DOI: 10.15439/2022R36
ISSN 2300-5963 ACSIS, Vol. 33

Development of Ensemble Tree Models for Generalized Blood Glucose Level Prediction

Aashima
Indian Institute of Technology Mandi
Kamand, Himachal Pradesh, India – 175005
aashimakodan@gmail.com

Shashank Bhargav
Indian Institute of Technology Mandi
Kamand, Himachal Pradesh, India – 175005
shashankbhargav55@gmail.com

Shruti Kaushik
RxDataScience, Inc.
USA – 277709
shrutikaushik15@gmail.com

Varun Dutt
Indian Institute of Technology Mandi
Kamand, Himachal Pradesh, India – 175005
varun@iitmandi.ac.in

Abstract*—Type-1 diabetes (T1D) patients must carefully monitor their insulin doses to avoid serious health complications. An effective regimen can be designed by predicting accurate blood glucose levels (BGLs). Several physiological and data-driven models for BGL prediction have been designed. However, less is known on the combination of different traditional machine learning (ML) algorithms for BGL prediction. Furthermore, most of the available models are patient-specific. This research aims to evaluate several traditional ML algorithms and their novel combinations for generalized BGL prediction. The data of forty T1D patients were generated using the Automated Insulin Dosage Advisor (AIDA) simulator. The twenty-four hour time-series contained samples at fifteen-minute intervals. The training data was obtained by joining eighty percent of each patient's time-series, and the remaining twenty percent time-series was joined to obtain the testing data. The models were trained using multiple patients' data so that they could make predictions for multiple patients. The traditional non-ensemble algorithms: linear regression (LR), support vector regression (SVR), k-nearest neighbors (KNN), multi-layer perceptron (MLP), decision tree (DCT), and extra tree (EXT) were evaluated for forecasting BGLs of multiple patients. A new ensemble, called the Tree-SVR model, was developed. The BGL predictions from the DCT and the EXT models were fed as features into the SVR model to obtain the final outcome. The ensemble approach used in this research was based on the stacking technique. The Tree-SVR model outperformed the non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) and other novel Tree variants (Tree-LR, Tree-MLP, and Tree-KNN). This research highlights the utility of designing ensembles using traditional ML algorithms for generalized BGL prediction.

***Index Terms*—diabetes, time-series, generalization, machine learning, stacking**

I. Introduction

Diabetes mellitus is a major global health concern as it is growing rapidly [1]. It has been reported that the adult population suffering from diabetes has tripled in the past two decades [1]. In 2019, this number was estimated to be 9.3%, which was an astounding 483 million of adults aged 20-79 years [1].

The normal blood glucose levels (BGLs) in a healthy adult after 8 hours of fasting should be between 70 mg/dl and 100 mg/dl [2]. The human pancreas maintains BGLs in this narrow range by releasing glucagon and insulin. Type-1 diabetes (T1D) is an incurable metabolic disorder characterized by high BGLs, caused by low or no insulin production by the pancreas. Complications due to T1D include cardio-vascular diseases, nerve, kidney, and eye damage. However, it has been shown that proper management can significantly reduce the complications and high costs related with diabetes [3]. T1D patients commonly use finger prick tests to measure their BGLs and adjust insulin doses multiple times in a day [3]. Diabetes management is incredibly difficult due to data inadequacy and, in some patients, due to improper data interpretation [3]. Thus, good BGL prediction models can provide great value for T1D patients.

The literature for BGL prediction contains physiological models [4][5], neural networks (NNs) [3] [6]-[21] and traditional machine learning (ML) algorithms [22]-[26]. Physiological models are pretty accurate but they require deep understanding of glucose metabolism, as the model parameters should be set only by an expert [2]. Therefore, data-driven models have an advantage that they can be used by individuals without any expert knowledge.

The major contributions of this study include development of novel combinations of traditional ML algorithms, and BGL prediction for multiple patients by using generalized models. Few research studies have combined different traditional ML algorithms using an ensemble approach. Furthermore, most of the existing research is patient-specific i.e., the models were trained to predict BGLs for a single patient at a time. The goal of this study is to evaluate several traditional ML algorithms for generalized BGL prediction of T1D patients, and combine them using an ensemble approach based on the stacking technique. Specifically, this research proposes an ensemble (Tree-SVR) of decision tree (DCT), extra tree (EXT) and support vector regression (SVR). The predictions from the DCT and the EXT models were fed as inputs to the SVR model to obtain final predictions. The Tree-SVR model is compared with traditional non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) and other novel Tree variants (Tree-LR, Tree-MLP, and Tree-KNN).

