accelerometer data were preprocessed using a Butterworth low-pass filter with cutoff frequency = 20 Hz, as explained in [27]. The resultant acceleration was calculated from the X, Y, and Z components and corrected for the gravitational acceleration in the y direction. The EDA sensor data were smoothed to remove any artefacts. To do this, a Gaussian low-pass filter is used, with a 40-point window and value of sigma = 400 ms [26]. The HR data are filtered using a band-pass filter, filtering activity outside the [0.5–4 Hz] range. IBI data are filtered using a filter defined in [28]. BVP data are filtered using a moving average smoothing filter, whereas temperature sensor data are filtered using a Savitzky-Golay filter [29]. This is explained in Figure 2.

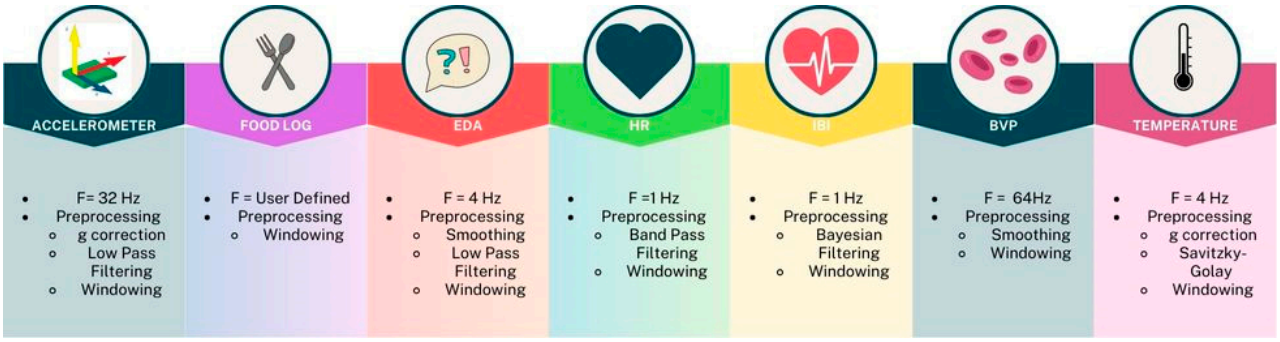


Figure 2. Preprocessing steps for each data source.

The features are calculated in this work as described in [13,18]. The features are broadly categorized into four main classes: food features, circadian features, statistical features, and autonomic nervous system features. These features are calculated for each 5 min window. The mathematical definition of the features is provided in Table 2.

Table 2. Features used in this work.

Feature	Description	Mathematical Expression
Biological Sex		
HbA1C	Glycated hemoglobin usually measured before the longitudinal data collection	
Mean of EDA, HR, IBI, T, and a	The mean of S (sensor value for the window for prediction usually equal to 5 min)	$\mu_S = \frac{\sum_{i=0}^{N} S}{N}$
Standard Deviation of EDA, HR, IBI, T, and a	The standard deviation of the S values for the length of the window	$\sigma_S = \sqrt{\frac{\left(\sum_{i=0}^{N} \mu_S - S\right)^2}{N}}$
Minimum Value of EDA, HR, IBI, T, and a		$\min_N S$
Maximum Value of EDA, HR, IBI, T, and a		$\max_N S$
First Quartile of EDA, HR, IBI, T and a	The value of 25% data point when the data are arranged in ascending order	$S(I)$ where I is the index of the S values in t ascending order rounded off to the nearest integer $I = \frac{N+1}{4}$
Third Quartile of EDA, HR, IBI, T and a	The value of 75% data point when the data are arranged in ascending order	$S(I)$ where I is the index of the S values in ascending order rounded off to the nearest integer $I = \frac{(N+1)x3}{4}$
Skewness of EDA, HR, IBI, T, and a	It is a measure of how symmetric the data are from the mean	$S = \frac{\left(\sum_{i=0}^{N} \mu_S - S\right)^3}{(N-1)\sigma_S^3}$
Peak of EDA values	Peak of EDA values in the prediction window for peaks of prominence 0.3	$\sum_{i=0}^{N} P$
Rolling mean of two hours of EDA values		
Rolling sum of 2 h of EDA peaks		
Standard Deviation of IBI (SDNN)	It is a measure of heart rate variability	
Root mean square of successive differences of inter-beat interval (RMSSD)	It is a measure of heart rate variability that is related to autonomic nervous system tone	$\mathrm{RMSSD} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N-1}\left[\mathrm{IBI}_{n+1} - \mathrm{IBI}_n\right]^2}$

Table 2. *Cont.*

Feature	Description	Mathematical Expression
Number of times IBI exceeds 50 ms (NN50)	It is a measure of heart rate variability and sympathetic nervous system activation	$NN_{50} = \sum_{i=1}^{n} a$ where a = 1 if $IBI_{i+1} - IBI_i > 50$ Else a = 0
pNN50	It is the measure of how many times in time N the IBI has exceeded 50 ms	$pNN_{50} = \frac{NN_{50}}{N}$
Calorie Sums	Rolling sums of 2 h, 8 h, and 24 h of calorie estimates are used	
Protein Sums	Rolling sums of 2 h, 8 h, and 24 h of protein consumption estimates are used	
Carbohydrate Sums	Rolling sums of 2 h, 8 h, and 24 h of carbohydrate consumption estimates are used	
Sugar Sums	Rolling sums of 2 h, 8 h, and 24 h of sugar consumption estimates are used	
Rolling mean of two-hour acceleration values	Used to estimate activity levels	
Rolling maximum value of two-hour acceleration values	Used to estimate activity levels	
Activity Bouts	When the mean of the window exceeds the rolling mean of acceleration values	
Individual Number	Used to model individuality	

The efficacy of the features was verified based on correlation and mutual information for discrete and t-Distributed Stochastic Neighbor Embedding (t-SNE) plots. Correlation is used to measure the independent features. These features are used to train ML models.

DT is a non-parametric ML algorithm used for classification and regression tasks. It splits the data into subsets based on the value of the input features, forming a tree structure where each node represents a feature, each branch represents a decision rule, and each leaf represents an outcome. The goal is to create a model that predicts the target variable by learning simple decision rules inferred from the data features.

SVM finds the hyperplane that best separates the classes in the feature space, maximizing the margin between the closest points of the classes (support vectors). SVM is effective in high-dimensional spaces and is particularly useful for problems where the number of dimensions exceeds the number of samples.

RF is an ensemble learning method that combines multiple decision trees to improve predictive performance and control overfitting. Each tree is trained on a random subset of the data and the features. The final prediction is made by averaging the outputs of individual trees (for regression) or by majority voting (for classification).

LDA is a dimensionality reduction technique used for classification. It projects the data onto a lower-dimensional space where the classes are most separable. LDA assumes that the features follow a Gaussian distribution and that different classes have identical covariances. It finds linear combinations of the features to predict the labels.

KNN is a simple, instance-based learning algorithm used for classification and regression. It predicts the target by finding the K training samples closest in distance to a new data point and returning the majority class (for classification) or the average value (for regression). KNN is non-parametric and makes predictions based on the entire dataset.

GNB is a probabilistic classifier based on Bayes' theorem, assuming independence between features given the class. It models the distribution of the features within each class as Gaussian.

LassoCV is a linear regression model that includes L1 regularization (lasso) to enforce sparsity, reducing the number of features by shrinking some coefficients to zero. The CV part stands for cross-validation, which is used to find the optimal regularization parameter. It helps prevent overfitting by penalizing the absolute size of the coefficients.

Ridge regression is a linear regression model with L2 regularization, which penalizes the squared magnitude of the coefficients. This regularization helps to prevent overfitting by shrinking the coefficients towards zero but unlike lasso, it does not enforce sparsity. It is useful when dealing with multicollinearity or when the number of predictors exceeds the number of observations.

AdaBoost, short for Adaptive Boosting, is an ensemble learning technique that combines multiple weak classifiers to create a strong classifier. It works by iteratively training classifiers on weighted versions of the data, where the weights are adjusted to focus on the hardest-to-classify samples. The final model is a weighted sum of the individual classifiers.

XGBoost (extreme gradient boosting) is an efficient and scalable implementation of the gradient boosting framework. It builds an ensemble of decision trees in a sequential manner, where each tree corrects the errors of its predecessor. XGBoost uses advanced regularization techniques to reduce overfitting and includes features like parallel tree construction, handling missing values, and optimized computations.

