Article:
Nemat, H., Khadem, H., Eissa, M.R. et al. (2022) Blood glucose level prediction : advanced deep-ensemble learning approach. IEEE Journal of Biomedical and Health Informatics, 26 (6). pp. 2758-2769. ISSN: 2168-2194

https://doi.org/10.1109/JBHI.2022.3144870

Blood Glucose Level Prediction: Advanced Deep-Ensemble Learning Approach

Hoda Nemat, Heydar Khadem, Mohammad R. Eissa, Jackie Elliott, and Mohammed Benaissa, *Senior Member, IEEE*

***Abstract*— Optimal and sustainable control of blood glucose levels (BGLs) is the aim of type-1 diabetes management. The automated prediction of BGL using machine learning (ML) algorithms is considered as a promising tool that can support this aim. In this context, this paper proposes new advanced ML architectures to predict BGL leveraging deep learning and ensemble learning. The deep-ensemble models are developed with novel meta-learning approaches, where the feasibility of changing the dimension of a univariate time series forecasting task is investigated. The models are evaluated regression-wise and clinical-wise. The performance of the proposed ensemble models are compared with benchmark non-ensemble models. The results show the superior performance of the developed ensemble models over developed non-ensemble benchmark models and also show the efficacy of the proposed meta-learning approaches.**

***Index Terms*— Blood glucose level, Deep learning, Diabetes mellitus, Meta-learning, Ensemble learning, Time series forecasting.**

I. Introduction

Effective management of type 1 diabetes mellitus (T1DM) reduces the associated complications [1]. The primary goal in T1DM management is to maintain the blood glucose level (BGL) within a target range [2]. BGL prediction models can contribute to better glycaemic control. These models estimate future BGLs utilizing the current and past information and provide early warnings concerning inadequate glycaemic control [3]. Additionally, the current continuous glucose monitoring sensors measure glucose in the interstitial fluid rather than the blood stream, which may introduce a delay, particularly when the glucose levels are changing rapidly [4]. Therefore, BGL measurement may need to rely on models that can predict the glucose level accurately. This accurate prediction becomes even more apparent when used in the artificial pancreas [5].

Machine learning (ML) is a widely used approach for developing time-series forecasting models for BGL [6], [7]. Despite many studies performed to predict BGL so far, there is a lack of decisive models. Hence, developing more reliable models is still desirable [8]. Also, using different datasets or input features in the literature has made the performance comparison of different models difficult. Hence, making fair comparisons, is a valuable area of research [9].

Among ML approaches, deep learning models could be more effective in detecting complicated systems' dynamics [10]. Ensemble learning is an advanced strategy that can enhance the performance of ML tasks by combining multiple models. Using deep and ensemble learning together has emerged in recent years as an attractive strategy due to the growth of computing capability. Recently a number of studies have been performed combining deep learning models and the ensemble learning concept in the BGL prediction field [11]–[13]. However, there is still a lack of a comprehensive investigation of deep and ensemble learning capability and comparison with benchmark models.

This work proposes three new advanced architectures to predict BGL in people with T1DM leveraging the combination of deep and ensemble learning. Two types of long short-term memory (LSTM) networks, including vanilla LSTM and bidirectional LSTM, postulated as effective approaches in BGL prediction [14]–[16] along with a linear regression model, are considered benchmark BGL prediction models. These benchmark models are also used as base-learners in the ensemble architectures. The developed approaches only use the BGL data from a continuous glucose monitoring sensor; hence the BGL prediction is solved as a univariate time series forecasting task. Three meta-learning approaches are used for output fusion of base-learners in the advanced architectures. One of them is based on stacked learning [17], an established concept in ensemble learning. The other two, named Multivariate and Subsequences, are novel approaches proposed in this work. In the Multivariate approach, the output vectors of base-learners were considered multivariate input for train ing a meta-learner. Hence, the univariate time series forecasting was considered as multivariate time series forecasting in the meta-learning. In the Subsequences approach, output vectors of base-learners were considered as different subsequences. This resulted in configuring the univariate time series forecasting as a two-dimensional data analysis.

The outline of this paper is as follows. The state-of-the-art in the area of BGL prediction is discussed in Section II. Section III describes the publicly available Ohio dataset that is used for model development and evaluation in this research. Section IV presents the developed methodologies for the BGL prediction task. The proposed models are experimentally validated and discussed in Section V. Section VI draws the conclusion.

II. Related works

Based on knowledge requirement, BGL prediction models could be classified into three main groups of physiological (extensive knowledge), hybrid (intermediate knowledge) and data-driven (black-box approach) models [3]. Data-driven models establish the relationship between the present and past BGL and future values. ML and classical time-series approaches are widely used for building these models [3], [18], [19]. The following section briefly discusses some recent works for BGL prediction.

In their study, Mirshekarian et al. [20] investigated several experiments for BGL prediction using continuous glucose monitoring (CGM), insulin, meal, and activity data from simulated and real T1DM datasets in the prediction horizons of up to one hour. They used the data of two diabetes simulators (i.e., AIDA and UVa/Padova) as synthetic datasets and the Ohio T1DM dataset as the real one

This work is not associated with funding Agency. (Corresponding author: Hoda Nemat.)

Hoda Nemat, Heydar Khadem, Mohammad R. Eissa, and Mohammed Benaissa are with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, S1 4DE, U.K. (e-mail: hoda.nemat@sheffield.ac.uk; h.khadem@sheffield.ac.uk; m.eissa@sheffield.ac.uk; m.benaissa@sheffield.ac.uk).

Jackie Elliott is with the Department of Oncology and Metabolism, University of Sheffield, Sheffield S10 2RX, U.K., and also with the Sheffield Teaching Hospitals, Diabetes and Endocrine Centre, Northern General Hospital, Sheffield S5 7AU, U.K. (e-mail: j.elliott@sheffield.ac.uk).

and developed a new memory-augmented LSTM for the time series forecasting task. They also considered an autoregressive integrated moving average model as a baseline and observed that the LSTM model meaningfully outperformed the baseline model. Based on the comparison results of the experiments for in-silico and real data, they found that the designed neural attention module improved prediction performance in synthetic data, although it failed to improve it in the real data. Contrarily, using day time as an extra input of the LSTM model enhanced BGL prediction performance only in real data. They concluded that the attitude of synthetic and real data is not always the same. Finally, by examining the LSTM on real data, they found that adding skin conductance and heart rate to BG, insulin, meal, and time could improve prediction performance.

Similarly, Martinsson et al. [21] presented an end-to-end system for predicting BGL in the prediction horizons of 30 and 60 minutes. To develop and evaluate their system, they used the Ohio T1DM dataset by considering the history of BGL as input and proposed a recurrent neural network (RNN) model for the regression task. They also estimated certainty for the predicted values, and uncertainty was the standard deviation (SD) of the prediction achieved by a parameterised univariate Gaussian distribution over the output. The mean and SD of the root mean square error (RMSE) over six T1DM patients using their proposed model was 18.867 ± 1.794 mg/dl and 31.403 ± 2.078 mg/dl for the 30- and 60-minute prediction horizons, respectively.

Moreover, Xie and Wang [9] evaluated a set of well-known ML approaches for predicting the BGL of people with T1DM using the data of the BGL, insulin injected, carbohydrate intakes, and exercises as inputs measured in the Ohio dataset. Furthermore, a classical autoregression with exogenous inputs (ARX) model was benchmarked against 10 different ML models. These models included Elastic-Net, Lasso, Huber, Random-Forest, Gradient-Boosting-Trees, Ridge, and support vector (with both linear and radial basis kernels) regressions along with two deep learning models (i.e., vanilla LSTM and temporal convolution networks). Their results showed that the ARX model and Ridge regression had the lowest average RMSE (19.48 ± 2.91 mg/dl) in the prediction horizon of 30 minutes for BGL prediction. However, the ARX model had worse robustness compared to the NNs. It over-predicted peaks while under-predicting valleys.

Jeon et al. [22] performed another investigation for predicting BGL in the prediction horizon of 30 minutes using the Ohio dataset. In their previous work [23], it was postulated that a gradient-boosted regression tree model outperformed a random forest regression and an LSTM model in predicting BGLs. They further found that the missing data of the sensors had been a challenging factor in BGL prediction. Furthermore, they explored the impact of 19 physiological and monitoring variables provided in the Ohio dataset. By grouping the variables into four classes and creating 15 combinations of these groups, they concluded that using all feature classes could benefit BGL prediction by evading probably lost information. They also examined 11 different imputation techniques and validated their methodology using two traditional train-test and online settings. They then selected five missing data imputation approaches to apply, including linear, spline, Stineman, Kalman, and the last-observed-carried-forward interpolations. They finally combined the predictions to generate an ensemble model and demonstrated that the ensemble model made better BGL predictions in both settings compared to the individual predictive models.

Zhu et al. [24] proposed a model using dilated RNNs for predicting BGL in the prediction horizon of 30 minutes. After investigating vanilla RNN, LSTM, and GRU cells, they selected a vanilla RNN cell to build the final model. The model was trained by BGL history data, bolus, and meal intake of the Ohio T1DM dataset and data from the UVa/Padova simulator. Overall, they observed that the performance of the proposed model for BGL prediction in the synthetic dataset was better compared to the Ohio dataset. In addition, their results showed that preprocessing steps such as interpolation and extrapolation could decrease the average of RMSE by 0.3 mg/dl. Applying transfer learning to exploit other subjects' data was useful for one subject with various missing data. Their model had a smaller RMSE compared to autoregressive, support vector regression, and conventional NNs. Hence, they expressed that the dilated RNN model could improve the performance of BGL prediction and suggested adding the exercise data to the input for future investigation.

Guemes et al. [25] introduced a data-driven approach for predicting nocturnal adverse glycaemia to alarm people with T1DM to take precautionary actions. To generate and evaluate their methodology, they used the Ohio dataset by considering CGM data, carbohydrate intake, and bolus during day time as inputs for the models. Accordingly, they developed three classification methodologies for predicting the occurrence of hypoglycaemia, normoglycaemia, and hyperglycaemia during bedtime by investigating several well-known binary classification algorithms and then presented the feasibility of the overnight glycaemia prediction. Based on their report, the extended tree classifier and support vector machine performed better at nocturnal normoglycaemia and hypoglycaemia prediction, while the random forest classifier predicted better hyperglycaemia. They further suggested applying state-of-the-art classification approaches such as LSTM networks using a larger dataset as future work.