This paper is ordered as: mention the literature in BGL prediction, data description, a brief explanation of the numerous regression models used in this study and their detailed calibration process, the results from different non-ensemble and ensemble models, discussions, conclusion and the implications of this study.

II. Background

Commonly used physiological models include the Meal Simulation Model of the Glucose-Insulin System [5] and the Automated Insulin Dosage Advisor (AIDA) [4].

Hamdi et al. [6], Pappada et al. [7], Asad et al. [11], and Kushner et al [12] utilized NNs for predicting BGLs of T1D patients. They predicted BGLs in the horizon of 15 minutes, 50-80 minutes and 60-240 minutes, respectively. Recurrent neural networks (RNNs) were implemented by Sandham et al. [8][9] for providing short-term therapy to T1D patients. Martinsson et al. [10] utilized a RNN which was trained to learn parameters of an uni-variate Gaussian output distribution, for making BGL predictions up to an hour. Munoz-Organero [13] implemented a RNN based on long short-term memory (LSTM) cells on AIDA data. Rabby et al. [14] used stacked LSTM with Kalman smoothing for predicting BGLs of 6 T1D patients. Genetic algorithm (GA) was applied on the output of RNN, LSTM, bidirectional LSTM, stacked LSTM, and gated recurrent unit to achieve improved performance by Kim et al. [15].

Zhu et al. [16] introduced a dilated RNN, which was shown to outperform autoregressive models (ARX), conventional NNs and SVR model. Li et al. [17] developed a combination of incremental learning with echo state networks (ESN) and feedback ESNs, which was shown to perform better than conventional methods. The ESN belongs to the RNN family. Wang et al. [18] showed that their proposed LSTM network optimized with improved particle swarm optimization (IPSO) and variational modal decomposition (VDM) performed better than LSTM, VDM-LSTM, VDM-PSO-LSTM for 56 diabetes patients. Zhu et al. [19] used convolutional neural network (CNN) and RNN to design a generative adversarial network (GAN) for BGL prediction.

Assadi et al. [20] implemented extended Kalman filter (EKF), ARX and extreme learning machine (ELM) for predicting BGLs predictions of 20 AIDA patients. The Levenberg-Marquardt algorithm was used by Robertson et al. [3] to train an Elman RNN model for making BGL predictions. A hybrid model based on grammatical evolution (GE) and physiological models was designed by Contreras et al. [2].

Robertson et al. [3] and Munoz-Organero [13] showed that their models were trained by using only one patient's data, and they did not perform well for predicting BGLs of other patients. Thus, it is imperative to use data of multiple patients to account for inter-individual variability [3]. Monte-Moreno [21] presented a system for simultaneous noninvasive blood pressure and BGL estimation. The models, designed using photoplethysmography (PPG) and ML techniques, did not need calibration over time and patients. Pappada et al. [7] developed generalized NNs that were trained on 11-17 patients' data, and evaluated using the remaining unseen data that was not included in the NN calibration.

Georga et al. [23] evaluated SVR models on 12 real patients' data. Hamdi et al. [22] looked into differential equations (DE) and SVR for predicting BGLs of 12 real patients. Another patient-specific SVR model was designed by Plis et al. [24] and Bunescu et al. [25], where input features were generated using a standard physiological model of BGL dynamics. Monte-Moreno [21] evaluated ridge linear regression (LR), random forest (RF), multi-layer perceptron (MLP), and SVR models for BGL prediction. Mordvanyuk et al. [26] implemented patient-specific k-nearest neighbors (KNN) model for sequential T1D data, considering only carbohydrates intake, bolus dose and preprandial BGL as input features. Maged et al. [27] evaluated several ML and DL patient-specific models, and found that that EXT regressor and ANN performed the best.

Saiti et al. [28] evaluated ensemble algorithms: linear, bagging and boosting meteregressor to show that they performed better than the individual component models for BGL prediction. Ma et al. [29] combined the residual compensation network (RCN) and the autoregressive moving average (ARMA) model for predicting BGLs in the horizon of 30 minutes and 60 minutes. Xie et al. [30] compared the performance of several models: Elastic Net, gradient boosting trees, RF, SVR, vanilla LSTM, temporal convolutional network (TCN) etc. with classical ARX model for predicting BGLs of T1D patients.

Recent studies on BGL prediction are moving towards data-driven models [2]. However, to the best of our knowledge, few researches have combined different traditional ML algorithms using an ensemble approach. Most of the existing research is patient-specific i.e., the models were trained for an individual and used to obtain predictions for the same. Furthermore, very less is known on the usage of EXT algorithm for BGL prediction. In this research, the stacking ensemble approach was used with a slight variation. The predictions from DCT and EXT models were fed as input to other non-ensemble models (LR, SVR, KNN, and MLP) to obtain the final predictions. BGLs for multiple patients can be predicted by using these models.