For both regression and classification, numerical features were normalized using the Z-score. The Z-score is defined in Equation (1).

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where x gives the value of the feature, μ is the mean of the feature's distribution, and σ is its standard deviation. There are two categorical features in this dataset: participant ID and gender. One-hot encoding is a method used to transform categorical data into a numerical format suitable for ML models. It converts each category into a unique binary vector, where only one element is set to 1 and the rest are 0. In this work, the categorical features are one-hot encoded. The ML models are trained using a subset of data called training data, learning to predict IG values with respect to the input smart watch and food log features. Features from an independent subset called validation data are used to predict IG values; these values are compared with actual IG values for determining the performance of the model.

The following performance metrics are used in the comparison of the regression models.

1. MAE: MAE measures the average absolute difference between the predicted and actual IG values. It is less sensitive to outliers;
2. RMSE: It is the root of the average squared difference between the predicted and actual IG values;
3. MAPE: MAPE measures the average ratio of error (difference between the actual and predicted value) with the actual IG value;
4. R-squared (R^2) and Adjusted R-squared: R^2 measures the proportion of variance in the CGM values explained by the input smart watch features. A higher R^2 indicates a better fit of the model to the data;
5. MSLE: It is the average difference between the log of the actual and predicted IG values. It is specifically less sensitive to outliers.

For classification, the following parameters are used to define the performance of the models. A true positive (TP) occurs when the model correctly predicts a positive class for an instance that is actually positive. A true negative (TN) is when the model correctly predicts a negative class for an instance that is actually negative. A false positive (FP), also known as a Type I error, happens when the model incorrectly predicts a positive class for an instance that is actually negative. Conversely, a false negative (FN), or Type II error, occurs when the model incorrectly predicts a negative class for an instance that is actually positive.

The following performance metrics are used to compare the classification models.

1. Accuracy (%): Accuracy measures the proportion of correctly classified instances out of the total instances. It is given by (TP+TN/TP+TN+FP+FN);
2. Precision: Precision, also known as positive predictive value, measures the proportion of true positive predictions among all positive predictions made by the model. It is given as (TP/TP+FP);
3. Recall: Recall, also known as sensitivity or the true positive rate, measures the proportion of true positives identified by the model out of all actual positives. It is given as (TP/TP+FN);
4. F1-Score: The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both precision and recall;

5. ROC (Receiver Operating Characteristic) Curve: The ROC curve plots the true positive rate (recall) against the false positive rate at various threshold settings. The ROC AUC (Area Under the Curve) measures the model's ability to discriminate between classes, with a higher AUC indicating a better performance.

To make sure that the difference between performance metrics is statistically significant, they are compared using the Friedman test. This is done by dividing the dataset into 10 folds, training it on 9 folds and testing on the remaining fold (once for each fold). The performance metrics are recorded, and these differences are interpreted using the Nemenyi post hoc test.

3. Results

3.1. Feature Calculation

These features are calculated in python using NumPy, pandas, and JAX.

The correlation heatmap in Figure 3 illustrates the relationships between various calculated features from the food log and smart watch data. Notably, some clusters of features, such as those related to proteins and carbohydrates over different time windows, exhibit strong correlations within their groups. Conversely, features like heart rate variability (HRV) metrics and activity measures demonstrate more independence, as indicated by their lighter shades.

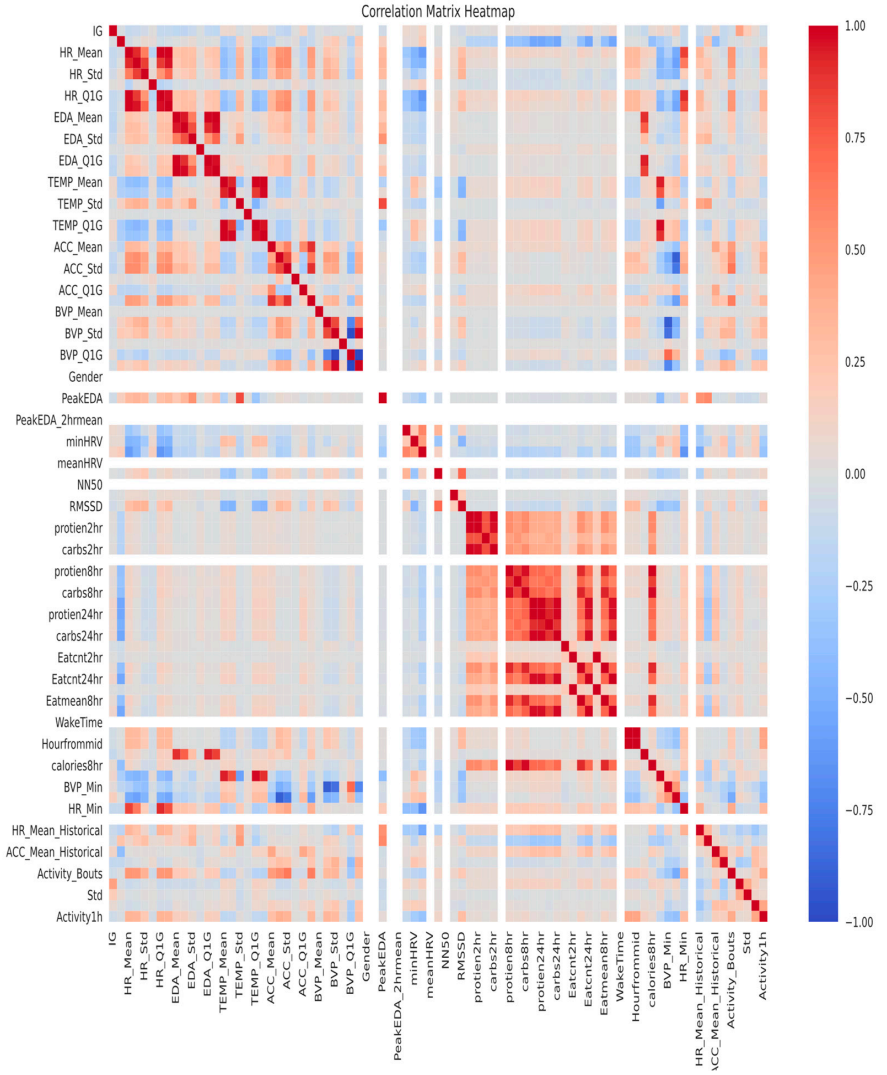


Figure 3. Correlation heatmap for all the calculated features. The stronger shades of red signify a positive correlation, and blue signifies a negative correlation. The lighter shades signify the features that have a smaller correlation, meaning that they are potentially independent.

Figure 4 illustrates the Pearson correlation coefficients between various features and IG values. Each feature's correlation with the dependent variable is visually represented, where positive correlations extend to the right and negative correlations to the left. Features such as activity counts, historical accelerometer data (ACC), heart rate (HR) metrics, and carbohydrate intake exhibit high degrees of correlation with the CGM values.