Rodriguez et al. [26] to enhance the management of T1DM, analysed extensive glycemia-related data of 25 people with T1DM collected from a monitoring period of 14 days within the context of the internet of things. To model BGL through patterns' identification, glycaemia, insulin, meal, steps count, heart rate, and sleep data were collected via various biosensors. The authors, to model glycaemia, used and compared four techniques; including Gaussian processes with radial basis function kernels, multi-layer perceptron, support vector machines, and bayesian regularised neural networks (BRNN). Their results showed that BRNN offered the best performance on R-squared and RMSE criteria and hence was the most capable technique for BGL modelling.

Although many studies have focused on this area of research, researchers are still exploring various ML approaches for predicting BGL. Moreover, it is worth mentioning that the used Ohio dataset in these works had only six T1DM patients, then some studies used an in-silico dataset along with the Ohio dataset. The current dataset used in this work now includes data collected from 12 T1DM patients, providing a more extensive dataset for developing and evaluating different models.

III. Dataset

To develop the forecasting algorithms, the publicly available Ohio T1DM dataset [27], [28] was used, containing eight weeks' worth of data for 12 people with T1DM. The data of six patients (PID numbers 559, 563, 570, 575, 588, and 591) were released in 2018 for the first BGL prediction challenge [28], followed by releasing data for an additional six patients (PID numbers 540, 544, 552, 567, 584, and 596) regarding the second BGL prediction challenge in 2020 [27].

Data contributors included five females and seven males who were in the age range of 20-80 years old at the time of data collection. Contributors were on insulin pump therapy. There were two separate XML files for each participant for training and testing sets. The last 10 days' worth of data for each contributor was allocated as the testing set, and the rest belonged to the training set.

The original Ohio dataset included CGM data collected every five minutes using the Medtronic Enlite CGM sensor, alongside other

types of data collected from a physical activity band, physiological sensor, and self-reported life-events. In the work presented in this paper we have used CGM data only and this is in line with other work reported in the literature [21], [29]–[33]. The problem in our case is defined as a univariate time series forecasting problem that allows objective evaluation of the proposed methodology whilst still alleviates the complexity and variability associated with using models having varying capabilities for handling multivariate variables.

Table I summarises the information in terms of gender, age, and the number of data points in the Ohio dataset. More information about the dataset can be found in [27], [28].

IV. Methods

This section describes the data preprocessing steps and the developed forecasting models for BGL prediction.

A. Preprocessing

The first step in the preprocessing was to deal with the missing data. Missing data in the training set were imputed using linear interpolation. Also, for the testing set, linear extrapolation was used in order to ensure that future data were not observed by the model and that the model can be used for a real-time application. So, BGL data were converted to a regular time series in 5-minute intervals without any missing data. For example, Figure 1 shows the first 1000 points of original and interpolated training data after data imputation for patient 575.

Another data preprocessing step was to reframe the time series problem to a supervised learning task. In the current work, the task of BGL prediction was approached as a sequence-to-sequence problem, where we looked for predicting the future BGL sequence based on the historical sequence of BGL. To do so, time series data were transformed into samples with lag observations as input and future observations as output. Then a rolling window with four different history lengths of 6, 12, 18, and 24 data points was investigated for the input, which carried the information of 30, 60, 90, and 120 minutes of history, respectively. The associated output was a vector with 6 and 12 data points corresponding to the 30- and 60-minute prediction horizons, respectively.

In the final step of preprocessing, input sequences were scaled to the minimum and maximum value over the entire training set of all subjects.

B. Prediction Models

Linear models could be appropriate tools for BGL prediction tasks as they are simple and only require low-cost computing. On the other hand, LSTM networks, as a type of RNNs, which are suitable for working with sequential data and time series forecasting [34], are effective in predicting BGL [14], [15]. A linear regressor and two different types of LSTM networks were developed in the present work, followed by proposing three different approaches using ensemble learning. The following section presents a naive baseline model, three non-ensemble models, and finally, three ensemble models developed for the BGL prediction task.

TABLE I: Gender, age, and the number of data points in training and testing sets related to the contributors of the Ohio dataset.

PID	Gender	Age	Training samples	Testing samples
540	male	20–40	11947	2896
544	male	40–60	10623	2716
552	male	20–40	9080	2364
559	female	40–60	10796	2514
563	male	40–60	12124	2570
567	female	20–40	10858	2389
570	male	40–60	10982	2745
575	female	40–60	11866	2590
584	male	40–60	12150	2665
588	female	40–60	12640	2791
591	female	40–60	10847	2760
596	male	60–80	10877	2743

Note. PID: Patient ID.

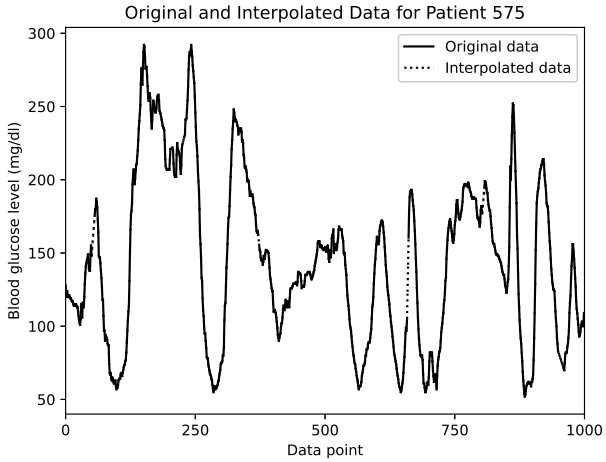


Fig. 1: The first 1000 blood glucose level data points of the training set for patient 575 after interpolation.

1) Baseline model: A baseline model requiring a comparison level of performance is crucial for any time series forecasting task. Being simple, fast, and repeatable are three characteristics of a good baseline model [34]. In this work, a naive baseline model, considering the last available BGL value as the forecast, was used.

2) Non-ensemble models: One linear model and two types of LSTM networks were developed as prediction models.

a) Linear regression: It is a simple and an easy-to-apply model with minimal computational cost. A linear regression model fits a model on the training dataset by minimising the error between real targets and predictions from a linear approximation [35]. Further, a simple linear model was developed for the BGL prediction task, and a linear model was fitted for each data contributor using the input and output vectors of the training set.

b) Vanilla long short-term memory (VLSTM): A vanilla LSTM network [36] with the vector output was used for multi-step ahead forecasting. The model was composed of an `LSTM` layer with 200 units, followed by a `Dense` layer with 100 units and an output layer with the number of future data points as the number of units. To train the model, the MSE was used as the loss function. The epoch size and batch size were considered as 500 and 32, respectively. The callback of `ReduceLROnPlateau` was employed for reducing the learning rate with the initial learning rate of 0.01 by a factor of 0.1 when validation loss has stopped improving for a patient number of 20 epochs. The initialiser, activation function, and optimiser were tuned for each history and horizon, which are discussed in the next section.

c) Bidirectional LSTM (BiLSTM): It is another type of RNNs that can be used for sequence forecasting tasks [34]. A BiLSTM model was implemented using a `Bidirectional LSTM` layer with 200 units, followed by a `Dense` layer with 100 units and an output layer. Similar to the VLSTM model, the loss function, epoch size, and batch size were considered as MSE, 500, and 32 for training the model, respectively. Moreover, `ReduceLROnPlateau`

was employed as the callback with an initial learning rate of 0.01. The initialiser, activation function, and optimiser were optimised for each history and horizon, which is discussed in the following section.

To optimise hyperparameters, the training set was divided into training and validation subsets. For this purpose, the first 80% of data was allocated to the training set, and the following 20% was considered for the validation set. Then, the parameters were fine-tuned by selecting the ones that resulted in the lowest average RMSE over the validation data of 12 subjects. In addition, the hyperparameters were separately optimised for the prediction horizons of 30 and 60 minutes.

The length of the history window was the first parameter to be optimised. To do so, four history window lengths of 30, 60, 90, and 120 minutes which were commonly used values in the literature [21], [24] for tuning, were investigated. These history lengths included 6, 12, 18, and 24 history points, respectively. The two LSTM models were individually fine-tuned for each history length to have a fair comparison between all histories.

To tune the LSTM models, due to computational costs, the epoch size was amended to 200. The initialiser and activation function related to layer configuration and optimiser related to the compilation process were tuned. To tune each parameter, the two other parameters were fixed and the variable was changed over its search space.

To tune the VLSTM model for the prediction horizon of 30 minutes, the kernel initialiser was selected among {Glorot uniform and He uniform} by considering ReLU and Adam as the activation function and the optimiser, respectively. As a result, He uniform and Glorot uniform were selected for the history window of 30 and 60, as well as 90 and 120 minutes, respectively. Then, the search space of {ReLU and Tanh} was explored to tune the activation function by considering the selected initialisers for each history and Adam as the optimiser. It should be noted that ReLU was selected for all histories. The optimiser was the last parameter to be tuned while considering the selected values for the initialiser and activation function. Additionally, for all histories from the search space of {Adam and Adagrad}, Adam optimiser was chosen.

A similar process to that of the VLSTM was repeated for tuning the BiLSTM model. For the prediction horizon of 30 minutes, Glorot uniform was selected for the history windows of 30 and 90 minutes regarding the prediction horizon of 30 minutes, and He uniform was chosen for the history windows of 60 and 120 minutes as the kernel initialiser. For all histories, ReLU and Adam were selected as the activation function and optimiser, respectively.

Similarly, the Glorot uniform was selected for the history windows of 30 and 120 minutes concerning the prediction horizon of 60 minutes in the VLSTM model, followed by choosing He uniform for the history windows of 60 and 90 minutes as the kernel initialiser. Further, ReLU and Adam were selected as the activation function and optimiser, respectively, for all histories. Regarding the BiLSTM model, Glorot uniform was chosen for the history windows of 30 and 120 minutes, and He uniform for history windows of 60 and 90 minutes as the kernel initialiser. Furthermore, ReLU and Adam were selected as the activation function and optimiser for all histories, respectively.