III. Methodology

A. Data

The AIDA simulator [31] was used to generate the data used in this research. It is a freeware simulator to observe the effects of glucose-insulin interaction in T1D patients [31]. It assumes that the patient's pancreas produces no insulin [31]. The program has been designed for teaching, demonstration and self-learning purposes [31]. The AIDA model provides interactive virtual patient scenarios by using comprehensive glucose and insulin sub-models based on mathematical differential equations [31]. The user can change the input variables

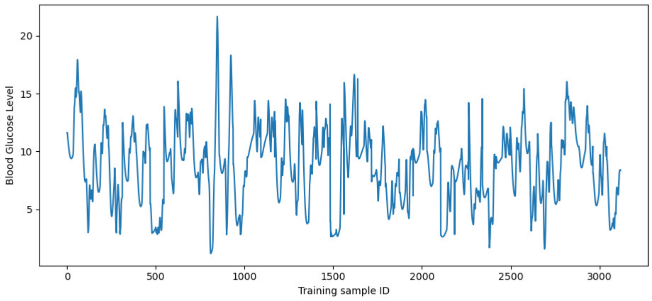

Fig. 1. Blood glucose levels of all training samples

for a patient scenario, and immediately see the impact on their BGLs [31].

The 24-hour time-series data for 40 T1D patients contained BGLs at 15-minute intervals. The BGL depended on 13 input features: patients' weight (kilograms), timestamp (hours) (0-24), carbohydrates intake (grams), short-acting injection (units) (effect in 2-5 hours), intermediate and long-acting injection (units) (effect in 24-48 hours), type of medication (units), kidney functioning renal glucose threshold (RTG) (mmol/l) (the kidneys start to excrete glucose into urine when blood glucose concentration reaches RTG), kidney function renal (ml/min) (a measure of of how well kidneys operate), liver insulin sensitivity (mmol/l), lower and upper glucose limit. The injections were taken by patients 15 minutes prior to their meal to lower BGLs.

There were 97 samples in the time-series data for each patient. The training data was obtained by stitching the initial eighty percent of each time-series. Similarly, the testing data was obtained by stitching the remaining twenty percent of each time-series. This resulted in 3120 training samples and 760 testing samples. The time-series for different patients were stitched because we aimed to develop a generalized model. Instead of personalized models, a single model should be able to learn and predict BGLs for multiple patients. Figure 1 shows the stitched training data, where the y-axis represent the BGLs, which are highly fluctuating in nature.

B. Error Metric

In this research we aimed to estimate BGL, which is a real value. The distance between the prediction values and the actual values is used to determine the quality of a regression model. Root mean squared error (RMSE) [32], which is a popular metric for regression analysis, was utilized in this study. The RMSE is obtained by taking the square root of the mean of squared differences between all the actual and predicted values. The error is defined by

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - y_i')^2}{n}} \qquad (1)$$

where n is the total number of samples, y_i is the predicted value and y_i' is the actual value. The results obtained from different models for different sets of hyperparameters were compared with each other by using RMSE.

C. Machine Learning Models

This research implemented traditional ML non-ensemble models and combined them with each other using an ensemble approach. Scikit-learn in Python [33] was used to implement all models except MLP. MLP was implemented using Keras in Python [34]. Model performance is influenced by the choice of hyperparameters [35]. Therefore, all models were trained with diverse sets of hyperparameters. The hyperparameters which obtained the lowest RMSE on the training data were chosen. The random seed for all algorithms was fixed, so that the results could be reproduced. The calibration process is explained as follows.

1) Linear Regression: It is a supervised algorithm that targets to find the best fit line, which has the least total error from all data points [36]. The error is the distance of a data point (training sample) from the line. The line in n-dimensional space is parameterized by n coefficients [36]. There are no main hyperparameters.

2) K-Nearest Neighbors: It is a supervised algorithm that stores all the training samples and estimates the outcome for a testing sample by using the target values of the K nearest neighbors [37]. The nearest neighbors among the training samples are calculated using their distance from the test sample.

The main hyperparameters are the number of nearest neighbors (K) and the distance metric [37]. The distance metric was varied as Euclidean, Manhattan, and Minkowski. The value of K was varied 5-99 (in steps of 1).