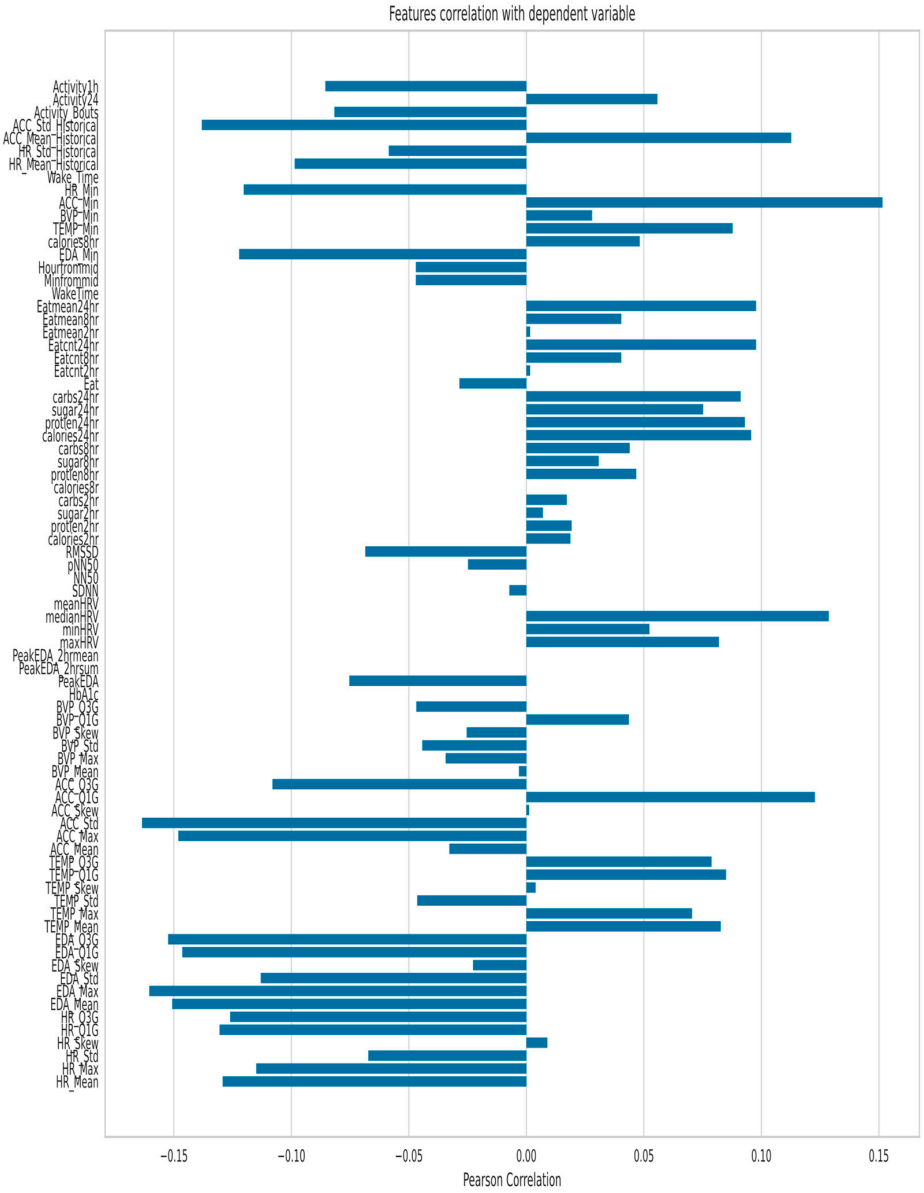


Figure 4. Feature correlation with Interstitial Glucose levels.

3.2. Regression

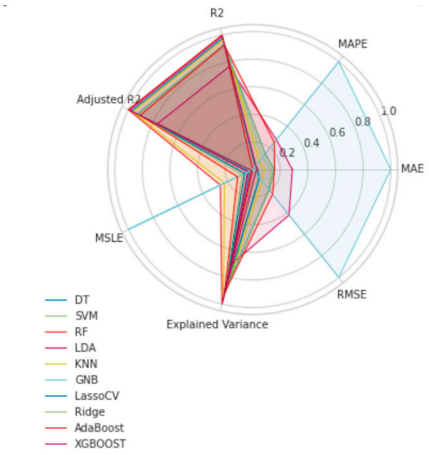
Regression models predict IG values based on smart watch and food log features. These ML models are used for IG prediction: DT, SVM, RF, LDA, KNN, GNB, LassoCV, Ridge, AdaBoost, and XGBoost. After identifying the best performing model type (RL), its hyperparameters were tuned using Bayesian Optimization.

The feature data were split into (70%) training and (30%) testing. The models were trained on the training set. The trained models were used to predict IG values on the unseen testing data. The predicted IG values were subtracted from the actual IG value to determine the *error* value. The error values were used to calculate various performance metrics of the regression models. Table 3 compares the performance metrics of the ML models in predicting IG values based on the input features.

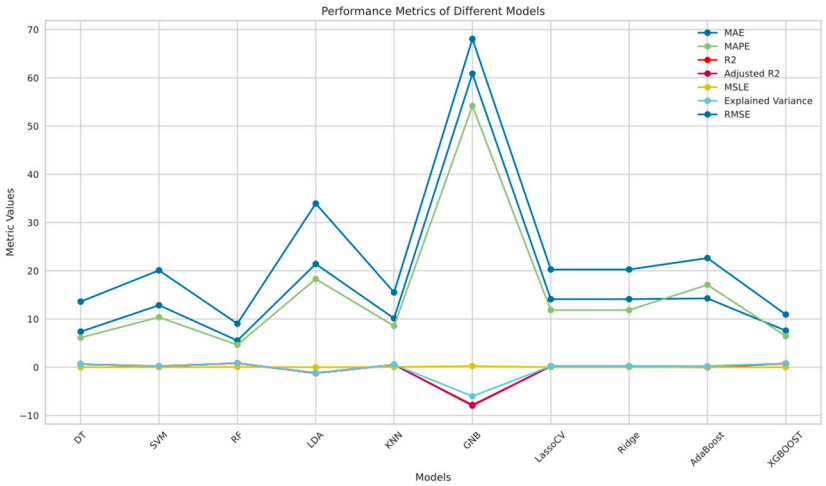
Table 3. Performance metrics of regression models.

Model	MAE (mg/dL)	MAPE	R^2	Adjusted R^2	MSLE (mg/dL)	Explained Variance	RMSE (mg/dL)
DT	7.379	6.15	0.64	0.64	0.0115	0.6475	13.61
SVM	12.86	10.37	0.23	0.21	0.004	0.25	20.09
RF	5.54	4.65	0.84	0.84	0.068	0.84	9.04
LDA	21.42	18.30	−1.19	−1.22	0.014	−1.16	33.94
KNN	10.14	8.57	0.54	0.53	0.06	0.54	15.51
GNB	60.85	54.18	−7.81	−7.96	0.25	−6.00	68.07
LassoCV	14.11	11.84	0.21	0.20	0.02	0.21	20.27
Ridge	14.12	11.85	0.21	0.20	0.025	0.21	20.27
AdaBoost	14.28	17.10	0.194	0.007	0.034	0.27	22.64
XGBoost	7.59	6.45	0.77	0.768	0.007	0.77	10.93

Figure 5 presents the performance parameters of the models visually. Residuals and QQ plots are plotted in the Supplementary Material. RF consistently has high performance across all metrics. The RF model has a high R^2, adjusted R^2, and explained variance. while maintaining a low MAE, MAPE, MSLE, and RMSE.



(**a**)



(**b**)

Figure 5. Comparison of the performance metrics of regression models: (**a**) Normalized spider plot for difference performance metrics of regression results; (**b**) bar plot for performance measures of different models.

To make sure that the difference between the metrics for each model is significant, a Friedman test is carried out for each metric. The results for each metric and model are reported in the Supplementary Material. An example of this analysis is shown here for reference. The Friedman statistic for values of MAE for all the models for all folds is 70.0, with ($p = 1.47 \times 10^{-12}$) showing that the difference is significant. To understand for which models the comparison is significant, a Nemenyi post hoc analysis for the results is conducted and plotted as a heatmap in Figure 6.

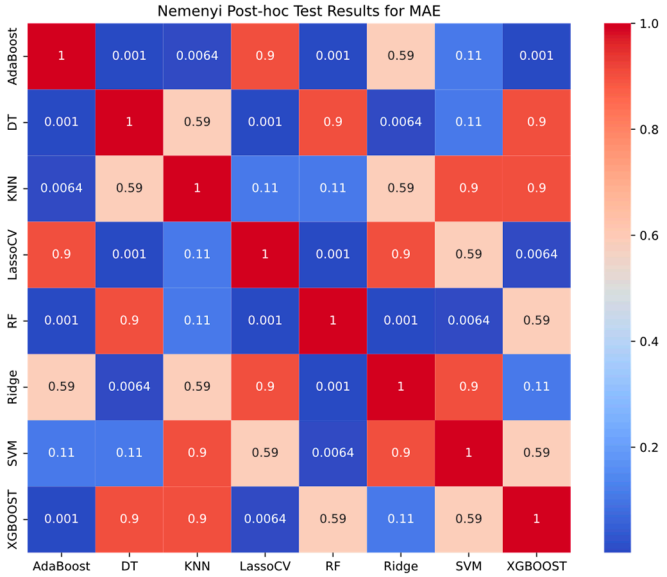


Figure 6. Nemenyi post hoc analysis of the Friedman test for MAE across all the models.

Based on the Nemenyi post hoc test results, we can observe how different models compare against each other in terms of performance. The heatmap visualizes the pairwise comparisons of the models, highlighting the statistically significant differences in their MAE values. Each cell in the heatmap represents a comparison between two models. A smaller value signifies a more substantial difference in performance between the compared models.