Eventually, using the validation set, the average RMSE over all patients for each history window was calculated and used as a criterion for choosing the history length. Figure 2a illustrates the results of this investigation for the prediction horizon of 30 minutes, and Figure 2b shows those for the prediction horizon of 60 minutes. The final chosen hyperparameters for VLSTM and BiLSTM models regarding both prediction horizons of 30 and 60 minutes are presented in Table II.

According to Figure 2, two graphs related to both prediction horizons of 30 and 60 minutes were also compared for each model. As shown, the Linear graphs for both prediction horizons using the four different history lengths resulted in the same average RMSE, implying that the performance of this model is similar for these history lengths. It can also be interpreted as robustness for the Linear model. Considering the VLSTM graphs, due to various RMSE among different history windows, the history length could noticeably affect the performance of this model. For both prediction horizons, the history of 90 minutes led to the least averaged RMSE for this model thus, it was chosen for the history length regarding training the model. Considering the BiLSTM graphs, moderate variation could be observed among the four different history window lengths as well. The history length of 60 minutes was the best one for this model in both prediction horizons.

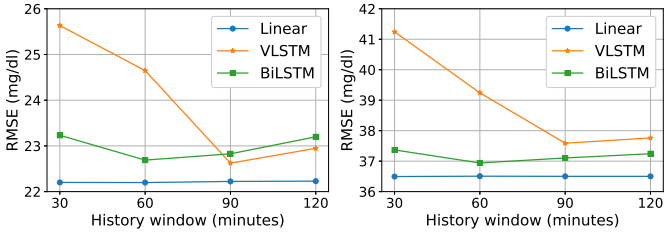


Fig. 2: Tuning the length of the history window for the prediction horizon of 30 (a) and 60 minutes (b).
Note. RMSE: Root mean square error; VLSTM: Vanilla long short-term memory; BiLSTM: Bidirectional long short-term memory.

TABLE II: Selected hyperparameters of the VLSTM and BiLSTM models.

Parameter	VLSTM		BiLSTM	
	PH: 30 min	PH: 60 min	PH: 30 min	PH: 60 min
History	90 minutes	90 minutes	60 minutes	60 minutes
Initialiser	Glorot uniform	He uniform	Glorot uniform	He uniform
Activation	ReLU	ReLU	ReLU	ReLU
Optimiser	Adam	Adam	Adam	Adam
cell type	Vanilla LSTM	Vanilla LSTM	Bidirectional LSTM	Bidirectional LSTM

Note. PH: Prediction horizon; VLSTM: Vanilla long short-term memory; BiLSTM: Bidirectional long short-term memory.

3) Ensemble models: Ensemble methods are advanced approaches for solving a range of machine learning tasks. These methods have two levels of learning. At the first level, multiple base-learner models are trained, followed by combining the predictions of base-learner models for making the final prediction at the second level. The core assumption of ensemble learning is that improvements could happen due to the compensation of the single base-learner's error by other base-learners [37].

This work looks into the second level of learning in three ways—i.e., univariate, multivariate, and two-dimensional data analysis. In the proposed methodologies, meta-learning output fusion was used at the second learning level to integrate base-learners outputs into one final prediction.

The non-ensemble models (i.e., Linear, VLSTM, and BiLSTM) were used as base-learners. The outputs of base-learners were used as the input of a meta-learner. To fuse the outputs of base-learners, stacking [17] and two novel approaches, named Multivariate and Subsequences, were investigated. The meta-learners were chosen for each approach based on the requirements of the output fusion in the second level of learning.

During reframing time series to sequence-to-sequence samples, non-equal history length of base-learners (30, 60 and 90 minutes for the Linear, BiLSTM, and VLSTM models, respectively) resulted in

generating twelve and six more samples for Linear and BiLSTM than VLSTM. The first twelve and six data points were discarded from training and testing subsets used for the BiLSTM and Linear models to equalise the sample sizes, which was an integration provision.

a) Stacking: In this model, the output sequences of three base-learners were stacked and used as the input sequence of a meta-learner. VLSTM, BiLSTM, and Linear models were considered as base-learners, and by virtue of the simplicity the Linear model was regarded as the meta-learner. Figure 3a depicts the schematic of this approach for the BGL prediction of 30 minutes in advance where $\hat{Y}_1$ is the output sequence of the Linear model consisting of six points ahead prediction values of $\hat{y}_{11}$, $\hat{y}_{12}$, $\hat{y}_{13}$, $\hat{y}_{14}$, $\hat{y}_{15}$, and $\hat{y}_{16}$. Similarly, $\hat{Y}_2 = [\hat{y}_{21}, \hat{y}_{22}, \hat{y}_{23}, \hat{y}_{24}, \hat{y}_{25}, \hat{y}_{26}]$ and $\hat{Y}_3 = [\hat{y}_{31}, \hat{y}_{32}, \hat{y}_{33}, \hat{y}_{34}, \hat{y}_{35}, \hat{y}_{36}]$ represent the output sequences of VLSTM and BiLSTM models, respectively. These three output sequences were concatenated to feed the meta-learner. The output of the meta-learner was the final prediction.

b) Multivariate approach: In this method, the outputs of the base-learners were considered as different variables. The existing univariate time series forecasting task at the first level of learning was converted to a three-variate time series forecasting task at the second level of learning. Considering the technique of meta-learning output fusion, a multivariate LSTM model was used as the meta-learner. Figure 3b illustrates a diagram of this methodology for the 30-minute prediction horizon. As shown, $\hat{Y}_1$, $\hat{Y}_2$, and $\hat{Y}_3$ (the output sequences of base-learners) were simultaneously used as a three-variable input sequence for the meta-learning process.

Due to similarities in the architectures of this model and the univariate VLSTM model and for a reduction in computational costs, the same hyperparameters tuned for the univariate model were used instead of performing a separate hyperparameter-tuning process. Hence, the model composed of an `LSTM` layer with 200 units followed by a fully-connected `Dense` layer with 100 nodes and an output layer. Both hidden layers used ReLU as the activation function. Glorot uniform and He uniform were used as the initialiser for the prediction horizons of 30 and 60 minutes, respectively. Furthermore, MSE and Adam were used as the loss function and the optimiser. The model was trained with 500 epochs with a learning rate of 0.01 and an epoch size of 32.

c) Subsequences approach: In this method, the VLSTM, BiLSTM, and Linear models were used as base-learners. In this regard, we looked into their output sequences and considered $\hat{Y}_1$, $\hat{Y}_2$, and $\hat{Y}_3$ output sequences as three subsequences for the meta-learner. In this way, our one-dimensional time series forecasting task was configured as a two-dimensional data analysis problem. To solve this two-dimensional problem, a convolutional LSTM (ConvLSTM) was applied as the meta-learner, which is shown to be suitable for two-dimensional spatial-temporal data analysis. This model comprised a convolutional NN as the encoder for reading and extracting important features from the input and a vanilla LSTM as the decoder for interpreting the output of the encoder. Several subsequences were needed for each sample in order to fit the model to our univariate time series analysis. Thus, the output sequences of base-learners were employed as these subsequences. The model was constructed of a `ConvLSTM2D` layer with 64 nodes, followed by a `flatten` layer to flatten the outputs before being interpreted. The fixed-length output was then provided using a `RepeatVector` layer, and the output sequence was fed to an `LSTM` layer with 200 nodes as the input. Next, a `Dense` layer with 100 nodes was used for interpreting time steps, along with the output layer. A `TimeDistributed` wrapper was also used to have the prediction for each time step. Further, ReLU, MSE, and Adam were used for all hidden layers as the activation function, loss function, and optimiser, respectively. The model was trained with 500 epochs with a learning rate of 0.01 and an epoch size of 32. Figure 3c displays a schematic of the developed method for the prediction horizon of 30 minutes.

C. Evaluation Criteria

In this work, the performance of the developed models was evaluated regression-wise and clinical-wise. A description of the evaluation criteria is presented in the following section.

1) Regression-wise evaluation: Two primary metrics of regression accuracy were calculated to evaluate the overall performance of the developed forecasting models, including the RMSE and mean absolute error (MAE) as Equations 1 and 2, respectively.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{1}$$

$$MAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N} \tag{2}$$

where N, y_i, and $\hat{y}_i$ are the size of the evaluation set, the reference value, and the predicted value in both equations, respectively.

2) Clinical-wise evaluation: The Matthews correlation coefficient (MCC) metric and surveillance error grid (SEG) [38] analysis were utilised to have a clinical insight regarding the performance of the developed BGL prediction models.

a) MCC: The MCC criterion was used to assess the effectiveness of the models in distinguishing between adverse glycaemic (hypoglycaemia (BGL < 70mg/dL) or hyperglycaemia (BGL > 180mg/dL)) and normoglycaemic (70mg/dL < BGL < 1800mg/dL) events [15]. Accordingly, adverse glycaemic and normoglycaemic events were considered as positive and negative classes, respectively. The predictions of the regression models were used to assign a prediction label. A confusion matrix was generated following comparing reference and predicted labels. The confusion matrix (Table III) comprised true positives (TP), the number of adverse glycaemic events correctly predicted as adverse glycaemic events; true negatives (TN), normoglycaemic events correctly predicted as normoglycaemic events; false positives (FP), the number of normoglycaemic events incorrectly predicted as adverse glycaemic events; and false negatives (FN), the number of adverse glycaemic events incorrectly predicted as normoglycaemic events. MCC was then calculated as Equation 3.

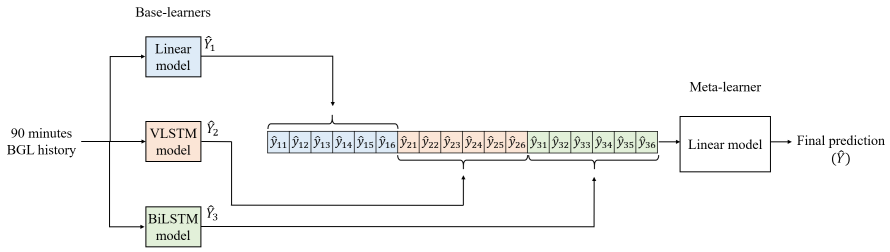
$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3}$$

b) SEG: It can analyse and visualise BGL prediction and allocates a risk value to each predicted BGL based on the comparison with the corresponding reference BGL [38]. The surveillance error (SE) criterion proposed in [21] was used based on the SEG in order to have a unique score for each patient. The SE, which is the average of a bilinear interpolation of the SEG, was considered as a metric for the clinical assessment of the performance of prediction models.