The Nemenyi post hoc test results for mean absolute error (MAE) reveal significant and non-significant differences in model performance. Significant differences ($p < 0.05$) were observed between AdaBoost and decision tree ($p = 0.001$), AdaBoost and KNN ($p = 0.006$), AdaBoost and Random Forest ($p = 0.001$), AdaBoost and XGBoost ($p = 0.001$), decision tree and LassoCV ($p = 0.001$), decision tree and Ridge ($p = 0.006$), Random Forest and LassoCV ($p = 0.001$), Random Forest and Ridge ($p = 0.001$), SVM and Random Forest ($p = 0.006$), and XGBoost and LassoCV ($p = 0.006$). No significant differences ($p \geq 0.05$) were found between AdaBoost and LassoCV, AdaBoost and Ridge, AdaBoost and SVM, decision tree and KNN, decision tree and Random Forest, decision tree and SVM, decision tree and XGBoost, KNN and LassoCV, KNN and Random Forest, KNN and Ridge, KNN and SVM, KNN and XGBoost, LassoCV and Ridge, LassoCV and SVM, Random Forest and XGBoost, Ridge and SVM, Ridge and XGBoost, and SVM and XGBoost.

The parameters of the RF model are tuned using Optuna. Optimal hyperparameters are reported in Table 2. Figure 7 shows the optimization process. The number of estimators is the number of decision trees in the RF model, maximum depth is the maximum depth of each decision tree, and minimum sample leaf is the smallest number of samples that should be present in the leaf node after splitting a node. This hyperparameters are reported in Table 4.

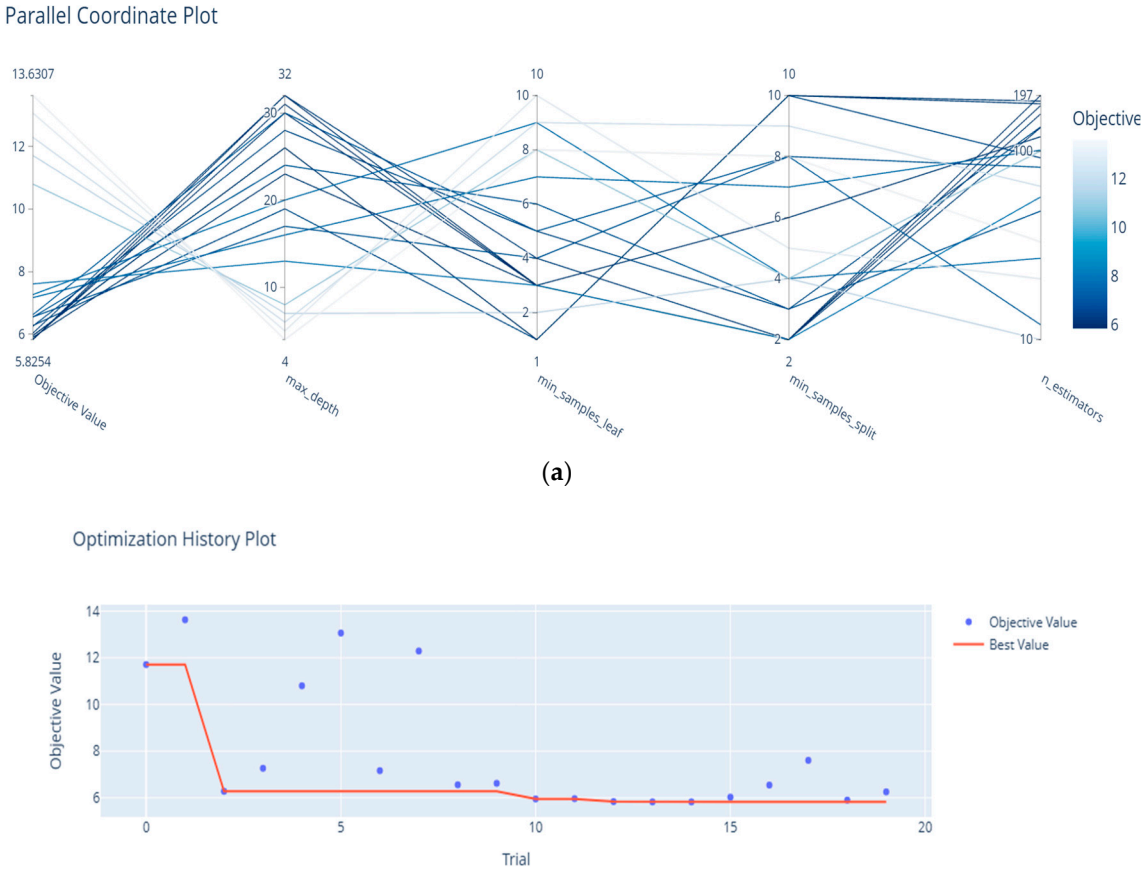


Figure 7. Bayesian Optimization for hyperparameter tuning: (**a**) Parallel coordinates shaded with the objective value; the objective for the optimization is the RMSE value. (**b**) The evolution of the RMSE over the number of iterations.

Table 4. Optimal hyperparameters measured using Bayesian Optimization using Optuna.

Hyperparameter	Number of Estimators	Maximum Depth	Minimum Sample Split	Minimum Leaves Per Sample
Value	178	26	10	1

3.3. Classification

The classification models categorize IG values into normal, high, and low. Most IG values belong to the normal class in this dataset. To overcome this class imbalance, the number of samples per class is stratified by downsampling the normal class to 2500 samples. The total number of samples is 7500 (2500 samples per class). A total of 70% of the samples are used in training the ML models, while 30% are used for testing the performance of those models (RF, DT, SVM, LDA, KNN, GNB, Ridge, AdaBoost and XGBoost). After identifying the best performing model type (RL), its hyperparameters are tuned using Bayesian Optimization.

Figure 8 represents the performance of classification models in terms of accuracy, precision, recall and F1-score. Table 5 presents the comparison of different classification models in glucose prediction.

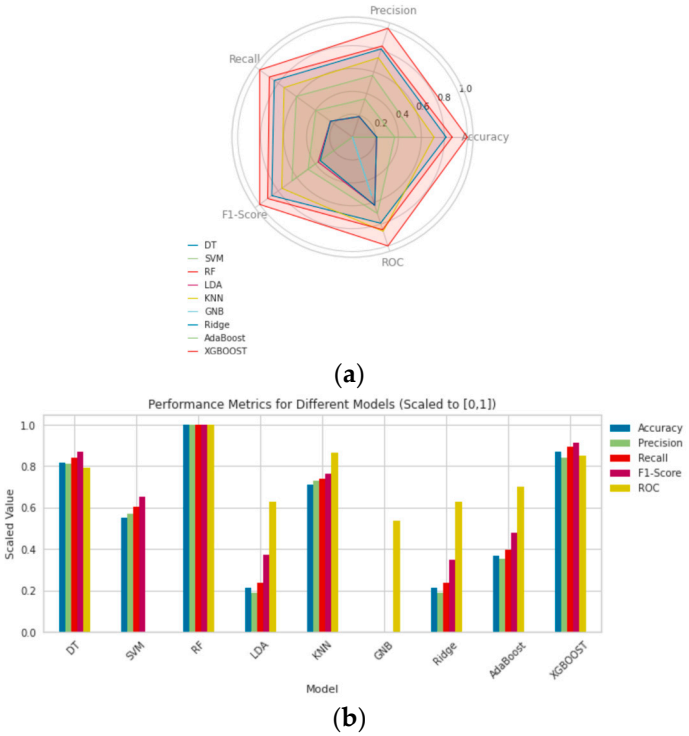


Figure 8. Comparison of the performance metrics of classification models: (**a**) Normalized spider plot for different performance metrics of classification; (**b**) bar plot for performance measures of different models.

Table 5. Performance metrics of classification models.

Model	Accuracy (%)	Precision	Recall	F1-Score	ROC
DT	71	71	71	0.71	0.78
SVM	61	62	62	0.61	0.25
RF	78	78	77	0.77	0.92
LDA	48	48	48	0.48	0.67
KNN	67	68	67	0.66	0.83
GNB	40	41	39	0.31	0.61
Ridge	48	48	48	0.47	0.67
AdaBoost	54	54	54	0.53	0.72
XGBoost	73	72	73	0.73	0.82

To make sure that the difference between the metrics for each model is significant, a Friedman test is carried out for each metric. The results for each metric and model are reported in the Supplementary Material. An example of this analysis is shown here for reference. The Friedman statistic for values of accuracy for all the models for all folds is 78.33 with ($p = 1.05 \times 10^{-13}$), showing that the difference is significant. To understand for which models the comparison is significant, a Nemenyi post hoc (Figure 9) analysis for the results is conducted and plotted as a heatmap.

The Nemenyi post hoc test results for mean absolute error (MAE) reveal significant and non-significant differences in model performance. Significant differences ($p < 0.05$) were observed between AdaBoost and XGBoost ($p = 0.001$), decision tree and GNB ($p = 0.001$), GNB and KNN ($p = 0.006$), Random Forest and GNB ($p = 0.001$), GNB and XGBoost ($p = 0.001$), LDA and RF ($p = 0.002$), and LDA and XGBoost ($p = 0.001$). No significant differences ($p \geq 0.05$) were found between AdaBoost and DT, AdaBoost and GNB, AdaBoost and KNN, AdaBoost and LDA, decision tree and KNN, decision tree and Random Forest, decision tree and SVM, decision tree and XGBoost, KNN and LassoCV, KNN and Random Forest, KNN and Ridge, KNN and SVM, KNN and XGBoost, LassoCV and Ridge, LassoCV and SVM, Random Forest and XGBoost, Ridge and SVM, and SVM and XGBoost.

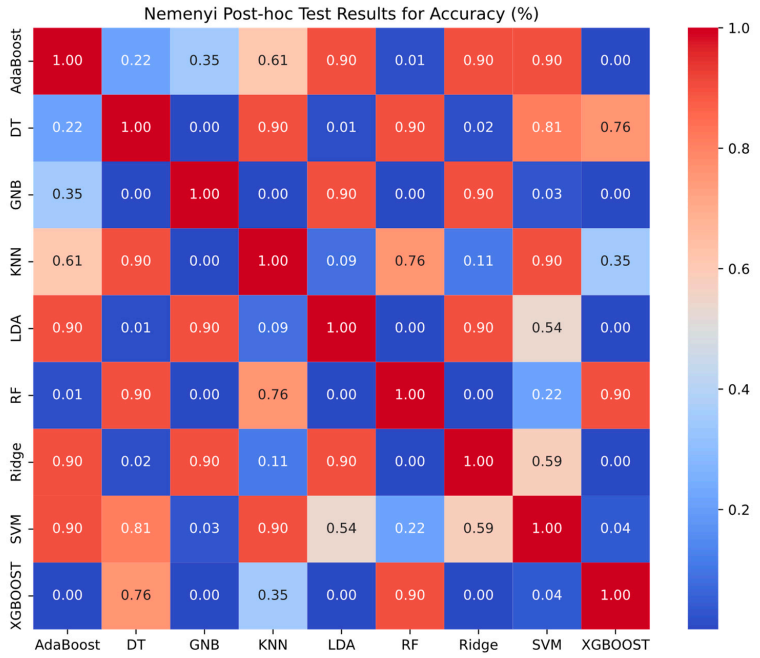
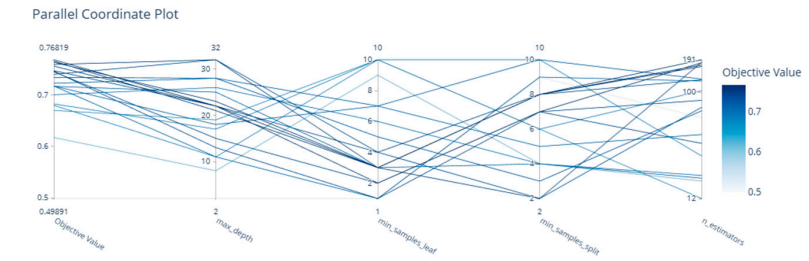


Figure 9. Nemenyi post hoc test results for accuracy (%).

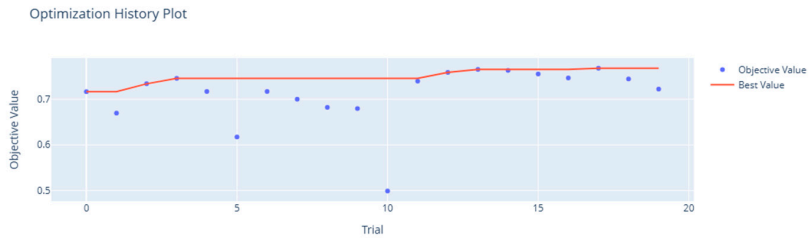
RF outperformed the other models across all the performance metrics. Other performance plots are provided in the Supplementary Material for further clarification. The class prediction error, confusion matrix, ROC curves, and precision recall curves of the classifier were trained using the tuned hyperparameters given in Table 6 in Figure 10.

Table 6. Optimal hyperparameters measured using Bayesian Optimization using Optuna.

Hyperparameter	Number of Estimators	Maximum Depth	Minimum Sample Split	Minimum Leaves Per Sample
Value	130	22	7	2



(**a**)



(**b**)

Figure 10. Bayesian Optimization for hyperparameter tuning: (**a**) Parallel coordinates shaded with the objective value; the objective for the optimization is accuracy. (**b**) The evolution of the accuracy over the number of iterations.

Figure 11 represents the performance of an RF model trained on 70 % data and tested on 30% testing data.

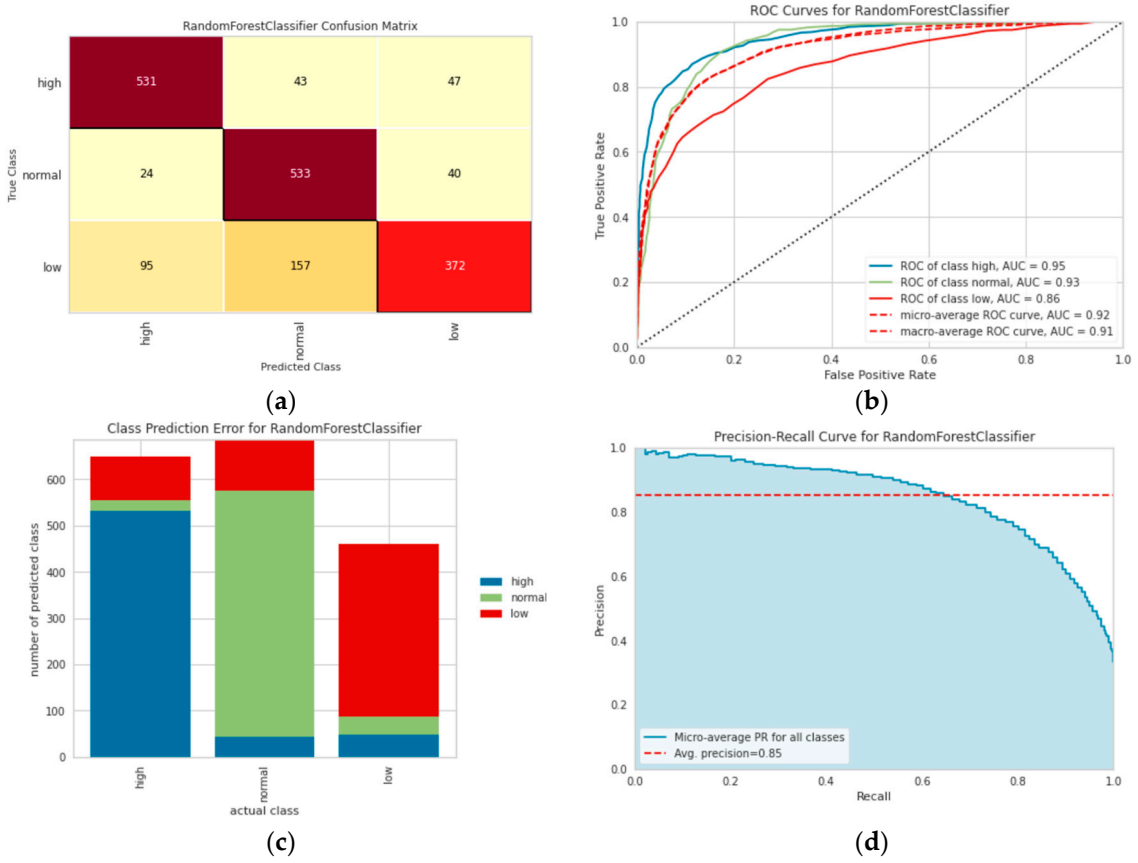


Figure 11. Performance of the tuned Random Forest model on validation data of the balanced dataset: (**a**) Confusion matrix of the tuned RF classifier for validation data of the balanced dataset, (**b**) ROC curves of the tuned RF classifier for validation data of the balanced dataset, (**c**) class prediction error of the tuned RF classifier for validation data of the balanced dataset, and (**d**) precision recall curve of the tuned RF classifier for validation data of the balanced dataset.