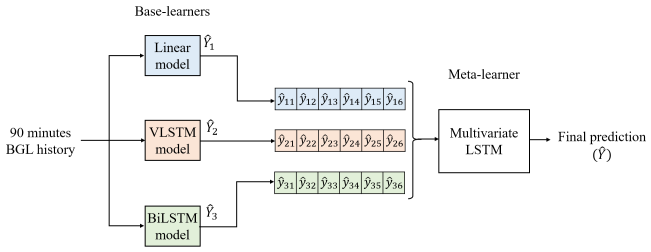
TABLE III: Confusion matrix for distinguishing between adverse and normoglycaemia events

		Reference	
		Adverse (P)	Normal (N)
Prediction	Adverse (P)	TP	FP
	Normal (N)	FN	TN

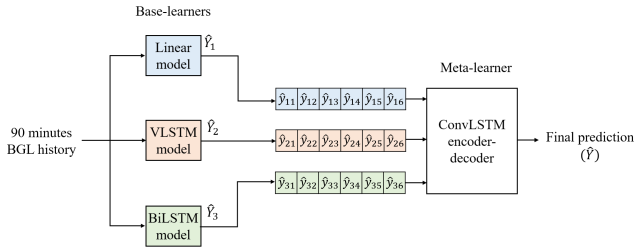
Note. TP: True positive; FN: False negative; FP: False positive; TN: True negative.



(a) The Stacking approach



(b) The Multivariate approach



(c) The Subsequences approach

Fig. 3: Diagrams of the proposed Stacking approach (a), Multivariate approach (b), and Subsequences approach (c) for the BGL prediction 30 minutes in advance by considering the Linear, VLSTM, and BiLSTM models as base-learners. In the Stacking approach, the output vectors of the base-learners were concatenated and fed as the input to the Linear meta-learner. In Multivariate approach, the output vectors of base-learners were considered as three different variables and fed to a multivariate LSTM meta-learner. In the Subsequences approach, the output vectors of base-leaners were considered as different subsequences for a two-dimensional ConvLSTM encoder-decoder meta-learner.
Note. VLSTM: Vanilla long short-term memory; BiLSTM: Bidirectional long short-term memory; BGL: Blood glucose level; ConvLSTM: convolutional long short-term memory.

D. Statistical Analysis

To statistically compare the performances of all the seven models on the 12 datasets of T1DM data contributors, the non-parametric Friedman test [39] was performed. Then, to pairwisely determine differences, post-hoc analysis utilising Wilcoxon test [40] was done. A significance threshold of 5% was considered. Also, to visualise the post-hoc results, a critical difference diagram (CDD) [41] was employed.

V. Results and Discussion

In this section, the results of all evaluation criteria consisting of RMSE, MAE, MCC, and SE are presented for baseline, non-ensemble, and ensemble models in both horizons of 30 and 60 minutes. The training and testing sets in the Ohio dataset were used for training and evaluation purposes, respectively. The extrapolated data in test sets were excluded in the calculation of evaluation metrics. In addition, due to their stochastic nature, NN models with performance depending on random initialisation were run five times. The mean and SD of results over the five runs are reported in this section.

A. Baseline Model

Table IV presents the evaluation results for the naive baseline model, which returns the last known value. The results show average evaluation criteria over the 12 patients for both prediction horizons of 30 and 60 minutes.

B. Non-ensemble Models

Table V provides the evaluation criteria of the three non-ensemble models for the BGL prediction, 30 and 60 minutes in advance. Comparing the results of Table V with those in Table IV, all developed non-ensemble models outperformed the baseline model regarding all evaluation criteria for both prediction horizons. Considering the

TABLE IV: Evaluation results of the naive baseline model for the prediction horizons of 30 and 60 minutes.

PID	PH=30 min				PH=60 min			
	RMSE	MAE	MCC	SE	RMSE	MAE	MCC	SE
540	28.42	21.15	0.668	0.312	47.62	36.19	0.473	0.509
544	22.33	16.47	0.701	0.238	37.56	28.33	0.474	0.405
552	20.62	14.75	0.726	0.234	33.51	24.44	0.550	0.379
559	23.16	16.63	0.747	0.229	39.05	28.74	0.542	0.393
563	20.75	15.44	0.698	0.226	33.95	25.52	0.492	0.363
567	27.37	19.81	0.641	0.305	45.51	33.55	0.376	0.510
570	18.97	13.85	0.828	0.138	31.84	24.26	0.707	0.239
575	25.66	17.83	0.707	0.273	39.83	28.95	0.516	0.439
584	24.64	17.77	0.724	0.244	40.99	29.69	0.540	0.393
588	21.95	16.06	0.724	0.213	35.86	26.74	0.558	0.347
591	24.41	17.96	0.635	0.300	38.37	28.97	0.399	0.481
596	21.03	15.21	0.689	0.244	35.16	25.76	0.481	0.394
Avg	23.27	16.91	0.707	0.246	38.27	28.43	0.509	0.404

Note. PID: Patient ID; PH: Prediction horizon; RMSE: Root mean square error; MAE: Mean absolute error; MCC: Matthews correlation coefficient; SE: Surveillance error.

VLSTM model, the average of evaluation metrics over all patients for the prediction horizon of 30 minutes was 19.83, 14.09, 0.748, and 0.209 for RMSE, MAE, MCC, and SE, implying an improvement of 14.78%, 16.67%, 5.79%, and 15.04% for these metrics, respectively, compared to the baseline.

Based on the comparison results of the non-ensemble models (Table V) and Figure 2, it can be seen that the performance of the Linear model was considerably better than the two LSTM models in the tuning process. However, this difference was negligible in the final evaluation process. This deviation is plausible because, in the final evaluation, a larger dataset for training was used —in tuning, 80% of the training data were used for training purposes rather than all. It is postulated that more training data can improve the performance of the deep learning models [42].

TABLE V: Evaluation results of non-ensemble models for the prediction horizons of 30 and 60 minutes.

PID	Model	PH=30 min				PH=60 min			
		RMSE	MAE	MCC	SE	RMSE	MAE	MCC	SE
540	Linear	22.08	16.60	0.740	0.244	41.10	31.81	0.530	0.438
	VLSTM	21.78 ± 0.12	16.25 ± 0.07	0.737 ± 0.00	0.241 ± 0.00	44.94 ± 2.89	33.13 ± 1.27	0.551 ± 0.02	0.440 ± 0.01
	BiLSTM	22.60 ± 0.78	16.72 ± 0.29	0.725 ± 0.01	0.247 ± 0.00	40.80 ± 0.74	31.22 ± 0.51	0.557 ± 0.01	0.425 ± 0.00
544	Linear	18.10	13.34	0.786	0.199	31.82	24.68	0.612	0.358
	VLSTM	18.09 ± 0.30	13.02 ± 0.22	0.787 ± 0.01	0.192 ± 0.00	31.59 ± 0.46	24.15 ± 0.54	0.602 ± 0.01	0.351 ± 0.01
	BiLSTM	18.35 ± 1.29	13.29 ± 1.08	0.789 ± 0.02	0.195 ± 0.02	31.30 ± 0.12	24.02 ± 0.13	0.616 ± 0.01	0.349 ± 0.00
552	Linear	16.79	12.77	0.739	0.212	30.25	23.65	0.586	0.358
	VLSTM	16.79 ± 0.09	12.61 ± 0.11	0.746 ± 0.01	0.206 ± 0.00	30.37 ± 0.47	23.34 ± 0.47	0.585 ± 0.01	0.352 ± 0.01
	BiLSTM	17.16 ± 0.16	12.78 ± 0.14	0.735 ± 0.00	0.209 ± 0.00	30.30 ± 0.13	22.98 ± 0.27	0.579 ± 0.00	0.349 ± 0.00
559	Linear	19.32	13.69	0.796	0.193	33.73	24.86	0.628	0.340
	VLSTM	19.26 ± 0.05	13.52 ± 0.04	0.790 ± 0.01	0.197 ± 0.00	35.03 ± 0.74	25.91 ± 0.50	0.624 ± 0.01	0.353 ± 0.01
	BiLSTM	20.36 ± 0.67	14.31 ± 0.56	0.781 ± 0.01	0.202 ± 0.01	34.00 ± 0.55	24.77 ± 0.31	0.626 ± 0.00	0.337 ± 0.00
563	Linear	19.25	13.16	0.763	0.183	30.47	22.08	0.559	0.304
	VLSTM	18.94 ± 0.12	13.02 ± 0.03	0.770 ± 0.01	0.179 ± 0.00	31.12 ± 0.26	22.38 ± 0.31	0.554 ± 0.01	0.305 ± 0.00
	BiLSTM	18.62 ± 0.10	13.03 ± 0.06	0.764 ± 0.01	0.179 ± 0.00	30.30 ± 0.24	22.01 ± 0.20	0.556 ± 0.02	0.298 ± 0.00
567	Linear	21.01	15.13	0.625	0.258	37.56	28.34	0.354	0.475
	VLSTM	20.70 ± 0.06	14.74 ± 0.06	0.658 ± 0.00	0.250 ± 0.00	37.39 ± 0.46	28.29 ± 0.40	0.382 ± 0.01	0.474 ± 0.01
	BiLSTM	21.48 ± 0.44	15.39 ± 0.34	0.648 ± 0.02	0.257 ± 0.01	39.01 ± 1.54	29.42 ± 1.23	0.347 ± 0.04	0.490 ± 0.02
570	Linear	16.59	11.87	0.858	0.115	28.71	21.41	0.753	0.204
	VLSTM	16.46 ± 0.13	11.43 ± 0.17	0.859 ± 0.00	0.111 ± 0.00	28.10 ± 0.41	20.04 ± 0.22	0.782 ± 0.00	0.188 ± 0.00
	BiLSTM	16.79 ± 0.64	11.71 ± 0.60	0.859 ± 0.01	0.113 ± 0.01	29.23 ± 0.55	21.49 ± 0.58	0.751 ± 0.01	0.202 ± 0.01
575	Linear	24.35	15.68	0.741	0.241	37.65	27.34	0.528	0.407
	VLSTM	24.20 ± 0.31	15.46 ± 0.09	0.729 ± 0.00	0.237 ± 0.00	37.80 ± 0.50	27.08 ± 0.33	0.501 ± 0.01	0.407 ± 0.01
	BiLSTM	24.23 ± 0.48	15.81 ± 0.43	0.721 ± 0.01	0.242 ± 0.01	37.38 ± 0.34	27.26 ± 0.59	0.504 ± 0.01	0.405 ± 0.01
584	Linear	21.96	16.10	0.765	0.223	36.64	27.58	0.602	0.371
	VLSTM	22.58 ± 0.19	16.58 ± 0.19	0.764 ± 0.00	0.228 ± 0.00	38.09 ± 1.54	28.52 ± 1.34	0.614 ± 0.02	0.377 ± 0.02
	BiLSTM	22.05 ± 0.34	16.03 ± 0.36	0.774 ± 0.00	0.220 ± 0.01	37.60 ± 0.13	28.22 ± 0.12	0.619 ± 0.00	0.371 ± 0.00
588	Linear	19.22	14.10	0.750	0.187	31.86	23.48	0.546	0.313
	VLSTM	19.47 ± 0.14	14.11 ± 0.09	0.732 ± 0.00	0.188 ± 0.00	31.87 ± 0.08	23.41 ± 0.04	0.540 ± 0.00	0.309 ± 0.00
	BiLSTM	19.16 ± 0.14	13.83 ± 0.09	0.742 ± 0.01	0.183 ± 0.00	32.08 ± 0.23	23.48 ± 0.28	0.548 ± 0.01	0.308 ± 0.00
591	Linear	21.74	15.92	0.635	0.275	34.00	26.75	0.401	0.436
	VLSTM	21.82 ± 0.15	15.65 ± 0.10	0.652 ± 0.00	0.270 ± 0.00	34.50 ± 0.61	26.67 ± 0.63	0.418 ± 0.02	0.430 ± 0.01
	BiLSTM	22.20 ± 0.59	16.12 ± 0.61	0.644 ± 0.01	0.277 ± 0.01	34.71 ± 0.33	26.78 ± 0.36	0.430 ± 0.02	0.432 ± 0.01
596	Linear	17.82	12.81	0.728	0.210	29.72	22.16	0.542	0.335
	VLSTM	17.86 ± 0.09	12.68 ± 0.13	0.752 ± 0.00	0.204 ± 0.00	29.77 ± 0.21	21.85 ± 0.16	0.585 ± 0.01	0.326 ± 0.00
	BiLSTM	17.57 ± 0.14	12.47 ± 0.13	0.752 ± 0.00	0.201 ± 0.00	29.77 ± 0.20	21.90 ± 0.14	0.567 ± 0.01	0.327 ± 0.00
Avg	Linear	19.85	14.26	0.744	0.212	33.63	25.34	0.553	0.361
	VLSTM	19.83 ± 0.05	14.09 ± 0.04	0.748 ± 0.00	0.209 ± 0.00	34.21 ± 0.15	25.40 ± 0.04	0.562 ± 0.00	0.359 ± 0.00
	BiLSTM	20.05 ± 0.14	14.29 ± 0.10	0.744 ± 0.00	0.211 ± 0.00	33.87 ± 0.12	25.29 ± 0.11	0.558 ± 0.00	0.358 ± 0.00

Note. PID: Patient ID; PH: Prediction horizon; RMSE: Root mean square error; MAE: Mean absolute error; MCC: Matthews correlation coefficient; SE: Surveillance error.

C. Ensemble Models

The evaluation results of the three developed ensemble models for both prediction horizons of 30 and 60 minutes are listed in Table VI.

It is notable to feed a unique input to meta-learners, among the five non-ensemble VLSTM and BiLSTM trained models, the model with the lowest RMSE on the 20% of the training data allocated to the validation data was selected for each base-learner. Then, the ensemble models were run five times, and the mean and SD over the five runs are presented accordingly.

According to the comparison of results in Tables IV and VI, all developed ensemble models performed better than the baseline regarding all evaluation criteria for both prediction horizons. Considering the Stacking model among ensemble models, the average values of evaluation metrics over all patients for the prediction horizon of 30 minutes were 19.63, 13.88, 0.756, and 0.204 for RMSE, MAE, MCC, and SE, indicating an improvement of 15.64%, 17.91%, 6.93%, and 17.07% for these metrics, respectively, in comparison with the baseline. This model also made an improvement of 12.59%, 13.29%, 12.96%, and 13.86% for RMSE, MAE, MCC, and SE metrics for the prediction horizon of 60 minutes, respectively.

According to the comparison between the results of Tables V and VI, ensemble models outperformed non-ensemble models for both prediction horizons. Further, it is worth mentioning that these improvements happened while due to computational costs, the meta-learners of ensemble models were not fine-tuned, but the hyperparameter optimisation was performed for non-ensemble models.

The colour-coded SEGs related to the predictions of the Stacking model 30 minutes in advance for patients 570 and 575 (with the best and the worst evaluation results, respectively) are illustrated in Figure 4 to have a clinical insight into BGL predictions. As shown in Figure 4a, BGL predictions for patient 570 are in the none and mild risk regions. However, some predictions are placed in the moderate to high risk regions for patient 575 in Figure 4b.

TABLE VI: Evaluation results of ensemble models for the prediction horizons of 30 and 60 minutes.

PID	Model	PH=30 min				PH=60 min			
		RMSE	MAE	MCC	SE	RMSE	MAE	MCC	SE
540	Stacking	21.98	16.16	0.740	0.240	40.43	30.64	0.564	0.415
	Multivariate	21.46 ± 0.07	16.11 ± 0.04	0.732 ± 0.00	0.240 ± 0.00	40.25 ± 0.13	30.81 ± 0.06	0.565 ± 0.00	0.417 ± 0.00
	Subsequences	21.54 ± 0.18	16.06 ± 0.05	0.733 ± 0.00	0.240 ± 0.00	40.25 ± 0.42	30.62 ± 0.28	0.571 ± 0.01	0.416 ± 0.00
544	Stacking	17.83	12.73	0.795	0.182	30.82	22.87	0.628	0.325
	Multivariate	17.88 ± 0.04	12.79 ± 0.04	0.793 ± 0.00	0.185 ± 0.00	30.96 ± 0.07	23.23 ± 0.14	0.620 ± 0.00	0.335 ± 0.00
	Subsequences	17.92 ± 0.10	12.80 ± 0.08	0.790 ± 0.00	0.185 ± 0.00	31.07 ± 0.19	23.16 ± 0.11	0.621 ± 0.00	0.333 ± 0.00
552	Stacking	16.42	12.13	0.758	0.199	30.24	22.54	0.600	0.344
	Multivariate	16.70 ± 0.03	12.48 ± 0.03	0.744 ± 0.00	0.205 ± 0.00	30.02 ± 0.06	22.84 ± 0.09	0.579 ± 0.00	0.348 ± 0.00
	Subsequences	16.68 ± 0.03	12.41 ± 0.04	0.744 ± 0.00	0.203 ± 0.00	29.95 ± 0.13	22.57 ± 0.21	0.579 ± 0.01	0.346 ± 0.00
559	Stacking	19.33	13.37	0.788	0.197	35.10	25.55	0.646	0.343
	Multivariate	19.45 ± 0.26	13.47 ± 0.09	0.790 ± 0.00	0.195 ± 0.00	34.91 ± 0.18	25.48 ± 0.08	0.634 ± 0.00	0.345 ± 0.00
	Subsequences	19.27 ± 0.12	13.33 ± 0.09	0.793 ± 0.00	0.194 ± 0.00	34.95 ± 0.16	25.47 ± 0.09	0.635 ± 0.00	0.345 ± 0.00
563	Stacking	18.86	12.97	0.770	0.178	30.92	22.02	0.568	0.302
	Multivariate	18.61 ± 0.05	12.91 ± 0.04	0.773 ± 0.00	0.177 ± 0.00	30.91 ± 0.32	22.12 ± 0.16	0.563 ± 0.00	0.299 ± 0.00
	Subsequences	18.56 ± 0.10	12.88 ± 0.03	0.770 ± 0.00	0.179 ± 0.00	30.69 ± 0.27	22.08 ± 0.13	0.563 ± 0.01	0.299 ± 0.00
567	Stacking	20.49	14.55	0.685	0.243	36.51	27.69	0.384	0.460
	Multivariate	20.52 ± 0.04	14.60 ± 0.06	0.666 ± 0.01	0.245 ± 0.00	37.06 ± 0.10	27.78 ± 0.07	0.384 ± 0.00	0.459 ± 0.00
	Subsequences	20.59 ± 0.07	14.67 ± 0.04	0.668 ± 0.00	0.246 ± 0.00	37.52 ± 0.65	27.93 ± 0.35	0.390 ± 0.01	0.457 ± 0.00
570	Stacking	16.39	11.24	0.869	0.108	27.63	19.93	0.780	0.188
	Multivariate	16.48 ± 0.09	11.37 ± 0.13	0.862 ± 0.00	0.111 ± 0.00	27.94 ± 0.12	20.16 ± 0.08	0.777 ± 0.00	0.190 ± 0.00
	Subsequences	16.44 ± 0.06	11.30 ± 0.06	0.862 ± 0.00	0.110 ± 0.00	28.01 ± 0.25	20.23 ± 0.24	0.775 ± 0.01	0.192 ± 0.00
575	Stacking	23.38	15.25	0.736	0.235	37.01	26.40	0.517	0.402
	Multivariate	23.86 ± 0.08	15.39 ± 0.01	0.730 ± 0.00	0.234 ± 0.00	37.40 ± 0.23	26.63 ± 0.11	0.499 ± 0.01	0.401 ± 0.00
	Subsequences	23.89 ± 0.12	15.38 ± 0.06	0.730 ± 0.00	0.235 ± 0.00	36.88 ± 0.74	25.98 ± 0.41	0.511 ± 0.01	0.394 ± 0.01
584	Stacking	22.08	16.01	0.773	0.220	36.92	27.59	0.624	0.363
	Multivariate	21.89 ± 0.12	15.85 ± 0.13	0.764 ± 0.01	0.218 ± 0.00	37.14 ± 0.26	27.40 ± 0.34	0.613 ± 0.01	0.361 ± 0.01
	Subsequences	21.97 ± 0.13	15.97 ± 0.17	0.762 ± 0.00	0.221 ± 0.00	37.15 ± 0.15	27.39 ± 0.27	0.617 ± 0.00	0.359 ± 0.01
588	Stacking	19.60	14.08	0.750	0.181	31.77	23.18	0.558	0.301
	Multivariate	19.41 ± 0.11	14.00 ± 0.11	0.740 ± 0.00	0.184 ± 0.00	31.90 ± 0.19	23.35 ± 0.15	0.549 ± 0.00	0.306 ± 0.00
	Subsequences	19.20 ± 0.10	13.85 ± 0.10	0.746 ± 0.00	0.183 ± 0.00	31.90 ± 0.07	23.39 ± 0.07	0.550 ± 0.00	0.307 ± 0.00
591	Stacking	21.50	15.64	0.644	0.270	33.87	25.65	0.444	0.416
	Multivariate	21.78 ± 0.09	15.62 ± 0.05	0.658 ± 0.00	0.269 ± 0.00	34.01 ± 0.19	26.06 ± 0.10	0.426 ± 0.00	0.421 ± 0.00
	Subsequences	21.75 ± 0.05	15.57 ± 0.05	0.649 ± 0.00	0.268 ± 0.00	34.17 ± 0.28	26.03 ± 0.15	0.447 ± 0.01	0.422 ± 0.00
596	Stacking	17.70	12.39	0.761	0.200	30.19	21.77	0.581	0.322
	Multivariate	17.70 ± 0.06	12.42 ± 0.04	0.754 ± 0.00	0.201 ± 0.00	30.37 ± 0.28	21.97 ± 0.12	0.592 ± 0.00	0.323 ± 0.00
	Subsequences	17.63 ± 0.18	12.34 ± 0.08	0.756 ± 0.00	0.200 ± 0.00	30.80 ± 0.35	22.15 ± 0.16	0.592 ± 0.00	0.325 ± 0.00
Avg	Stacking	19.63	13.88	0.756	0.204	33.45	24.65	0.575	0.348
	Multivariate	19.64 ± 0.02	13.92 ± 0.01	0.751 ± 0.00	0.205 ± 0.00	33.57 ± 0.03	24.82 ± 0.03	0.567 ± 0.00	0.350 ± 0.00
	Subsequences	19.62 ± 0.02	13.88 ± 0.01	0.750 ± 0.00	0.205 ± 0.00	33.61 ± 0.04	24.75 ± 0.06	0.571 ± 0.00	0.350 ± 0.00