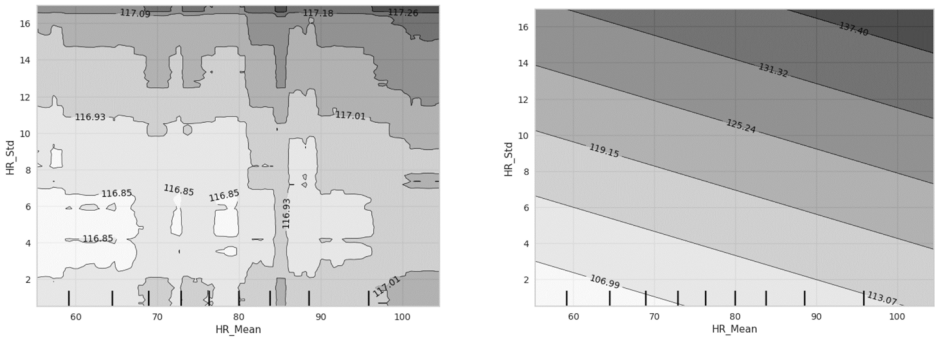
3.4. Model Explanations

The best regression and classification models are explained. To better understand why tree models perform better than kernel-based models such as SVMs, generative models such as GNB, and non-parametric models such as KNNs, partial dependence plots are used to show the complex interaction of the features modeled.

According to the literature, tree-based models are robust to noise and suitable for visualizing feature interactions. Here, we plot the partial dependence plots of two features we believe interact with each other in a complex manner (HR_Mean and HR_Std). As can be seen from the partial dependence plot (PDP) from the RF and the LDA in Figure 12, the PDP of the RF model represents a complex relationship, whereas for the linear model, the PDP shows a linear relationship, resulting in lower performance metrics for regression.

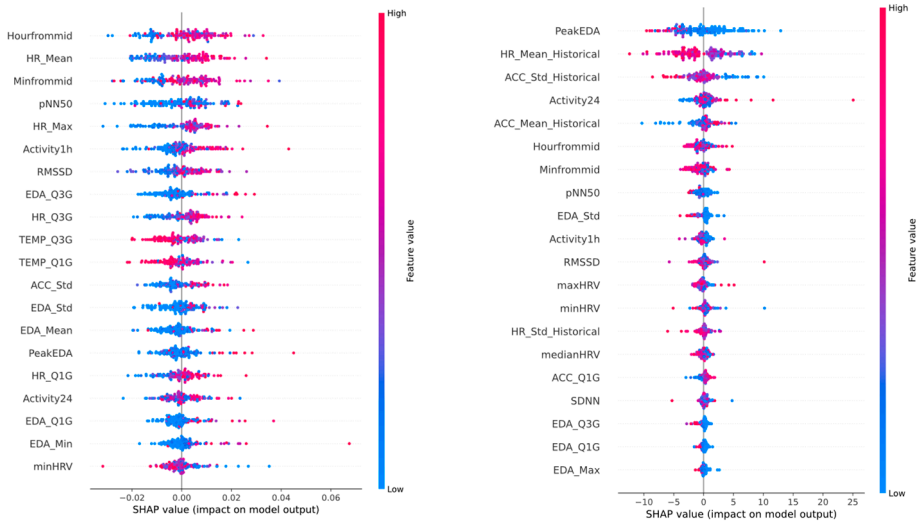
The SHAP plot in Figure 13a reveals the distribution and impact of features on the RF model's predictions for classifying CGM values into high, low, and normal categories. Each feature's influence is illustrated by the spread of dots along the x-axis, indicating the SHAP values. A wider spread of dots signifies a greater variance in the feature's impact across different data points. For instance, Hourfrommid and HR_Mean have a broad range of SHAP values, showing they can significantly sway the predictions towards both high and low CGM classes. The color gradient of the dots, from blue (low feature value) to red (high feature value), further elucidates how different levels of a feature affect the

prediction. This spread and color coding collectively depict the nuanced contributions of each feature, demonstrating the complex interplay between physiological metrics, food intake, and glucose levels. This shows that for the classification problem, the circadian features are relatively more important than other features. This is consistent with earlier works [13,15,18].



(**a**) (**b**)

Figure 12. Comparison of PDP plots for standard deviations of heart rate and mean heart rate: (**a**) The RF PDP captures a complex relationship, resulting in a higher accuracy; (**b**) the LDA assumes a linear relationship, resulting in a lower performance.



(**a**) (**b**)

Figure 13. SHAP summary plots for classification and regression. (**a**) SHAP values for classification, (**b**) SHAP values for regression.

Figure 11b demonstrates how each feature contributes to the regression model's prediction of CGM values, highlighting the importance and influence of physiological metrics and activity levels captured by the Empatica E4 smart watch, as well as historical data. Key features like '*PeakEDA*', '*HR_Mean_Historical*', and '*ACC_Std_Historical*' exhibit a broad range of SHAP values, indicating significant variability in their impact on CGM predictions. For instance, high values of '*PeakEDA*' tend to push predictions higher, while low values often push predictions lower. Similarly, '*HR_Mean_Historical*' consistently shows a positive impact on CGM values, with high feature values leading to higher predictions. The varied influence of '*ACC_Std_Historical*' and 'Activity24' further underscores the complexity and interplay of factors affecting glucose levels. This comprehensive visualization of feature contributions provides valuable insights into the model's decision-making process.

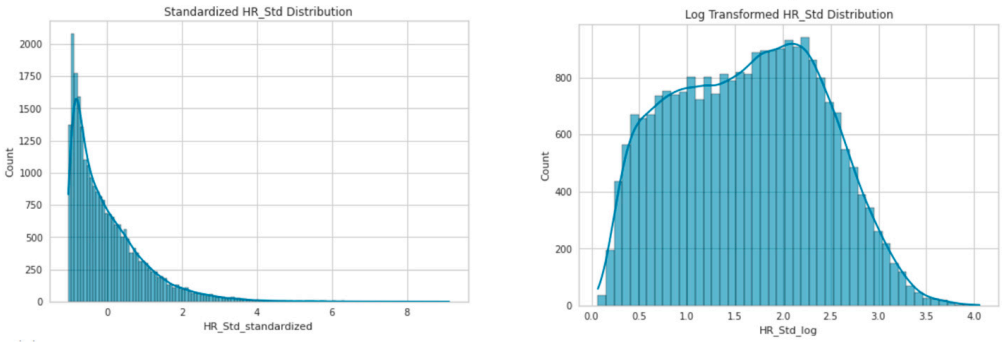
4. Discussion

For both classification and regression tasks RF, has a superior performance, while the other tree-based model, DT, is not that far behind. XGBoost, which is also a tree model, performs well in both the tasks as well. Tree models are known to perform well in cases when there are nonlinear relationships. KNNs [30] and tree-based models (RF, DT, and XGBoost) are both equipped to handle nonlinear relationships between the data [31–33].

Gaussian Naïve Bayes (GNB) is best suited for datasets with conditionally independent features, linear relationships, and normalized data [34]. While it assumes conditional feature independence, which simplifies computation, this assumption can limit its performance with more complex, dependent features. The feature interactions that GNB can model are also linear, which is not the case for the complex relationship between many of these variables—for example, the interaction between the rolling sum of carbohydrates consumed and hours from midnight. Food can be consumed at the beginning of the day, which is less far from midnight, but that potentially increases CGM values in the subsequent windows of prediction.

KNN relies on distance measurements, which can be less effective with mixed data types and skewed distributions. Since most of these variables are skewed, KNN underperforms the tree models but outperforms GNB.

Support vector machines (SVMs) can capture complex feature relationships but often require extensive feature engineering to perform optimally [35]. For instance, standardization and transformations such as taking the logarithm of skewed features can improve SVM performance [35]. Figure 14 illustrates the impact of standardization on the skewness of the heart rate (HR) standard deviation: normalization using the Z-score does not eliminate skewness, but applying a log transformation makes the changes more prominent. To illustrate the impact of skewness on model performance, models sensitive to skewness (SVM and GNB) are trained on the training set. The accuracy calculations on the validation set reveals that taking a log of HR_Std increases the accuracy of GNB (40% to 44%) and SVM (61% to 64%), although that increase is small.