Note. PID: Patient ID; PH: Prediction horizon; RMSE: Root mean square error; MAE: Mean absolute error; MCC: Matthews correlation coefficient; SE: Surveillance error.

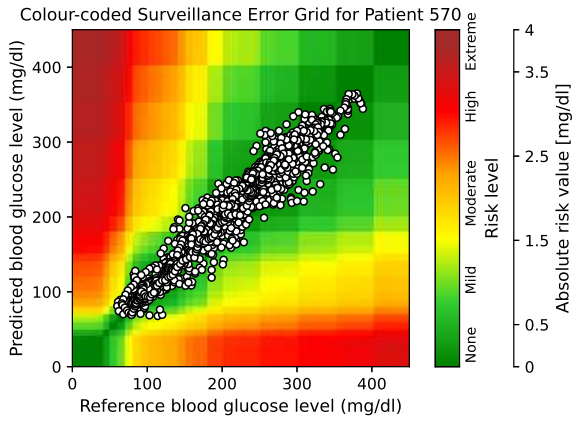
D. Statistical Analysis

The results of statistical analysis including the p-values of the Wilcoxon post-hoc test for all pairwise comparisons and the CDDs according to each evaluation metric are presented in the APPENDIX section. To have an statistical overview, Figure 5 graphically represents CDDs where a thick horizontal line connects groups of not-significantly different prediction models. The graphs are according to the average ranking over all evaluation criteria (RMSE, MAE, MCC, and SE) for both prediction horizons of 30 (5a) and 60 (5b) minutes.

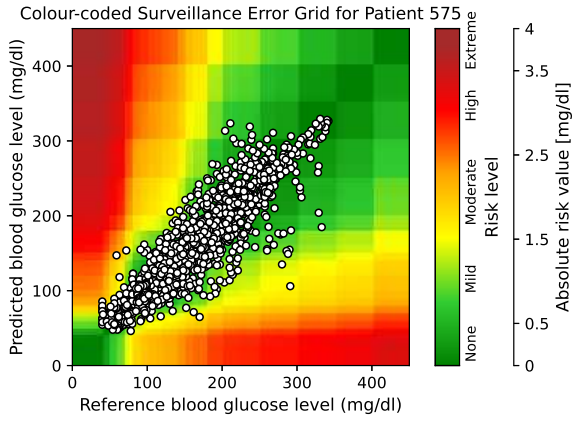
Considering the statistical analysis, it can be concluded that three non-ensemble models predicted BGL with the statistically significant improvement compared with the baseline model and no overall significant difference in between. Also, the ensemble models performed statistically significantly better than baseline and non-ensemble models with no significant intra-difference. The provided results in the APPENDIX (Tables VII, VIII, IX, and X, and Figure 6) in detail compared all the models pairwisely according to each evaluation metric.

E. Computational Analysis

The developed models rely on exploiting patterns in BGL data for the prediction. Therefore, changes in the patterns, for example, when a person's habit changes, may require a readjustment to the prediction models. Hence, it is valuable to investigate the time for retraining models relative to the time required for new data collection. The average execution time of training the developed models across all patients for running codes using a commodity laptop computer (specifications: core i7 2.8 GHz processor, 16 GB of RAM, and NVIDIA GeForce GTX 1050 Ti GPU) approximately was: a few seconds for the baseline and Linear models, 40 minutes for the VLSTM, 50 minutes for BiLSTM, 90 minutes for the Stacking, 120 minutes for the Multivariate, and 170 minutes for the Subsequences. Although the training times of developed ensemble models are considerably longer than the non-ensemble models, these training times

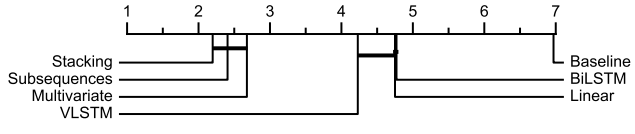


(a)

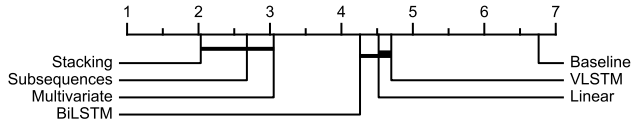


(b)

Fig. 4: The colour-coded surveillance error grid of the Stacking approach for patients 570 (a) and 575 (b). The white circles illustrate blood glucose level predictions and the corresponding reference blood glucose levels. In addition, the risk value of each prediction comparing with its reference value was coded by colour. There are five categories for a risk level, including none, mild, moderate, high, and extreme.



(a)



(b)

Fig. 5: Critical difference diagram showing comparison of all prediction models against each other over the 12 datasets of T1DM data contributors according to average over all criteria for prediction horizon of 30 (a) and 60 (b) minutes.

are considerably less than the time required for collecting new data for retraining purposes. Also, it is worth remarking that the simple Linear model produced results comparable to the two more complicated LSTM models, which are popular for time series forecasting. It could imply that even a slight improvement in the BGL prediction task would be challenging, and it could not be an easy trade-off between the complexity and accuracy of the prediction. Hence, a slight improvement of ensemble approaches could be appreciable.

VI. Conclusion

This work contributes to predicting BGL 30 and 60 minutes in advance by proposing three methodologies using deep and ensemble learning and comparing their performance with three non-ensemble benchmark models as well as a naive baseline model. The Linear, VLSTM, and BiLSTM models were the applied non-ensemble models. The benchmark models were used as base-learners for developing the ensemble models. The outputs of the base-learners were then fused using the meta-learning approach in three different ways, including univariate time series forecasting, multivariate time series forecasting, and two-dimensional data analysis. The relevant resultant ensemble models were named Stacking, Multivariate, and Subsequences, respectively.

In the Stacking approach, the output vectors of the base-learners were concatenated and fed to the Linear model as the meta-learner. In the Multivariate approach, the output vectors of base-learners were considered as different variables. Therefore, the univariate time series forecasting was converted to a multivariate time series analysis using a multivariate LSTM as the meta-learner. In the Subsequences approach, the output vectors of base-learners were considered as different subsequences. The one-dimensional time series forecasting was configured as a two-dimensional data analysis using a ConvLSTM as the meta-learner.

Overall, the results obtained show that all the developed non-ensemble models outperformed the naive baseline model. Moreover, the novel advanced ensemble models resulted in a statistically significant improvement over the non-ensemble models. Among all developed ensemble models, the Stacking approach represented slightly better performance.

In this work, using the compatibility of ensemble learning, three proposed methodologies significantly enhanced the BGL prediction accuracy. This work also offered an overview of the feasibility and usefulness of meta-learning in changing the dimension of a univariate time series forecasting task by proposing two novel Multivariate and Subsequences meta-learning approaches which provided results comparable to the Stacking approach.

This work used only CGM data for developing the BGL prediction models. For future work, it is recommended to investigate the impact of considering additional variables such as carbohydrate intake, insulin, and exercise on the performance of the BGL prediction using the proposed methodologies and comparing different variable combinations. More specifically, it would be interesting to investigate coupling appropriate data fusion techniques to the established methodology to optimally add exogenous variables to the proposed models. Also, hyperparameter-tuning was performed only for the non-ensemble models due to computational costs. Hence, it is worth optimising and fine-tuning the hyperparameters for the ensemble models as well. Examining other models as base-learners and meta-learners would also be valuable as a future investigation.