(**a**) (**b**)

Figure 14. Comparison of HR standard deviation skewness. (**a**) Normalization of HR values using the Z-score does not eliminate the skewness of the data. (**b**) Taking a log of this value makes the changes more prominent.

Tree-based models are particularly adept at handling skewed features and mixed data types [36]. Of the tree models, ensemble methods like RF and XGBoost outperform other models by better handling outliers. The presence of influential outliers in the data is evidenced by the Cook's distance plot in Figure 15. RF's better performance over XGBoost in this study may be due to the noise in the wearable data, suggesting that quality metrics should be used during data collection to minimize noise influence.

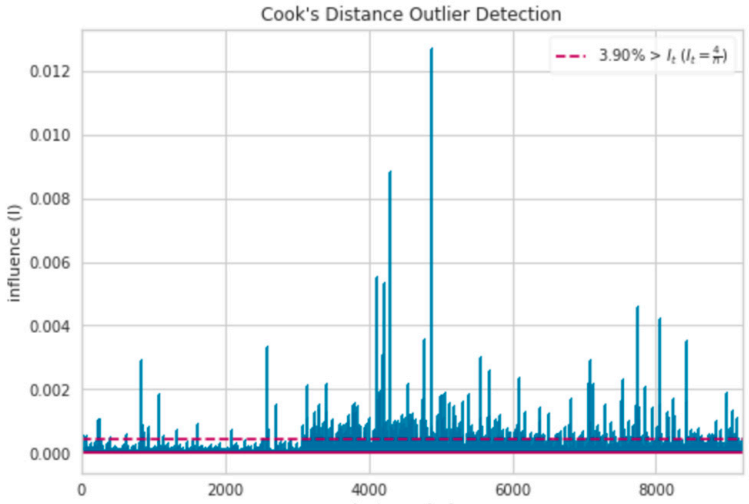


Figure 15. Cook's distance plot shows influential outliers.

Similarly, tree-based models are better at learning mixed data types (categorical and numerical) than KNN models, because the latter rely on distance measurements. Amongst the three models, ensemble methods (RF and XGBOOST) performed better, because of their better handling of outliers.

The preprocessing, feature engineering, and hyperparameter tuning in this work results in models with superior performance. Table 7 summarizes the comparison of the best performing models in this work whereas Table 8 presents the comparison of trained models with earlier works.

Table 7. Performance of the RF model trained on tuned hyperparameters.

Class	Precision	Recall	F1-Score	Accuracy
High	0.80	0.85	0.82	0.80
Normal	0.71	0.89	0.79	0.70
Low	0.81	0.58	0.67	0.80

Table 8. Comparison of trained models with similar work.

Study	Type	Model	Performance
[18]	Regression	DT	MSE = 21.22 $\pm$ 4.14 mg/dL
This Work	Regression	RF	MSE = 9.04 mg/dL
[14]	Classification	DT	AUROC = 0.72
This work	Classification	RF	AUROC = 0.86

Tree-based models are effective in predicting glucose levels both for classification and regression. The explanations also provide a basis for future model development and feature engineering; for example, it is worthwhile to convert the skewed features to the log of these features or using PDP plots to engineer new features.

This study has potential limitations. The values of ground truth measured using a Dexcom sensor that define the labels are affected by motion [37]. However, the study clearly states the values it predicts. The models that are compared in this study are compared based on the features that have been engineered. Future feature engineering or postprocessing of the features, such as taking a log of the features, can affect the performance of different model types.

These results underscore the importance of using robust ensemble methods for glucose level prediction, suggesting that these models can significantly improve the accuracy and reliability of real-time glucose-monitoring systems. In practical terms, the enhanced performance of these models can lead to better glucose management and improved health

outcomes for individuals with diabetes. Additionally, this study's insights into feature engineering, such as the benefits of log transformation for skewed data, provide a valuable framework for developing more accurate predictive models in future research. Implementing these findings in healthcare settings could facilitate more personalized and effective diabetes management, ultimately contributing to better patient care and quality of life.

5. Conclusions

This study has demonstrated that tree-based models, particularly Random Forest (RF) and decision tree (DT), exhibit superior performance in predicting Interstitial Glucose (IG) levels from wrist-worn wearable sensor data. These models outperformed other machine learning (ML) models in both classification and regression tasks, achieving higher accuracy, precision, recall, and F1-scores for classification, as well as lower root mean square error (RMSE) and higher R-squared values for regression.

In conclusion, the findings of this study highlight the potential of using wearable sensor data and tree-based ML models to provide insights into metabolic health and disease states. Future work should focus on improving data quality through noise reduction techniques and exploring advanced feature engineering methods, such as partial dependence plots (PDP) and transforming skewed features. Implementing these improvements can further enhance the accuracy and reliability of IG level predictions, ultimately contributing to the better management of metabolic syndromes and diseases.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics13163192/s1 Table S1: Demographics of the participants in the dataset, Table S2: Features Definition, Figure S1: Performance comparison of the regression models, Figure S2: QQ plot of the Regression models, Figure S3: Prediction Error of the Regression Models, Figure S4: Regression Model performance (kfold k = 10), Figure S5: t-SNE plot of the features, Figure S6: Confusion Matrix of the Classification Models, Figure S7: Classification Model performance (kfold k = 10), Figure S8: Nemenyi Post-hoc results for MAE, Figure S9: Nemenyi Post-hoc results for Adjusted R2, Figure S10: Nemenyi Post-hoc results for Explained Variance, Figure S11: Nemenyi Post-hoc results for MSLE, Figure S12: Nemenyi Post-hoc results for MAE, Figure S13: Nemenyi Post-hoc results for Accuracy, Figure S14: Nemenyi Post-hoc results for precision, Figure S15: Nemenyi Post-hoc results for F-1 Score, Figure S16: Nemenyi Post-hoc results for Recall, Figure S17: Nemenyi Post-hoc results for ROC-AUC.

Author Contributions: Conceptualization, H.A., I.K.N., S.M., M.N.A. and D.W.; methodology, I.K.N., M.N.A. and S.M.; software, H.A., I.K.N., M.N.A. and S.M.; validation, I.K.N., D.W. and S.M.; formal analysis, H.A., I.K.N., M.N.A. and S.M.; investigation, H.A.; resources, I.K.N. and D.W.; data curation, H.A.; writing—original draft preparation, H.A.; writing—review and editing, H.A., M.N.A., S.M. and D.W.; visualization, H.A.; supervision, I.K.N., S.M. and D.W.; project administration, D.W.; funding acquisition, I.K.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data can be made available on reasonable request. The dataset is available at https://physionet.org/content/big-ideas-glycemic-wearable/1.1.1/ (accessed on 6 June 2024).

Acknowledgments: The authors acknowledge support by the New Zealand College of Chiropractic PhD Scholarship (funding number: 20126384), and computational support is provided by New Zealand E Science Infrastructure (NESI grant number AUT 03802).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ML	Machine Learning
IG	Interstitial Glucose
CGM	Continuous Glucose Monitoring
ACC	Accelerometer
BVP	Blood Volume Pulse
EDA	Electrodermal Activity
HR	Heart Rate
IBI	Inter-Beat Interval
DT	Decision Tree
SVR	Support Vector Regression
RF	Random Forest
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbors
GNB	Gaussian Naïve Bayes
LassoCV	Lasso Cross-Validation
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
MSLE	Mean Square Logarithmic Error
ROC	Receiver Operator Curve
AUROC (Area Under Receiver Operator Curve)	Area Under Receiver Operator Curve
PDP	Partial Dependence Plot
SHAP	Shapley Additive Explanations