VII. Code and Data Availability

To implement the methodologies, `Python 3.6`, `TensorFlow 1.15.0` [43], and `Keras 2.2.5` [44] were employed. `Pandas` [45], `NumPy` [46], `SciPy` [47], and `Sklearn` [48] packages of

`Python` were used as well. Also, the statistical analysis was performed using `statsmodels` [49], `scikit-posthocs` [50], and cd-diagram [51]. All the implemented codes are available at the Gitlab repository. Also, the Ohio dataset used in this work can be accessed after executing a data use agreement

VIII. Appendix

Tables VII, VIII, IX, and X show the p-values of the Wilcoxon post-hoc test for comparison of all the models pairwisely for RMSE, MAE, MCC, and SE, respectively with significance threshold of 5%. Also, to quickly assimilate the results, the significant p-values are marked with bold font.

TABLE VII: p-values related to the post-hoc Wilcoxon test comparing all prediction models against each other over the 12 datasets of T1DM data contributors for RMSE.

PH	Model	Baseline	Linear	VLSTM	BiLSTM	Stacking	Multivariate	Subsequences
30 min	Baseline	1.000	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
	Linear	**<0.001**	1.000	0.424	0.151	**0.042**	0.052	**0.002**
	VLSTM	**<0.001**	0.424	1.000	0.176	**0.027**	**0.012**	**<0.001**
	BiLSTM	**<0.001**	0.151	0.176	1.000	**0.027**	**0.007**	**0.002**
	Stacking	**<0.001**	**0.042**	**0.027**	**0.027**	1.000	0.722	0.910
	Multivariate	**<0.001**	0.052	**0.012**	**0.007**	0.722	1.000	0.733
	Subsequences	**<0.001**	**0.002**	**<0.001**	**0.002**	0.910	0.733	1.000
60 min	Baseline	1.000	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
	Linear	**<0.001**	1.000	0.129	0.233	0.380	0.791	0.970
	VLSTM	**<0.001**	0.129	1.000	0.424	**0.007**	**0.021**	0.077
	BiLSTM	**<0.001**	0.233	0.424	1.000	0.204	0.339	0.233
	Stacking	**<0.001**	0.380	**0.007**	0.204	1.000	0.204	0.204
	Multivariate	**<0.001**	0.791	**0.021**	0.339	0.204	1.000	0.475
	Subsequences	**<0.001**	0.970	0.077	0.233	0.204	0.475	1.000

TABLE VIII: p-values related to the post-hoc Wilcoxon test comparing all prediction models against each other over the 12 datasets of T1DM data contributors for MAE.

PH	Model	Baseline	Linear	VLSTM	BiLSTM	Stacking	Multivariate	Subsequences
30 min	Baseline	1.000	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**< 0.001**
	Linear	**<0.001**	1.000	**0.042**	0.970	**<0.001**	**<0.001**	**< 0.001**
	VLSTM	**<0.001**	**0.042**	1.000	0.092	**<0.001**	**<0.001**	**<0.001**
	BiLSTM	**<0.001**	0.970	0.092	1.000	**0.003**	**0.002**	**<0.001**
	Stacking	**<0.001**	**<0.001**	**<0.001**	**0.003**	1.000	0.470	0.910
	Multivariate	**<0.001**	**<0.001**	**<0.001**	**0.002**	0.470	1.000	0.151
	Subsequences	**<0.001**	**<0.001**	**<0.001**	**<0.001**	0.910	0.151	1.000
60 min	Baseline	1.000	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
	Linear	**<0.001**	1.000	0.791	0.398	**0.009**	**0.009**	**0.013**
	VLSTM	**<0.001**	0.791	1.000	0.569	**<0.001**	**0.005**	**0.005**
	BiLSTM	**<0.001**	0.398	0.569	1.000	**0.012**	**0.021**	**0.021**
	Stacking	**<0.001**	**0.009**	**<0.001**	**0.012**	1.000	**0.009**	0.204
	Multivariate	**<0.001**	**0.009**	**0.005**	**0.021**	**0.009**	1.000	0.424
	Subsequences	**<0.001**	**0.013**	**0.005**	**0.021**	0.204	0.424	1.000

TABLE IX: p-values related to the post-hoc Wilcoxon test comparing all prediction models against each other over the 12 datasets of T1DM data contributors for MCC.

PH	Model	Baseline	Linear	VLSTM	BiLSTM	Stacking	Multivariate	Subsequences
30 min	Baseline	1.000	**0.006**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
	Linear	**0.006**	1.000	0.470	0.850	**0.019**	0.380	0.233
	VLSTM	**<0.001**	0.470	1.000	0.201	**0.014**	0.052	0.210
	BiLSTM	**<0.001**	0.850	0.201	1.000	**0.004**	**0.027**	**0.021**
	Stacking	**<0.001**	**0.019**	**0.014**	**0.004**	1.000	0.064	**0.020**
	Multivariate	**<0.001**	0.380	0.052	**0.027**	0.064	1.000	0.858
	Subsequences	**<0.001**	0.233	0.210	**0.021**	**0.020**	0.858	1.000
60 min	Baseline	1.000	**0.007**	**0.005**	**0.009**	**0.003**	**0.005**	**0.002**
	Linear	**0.007**	1.000	0.266	0.519	**0.001**	**0.042**	**0.021**
	VLSTM	**0.005**	0.266	1.000	0.970	**0.003**	**0.034**	**0.005**
	BiLSTM	**0.009**	0.519	0.970	1.000	**<0.001**	0.056	**0.005**
	Stacking	**0.003**	**0.001**	**0.003**	**<0.001**	1.000	**0.023**	0.233
	Multivariate	**0.005**	**0.042**	**0.034**	0.056	**0.023**	1.000	**0.028**
	Subsequences	**0.002**	**0.021**	**0.005**	**0.005**	0.233	**0.028**	1.000

Figure 6 shows CDDs related to the comparison of all prediction models against each other over the 12 datasets of T1DM data contributors according to each evaluation metric for both prediction horizons of 30 and 60 minutes.

TABLE X: p-values related to the post-hoc Wilcoxon test comparing all prediction models against each other over the 12 datasets of T1DM data contributors for SE.

PH	Model	Baseline	Linear	VLSTM	BiLSTM	Stacking	Multivariate	Subsequences
30 min	Baseline	1.000	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
	Linear	**<0.001**	1.000	**0.042**	0.233	**0.002**	**<0.001**	**<0.001**
	VLSTM	**<0.001**	**0.042**	1.000	0.212	**0.005**	**0.003**	**0.003**
	BiLSTM	**<0.001**	0.233	0.212	1.000	**0.003**	**0.004**	**0.008**
	Stacking	**<0.001**	**0.002**	**0.005**	**0.003**	1.000	0.210	0.170
	Multivariate	**<0.001**	**<0.001**	**0.003**	**0.004**	0.210	1.000	0.754
	Subsequences	**<0.001**	**<0.001**	**0.003**	**0.008**	0.170	0.754	1.000
60 min	Baseline	1.000	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
	Linear	**<0.001**	1.000	0.265	**0.050**	**0.001**	**<0.001**	**<0.001**
	VLSTM	**<0.001**	0.265	1.000	0.380	**0.003**	**<0.001**	**0.002**
	BiLSTM	**<0.001**	**0.050**	0.380	1.000	**0.009**	**0.012**	**0.009**
	Stacking	**<0.001**	**0.001**	**0.003**	**0.009**	1.000	0.110	0.470
	Multivariate	**<0.001**	**<0.001**	**<0.001**	**0.012**	0.110	1.000	0.272
	Subsequences	**<0.001**	**<0.001**	**0.002**	**0.009**	0.470	0.272	1.000

References

[1] A. D. Association, "Diagnosis and classification of diabetes mellitus," *Diabetes care*, vol. 33 (Supplement 1), pp. 62–69, 2010.

[2] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine learning and data mining methods in diabetes research," *Computational and structural biotechnology journal*, vol. 15, pp. 104–116, 2017.

[3] A. Z. Woldaregay, E. Årsand, T. Botsis, D. Albers, L. Mamykina, and G. Hartvigsen, "Data-driven blood glucose pattern classification and anomalies detection: Machine-learning applications in type 1 diabetes," *Journal of medical Internet research*, vol. 21, no. 5, e11030, 2019.

[4] G. Freckmann, J. H. Nichols, R. Hinzmann, D. C. Klonoff, Y. Ju, P. Diem, K. Makris, and R. J. Slingerland, "Standardization process of continuous glucose monitoring: Traceability and performance," *Clinica Chimica Acta*, vol. 515, pp. 5–12, 2021.

[5] C. Cobelli, E. Renard, and B. Kovatchev, "Artificial pancreas: Past, present, future," *Diabetes*, vol. 60, no. 11, pp. 2672–2682, 2011.

[6] M. Rigla, G. García-Sáez, B. Pons, and M. E. Hernando, "Artificial intelligence methodologies and their application to diabetes," *Journal of diabetes science and technology*, vol. 12, no. 2, pp. 303–310, 2018.

[7] I. Contreras and J. Vehi, "Artificial intelligence for diabetes management and decision support: Literature review," *Journal of medical Internet research*, vol. 20, no. 5, e10775, 2018.

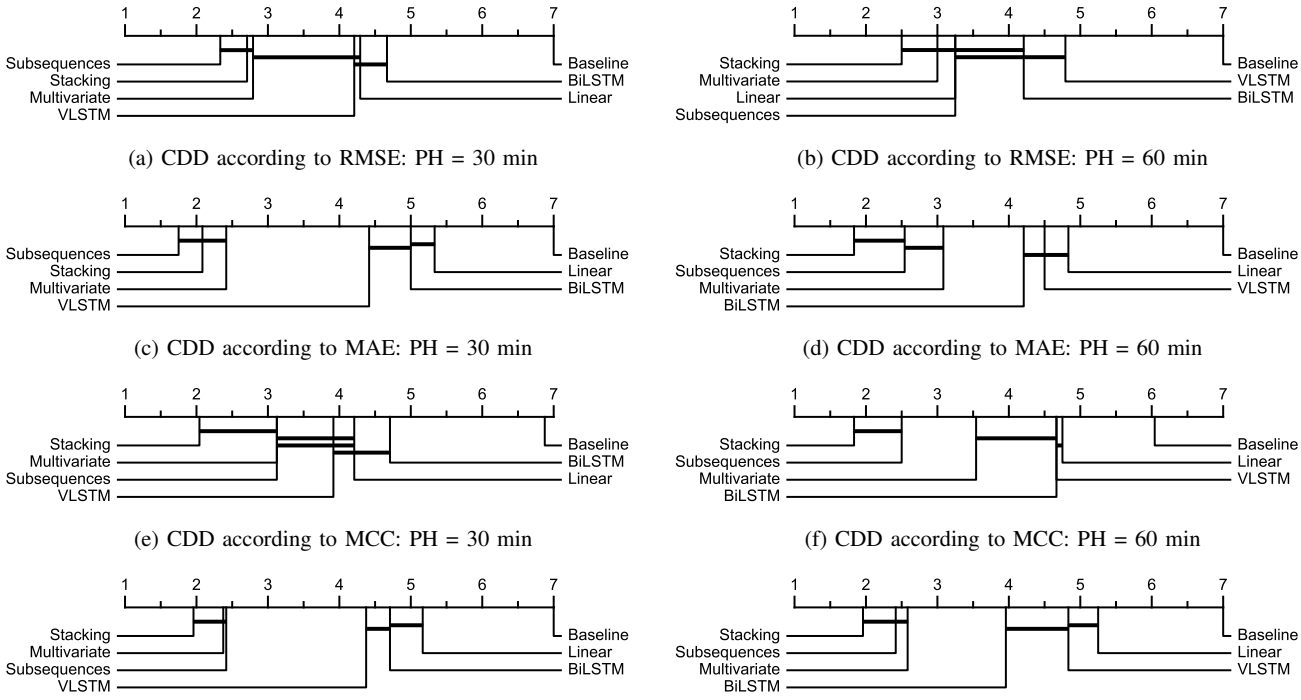
[8] D. Rodbard, "Continuous glucose monitoring: A review of successes, challenges, and opportunities," *Diabetes technology & therapeutics*, vol. 18, no. S2, S2–3, 2016.

[9] J. Xie and Q. Wang, "Benchmarking machine learning algorithms on blood glucose prediction for type 1 diabetes in comparison with classical time-series models," *IEEE Transactions on Biomedical Engineering*, 2020.

[10] K. Li, C. Liu, T. Zhu, P. Herrero, and P. Georgiou, "Glunet: A deep learning framework for accurate glucose forecasting," *IEEE journal of biomedical and health informatics*, vol. 24, no. 2, pp. 414–423, 2019.

[11] H. Nemat, H. Khadem, J. Elliott, and M. Benaissa, "Data fusion of activity and CGM for predicting blood glucose levels," in *5th International Workshop on Knowledge Discovery in Healthcare Data*, CEUR Workshop Proceedings, vol. 2675, 2020, pp. 120–124.

[12] H. Khadem, H. Nemat, J. Elliott, and M. Benaissa, "Multi-lag stacking for blood glucose level prediction," in *5th International Workshop on Knowledge Discovery in Healthcare Data*, CEUR-Workshop Proceedings, vol. 2675, 2020, pp. 146–150.



(a) CDD according to RMSE: PH = 30 min

(b) CDD according to RMSE: PH = 60 min

(c) CDD according to MAE: PH = 30 min

(d) CDD according to MAE: PH = 60 min

(e) CDD according to MCC: PH = 30 min

(f) CDD according to MCC: PH = 60 min

(g) CDD according to SE: PH = 30 min

(h) CDD according to SE: PH = 60 min

Fig. 6: Critical difference diagram showing comparison of all prediction models against each other over the 12 datasets of T1DM data contributors according to RMSE (a), (b), MAE (c), (d), MCC (e), (f), and SE (g), (h) for prediction horizon of 30 (a), (c), (e), (g) and 60 (b), (d), (f), (h) minutes.

[13] M. F. Rabby, Y. Tu, M. I. Hossen, I. Lee, A. S. Maida, and X. Hei, "Stacked lstm based deep recurrent neural network with kalman smoothing for blood glucose prediction," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 1–15, 2021.

[14] S. Mirshekarian, H. Shen, R. Bunescu, and C. Marling, "LSTMs and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2019, pp. 706–712.

[15] K. Li, J. Daniels, C. Liu, P. Herrero, and P. Georgiou, "Convolutional recurrent neural networks for glucose prediction," *IEEE journal of biomedical and health informatics*, vol. 24, no. 2, pp. 603–613, 2019.

[16] Q. Sun, M. V. Jankovic, L. Bally, and S. G. Mougiakakou, "Predicting blood glucose with an lstm and bi-lstm based deep neural network," in *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, IEEE, 2018, pp. 1–5.

[17] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[18] S. Oviedo, J. Vehí, R. Calm, and J. Armengol, "A review of personalized blood glucose prediction strategies for T1DM patients," *International journal for numerical methods in biomedical engineering*, vol. 33, no. 6, e2833, 2017.

[19] G. Sparacino, F. Zanderigo, S. Corazza, A. Maran, A. Facchinetti, and C. Cobelli, "Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series," *IEEE Transactions on biomedical engineering*, vol. 54, no. 5, pp. 931–937, 2007.

[20] S. Mirshekarian, R. Bunescu, C. Marling, and F. Schwartz, "Using LSTMs to learn physiological models of blood glucose behavior," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2017, pp. 2887–2891.

[21] J. Martinsson, A. Schliep, B. Eliasson, and O. Mogren, "Blood glucose prediction with variance estimation using recurrent neural networks," *Journal of Healthcare Informatics Research*, vol. 4, no. 1, pp. 1–18, 2020.

[22] J. Jeon, P. J. Leimbigler, G. Baruah, M. H. Li, Y. Fossat, and A. J. Whitehead, "Predicting glycaemia in type 1 diabetes patients: Experiments in feature engineering and data imputation," *Journal of Healthcare Informatics Research*, vol. 4, no. 1, pp. 71–90, 2020.

[23] C. Midroni, P. J. Leimbigler, G. Baruah, M. Kolla, A. J. Whitehead, and Y. Fossat, "Predicting glycemia in type 1 diabetes patients: Experiments with xgboost," *heart*, vol. 60, no. 90, pp. 79–84, 2018.

[24] T. Zhu, K. Li, J. Chen, P. Herrero, and P. Georgiou, "Dilated recurrent neural networks for glucose forecasting in type 1 diabetes," *Journal of Healthcare Informatics Research*, pp. 1–17, 2020.

[25] A. Güemes, G. Cappon, B. Hernandez, M. Reddy, N. Oliver, P. Georgiou, and P. Herrero, "Predicting quality of overnight glycaemic control in type 1 diabetes using binary classifiers," *IEEE journal of biomedical and health informatics*, vol. 24, no. 5, pp. 1439–1446, 2019.

[26] I. Rodríguez-Rodríguez, J.-V. Rodríguez, J.-M. Molina-García-Pardo, M.-Á. Zamora-Izquierdo, M.-T. M.-I. I. Martínez-Inglés, *et al.*, "A comparison of different models of glycemia dynamics for improved type 1 diabetes mellitus management

with advanced intelligent analysis in an internet of things context," *Applied Sciences*, vol. 10, no. 12, p. 4381, 2020.

[27] C. Marling and R. Bunescu, "The Ohio T1DM Dataset for Blood Glucose Level Prediction: Update 2020," *5th International Workshop on Knowledge Discovery in Healthcare Data*, 2020.

[28] C. Marling and R. C. Bunescu, "The OhioT1DM Dataset For Blood Glucose Level Prediction.," in *3rd International Workshop on Knowledge Discovery in Healthcare Data*, 2018, pp. 60–63.

[29] J. B. Ali, T. Hamdi, N. Fnaiech, V. Di Costanzo, F. Fnaiech, and J.-M. Ginoux, "Continuous blood glucose level prediction of type 1 diabetes based on artificial neural network," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 4, pp. 828–840, 2018.

[30] T. Hamdi, J. B. Ali, V. Di Costanzo, F. Fnaiech, E. Moreau, and J.-M. Ginoux, "Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 2, pp. 362–372, 2018.

[31] F. D'Antoni, M. Merone, V. Piemonte, P. Pozzilli, G. Iannello, and P. Soda, "Early experience in forecasting blood glucose levels using a delayed and auto-regressive jump neural network," in *2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, 2019, pp. 394–402.

[32] G. Alfian, M. Syafrudin, M. Anshari, F. Benes, F. T. D. Atmaji, I. Fahrurrozi, A. F. Hidayatullah, and J. Rhee, "Blood glucose prediction model for type 1 diabetes based on artificial neural network with time-domain features," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 4, pp. 1586–1599, 2020.

[33] H. V. Dudukcu, M. Taskiran, and T. Yildirim, "Blood glucose prediction with deep neural networks using weighted decision level fusion," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 3, pp. 1208–1223, 2021.

[34] J. Brownlee, *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.

[35] X. Yan and X. Su, *Linear regression analysis: theory and computing*. World Scientific, 2009.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.

[38] D. C. Klonoff, C. Lias, R. Vigersky, W. Clarke, J. L. Parkes, D. B. Sacks, M. S. Kirkman, B. Kovatchev, and E. G. Panel, "The surveillance error grid," *Journal of Diabetes Science and Technology*, vol. 8, no. 4, pp. 658–672, 2014.

[39] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[40] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945, ISSN: 00994987. [Online]. Available: http://www.jstor.org/stable/3001968.

[41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[42] Y. Cao, M. Raoof, S. Montgomery, J. Ottosson, and I. Näslund, "Predicting long-term health-related quality of life after bariatric surgery using a conventional neural network: A study based on the scandinavian obesity surgery registry," *Journal of clinical medicine*, vol. 8, no. 12, p. 2149, 2019.

[43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[44] F. Chollet *et al.*, *Keras*, https://github.com/keras-team/keras, 2015.

[45] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 –61. DOI: 10.25080/Majora-92bf1922-00a.

[46] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[47] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey,
bibinitperiodI. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

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

[49] S Seabold and J Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, 2010, pp. 92–96.

[50] M. Terpilowski, "Scikit-posthocs: Pairwise multiple comparison tests in python," *The Journal of Open Source Software*, vol. 4, no. 36, p. 1169, 2019. DOI: 10.21105/joss.01169.

[51] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.