References

1. Maged, Y.; Atia, A. The Prediction Of Blood Glucose Level By Using The ECG Sensor of Smartwatches. In Proceedings of the 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 8–9 May 2022; pp. 406–411. [CrossRef]
2. Bent, B.; Cho, P.J.; Wittmann, A.; Thacker, C.; Muppidi, S.; Snyder, M.; Crowley, M.J.; Feinglos, M.; Dunn, J.P. Non-invasive wearables for remote monitoring of HbA1c and glucose variability: Proof of concept. *BMJ Open Diabetes Res. Care* **2021**, *9*, e002027. [CrossRef]
3. International Diabetes Federation. IDF Diabetes Atlas Tenth Edition 2021. Available online: https://diabetesatlas.org/ (accessed on 3 June 2024).
4. Aguilar, M.; Bhuket, T.; Torres, S.; Liu, B.; Wong, R.J. Prevalence of the Metabolic Syndrome in the United States, 2003–2012. *JAMA* **2015**, *313*, 1973–1974. [CrossRef] [PubMed]
5. CDC. National Diabetes Statistics Report, Diabetes. Available online: https://www.cdc.gov/diabetes/php/data-research/index.html (accessed on 3 June 2024).
6. Grundy, S.M.; Brewer, H.B., Jr.; Cleeman, J.I.; Smith, S.C., Jr.; Lenfant, C. Definition of Metabolic Syndrome: Report of the National Heart, Lung, and Blood Institute/American Heart Association conference on scientific issues related to definition. *Circulation* **2004**, *109*, 433–438. [CrossRef] [PubMed]
7. Ervin, R.B. Prevalence of metabolic syndrome among adults 20 years of age and over, by sex, age, race and ethnicity, and body mass index: United States, 2003–2006. *Natl. Health Stat. Rep.* **2009**, *13*, 1–7.
8. Ford, E.S.; Li, C.; Sattar, N. Metabolic syndrome and incident diabetes: Current state of the evidence. *Diabetes Care* **2008**, *31*, 1898–1904.
9. Jarvis, P.R.; Cardin, J.L.; Nisevich-Bede, P.M.; McCarter, J.P. Continuous glucose monitoring in a healthy population: Understanding the post-prandial glycemic response in individuals without diabetes mellitus. *Metabolism* **2023**, *146*, 155640. [CrossRef]
10. CDC. Prediabetes—Your Chance to Prevent Type 2 Diabetes, Diabetes. Available online: https://www.cdc.gov/diabetes/prevention-type-2/prediabetes-prevent-type-2.html (accessed on 3 June 2024).
11. Zoungas, S.; Chalmers, J.; Ninomiya, T.; Li, Q.; Cooper, M.E.; Colagiuri, S.; Fulcher, G.; De Galan, B.E.; Harrap, S.; Hamet, P.; et al. Association of HbA1c levels with vascular complications and death in patients with type 2 diabetes: Evidence of glycaemic thresholds. *Diabetologia* **2012**, *55*, 636–643. [CrossRef]
12. Beck, R.W.; Bergenstal, R.M.; Riddlesworth, T.D.; Kollman, C.; Li, Z.; Brown, A.S.; Close, K.L. Validation of Time in Range as an Outcome Measure for Diabetes Clinical Trials. *Diabetes Care* **2018**, *42*, 400–405. [CrossRef]
13. Cho, P.; Kim, J.; Bent, B.; Dunn, J. BIG IDEAs Lab Glycemic Variability and Wearable Device Data. *PhysioNet* **2023**.

14. Ali, H.; Madanian, S.; Malik, N.; White, D.; Russel, B.K.; Niazi, I.K. Prediction of Interstitial Glucose Levels Through Wearable Sensors Using Machine Learning. In *2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*; IEEE: New York, NY, USA, 2023; pp. 1–6. [CrossRef]
15. Adams, D.; Nsugbe, E. Predictive Glucose Monitoring for People with Diabetes Using Wearable Sensors. *Eng. Proc.* **2021**, *10*, 20. [CrossRef]
16. Ali, H.; Madanain, S.; White, D.; Akhter, M.N.; Niazi, I.K. From wearable activity trackers to Interstitial Glucose: Data to Insight-A proposed scientific journey. In Proceedings of the 2024 Australasian Computer Science Week, Sydney, Australia, 29 January–1February 2024; pp. 61–64. [CrossRef]
17. Zahedani, A.D.; McLaughlin, T.; Veluvali, A.; Aghaeepour, N.; Hosseinian, A.; Agarwal, S.; Ruan, J.; Tripathi, S.; Woodward, M.; Hashemi, N.; et al. Digital health application integrating wearable data and behavioral patterns improves metabolic health. *NPJ Digit. Med.* **2023**, *6*, 216. [CrossRef] [PubMed]
18. Bent, B.; Henriquez, M.; Dunn, J.P. Cgmquantify: Python and R Software Packages for Comprehensive Analysis of Interstitial Glucose and Glycemic Variability from Continuous Glucose Monitor Data. *IEEE Open J. Eng. Med. Biol.* **2021**, *2*, 263–266. [CrossRef] [PubMed]
19. Bent, B.; Cho, P.J.; Henriquez, M.; Wittmann, A.; Thacker, C.; Feinglos, M.; Crowley, M.J.; Dunn, J.P. Engineering digital biomarkers of interstitial glucose from noninvasive smartwatches. *NPJ Digit. Med.* **2021**, *4*, 89. [CrossRef] [PubMed]
20. Qi, W.; Wang, N.; Su, H.; Aliverti, A. DCNN based human activity recognition framework with depth vision guiding. *Neurocomputing* **2021**, *486*, 261–271. [CrossRef]
21. Zhao, J.; Lv, Y.; Zeng, Q.; Wan, L. Online Policy Learning-Based Output-Feedback Optimal Control of Continuous-Time Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *71*, 652–656. [CrossRef]
22. Lehmann, V.; Föll, S.; Maritsch, M.; van Weenen, E.; Kraus, M.; Lagger, S.; Odermatt, K.; Albrecht, C.; Fleisch, E.; Zueger, T.; et al. Noninvasive Hypoglycemia Detection in People With Diabetes Using Smartwatch Data. *Diabetes Care* **2023**, *46*, 993–997. [CrossRef]
23. Huang, X.; Schmelter, F.; Seitzer, C.; Martensen, L.; Otzen, H.; Piet, A.; Witt, O.; Schröder, T.; Günther, U.; Grzegorzek, M.; et al. From Data to Insight: Predicting Interstitial Glucose in Healthy Cohort with Non-invasive Sensor Technology and Machine Learning. *arXiv* **2023**. [CrossRef]
24. Optuna—A Hyperparameter Optimization Framework. Optuna. Available online: https://optuna.org/ (accessed on 8 June 2024).
25. Liang, Y.; Elgendi, M.; Chen, Z.; Ward, R. An optimal filter for short photoplethysmogram signals. *Sci. Data* **2018**, *5*, 180076. [CrossRef] [PubMed]
26. Nabian, M.; Yin, Y.; Wormwood, J.; Quigley, K.S.; Barrett, L.F.; Ostadabbas, S. An Open-Source Feature Extraction Tool for the Analysis of Peripheral Physiological Data. *IEEE J. Transl. Eng. Heal. Med.* **2018**, *6*, 2800711. [CrossRef]
27. Lam, B.; Catt, M.; Cassidy, S.; Bacardit, J.; Darke, P.; Butterfield, S.; Alshabrawy, O.; Trenell, M.; Missier, P. Using Wearable Activity Trackers to Predict Type 2 Diabetes: Machine Learning–Based Cross-sectional Study of the UK Biobank Accelerometer Cohort. *JMIR Diabetes* **2021**, *6*, e23364. [CrossRef]
28. Interbeat Interval Filtering. Available online: https://arxiv.org/html/2406.01846v1#S3 (accessed on 22 July 2024).
29. Chandra, V.; Priyarup, A.; Sethia, D. Comparative Study of Physiological Signals from Empatica E4 Wristband for Stress Classification. In *Advances in Computing and Data Sciences*; Singh, M., Tyagi, V., Gupta, P.K., Flusser, J., Ören, T., Sonawane, V.R., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 218–229. [CrossRef]
30. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
31. Hastie, T.; Tibshirani, R.; Friedman, J. Random Forests. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Hastie, T., Tibshirani, R., Friedman, J., Eds.; Springer: New York, NY, USA, 2009; pp. 587–604. [CrossRef]
32. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
33. Hastie, T.; Tibshirani, R.; Friedman, J. Boosting and Additive Trees. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Hastie, T., Tibshirani, R., Friedman, J., Eds.; Springer: New York, NY, USA, 2009; pp. 337–387. [CrossRef]
34. Zhang, H. The Optimality of Naive Bayes. AAAI. Available online: https://aaai.org/papers/flairs-2004-097/ (accessed on 5 June 2024).
35. Wang, H.; Hu, D. Comparison of SVM and LS-SVM for regression. In *2005 International Conference on Neural Networks and Brain*; IEEE: New York, NY, USA, 2005; pp. 279–283.
36. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Salzberg, S.L. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Mach. Learn.* **1994**, *16*, 235–240. [CrossRef]