3) Support Vector Regression: It is a supervised algorithm which is based on the support vector machines (SVMs). The SVM model aims to fit a hyper-plane in the higher-dimensional feature space, such that the margin (minimum distance) of class boundaries is maximized [38]. The model is penalized by an objective function upon misclassification or if a sample lies within the margin [38]. The support vector regression (SVR) algorithm uses this principle for regression problems.

The main hyperparameters are kernel, gamma, and the degree of the regularization [39]. The kernel specifies the shape of the hyper-plane and gamma specifies the kernel coefficient [39]. The degree of regularization (C) is used to control overfitting on training data. The kernel was varied as gaussian, sigmoid, and polynomial. For gaussian kernel, C was varied 0.1-2.0 (in steps of 0.1), 10-1000 (in steps of 10), and 1000-7000 (in steps of 1000), respectively. For sigmoid kernel, C was varied 0.1-2.0 (in steps of 0.1). For polynomial kernel, degree was varied 1-10 (in steps of 1). For the three degree values that obtained the lowest training RMSE, C was varied 0.1-2.0 (in steps of 0.1), 10-1000 (in steps of 10), and 1000-7000 (in steps of 1000). The gamma parameter was set to 'scale', which is the default value in scikit-learn [39].

4) Multi-layer Perceptron: A multi-layer perceptron is a feed-forward fully-connected NN containing at least three layers: an input layer, a hidden layer, and an output layer [40].

A layer is an array of perceptrons. A perceptron is a simple computational unit that calculates output from weighted inputs by applying a non-linear activation function [41]. The weights are randomly initialized and then updated by backpropagation [40]. The aim is to minimize the gap between actual and predicted output [40].

The various hyperparameters include number of hidden layers, number of nodes in each layer, learning rate, activation function, error metric, regularization penalty, and the training algorithm [40][41]. Each training algorithm has its own additional hyperparameters. The learning rate is used to determine the step size of weight updation and regularization is used to control overfitting on the training data [41].

We evaluated three MLP models. The first layer for each model was the input layer containing 13 nodes, because each sample had 13 input features. The specifications for each layer are described as [number of nodes, activation function], and the different layers are written in the order of occurrence. The first model had 3 layers - [13,none], [32,relu] and [1,relu]. The second model had 4 layers - [13,none], [32,tanh], [64,tanh] and [1,relu]. The third model had 5 layers - [13,none], [32,tanh], [64,tanh], [32,tanh] and [1,relu]. The training algorithm (optimizer), weight initializer, and regularizer in each model were chosen to be adam [42], glorot normal [43] and L1-L2 [44], respectively. For each model, number of epochs was varied as 10, 50, 100, and 200, with and without regularization.

5) Decision Tree: It is a supervised algorithm [45]. It has a hierarchical, tree structure with decision nodes and branches [45]. An attribute is associated with each decision node, and the node splits into two or more branches [45]. A set of attribute value(s) is associated with each branch [45]. The target value is placed at the leaf node, which has no further branches [45]. During training, the data is broken into smaller subsets at each node [45]. The aim is to attain maximum homogeneity at each decision node [45]. To call a subset as completely homogeneous, it should contain instances with similar values [45]. For obtaining the output for a test sample, the algorithm travels down the tree by following the decision rules present at each node.

Maximum depth and minimum split are the main hyperparameters [46]. The maximum distance i.e., the number of branches from the root node to a leaf node of the tree, is called the maximum depth [46]. The minimum number of samples that must be available at each node to attempt a split is called the minimum split [46]. Unpruned and fully grown trees can be obtained by using the default values of these parameters [46]. The maximum depth was varied 1-50 (in steps of 1), for each minimum split value of 2, 3, and 4.

6) Extra Tree: An extremely randomized tree is similar to DCT, but while splitting the training dataset at each decision node, random splits are drawn from the training data present at that node and the candidate attributes are also selected randomly [47].

Maximum depth and minimum split are the main hyperparameters [47]. The maximum depth was varied 1-50 (in steps of 1), for each minimum split value of 2, 3, and 4.

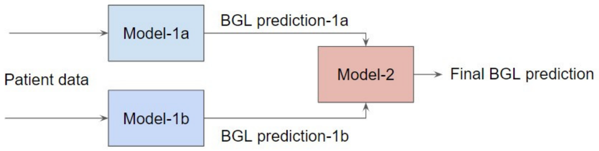

Fig. 2. The ensemble structure used in this research.

7) Ensemble: In this research, we designed ensembles inspired by the stacking approach. Stacking is an ensemble approach in which the predictions from multiple base models are aggregated using a meta-model [48]. The meta-model is usually a naive algorithm [48]. The model is trained using k-fold cross-validation [48]. The $k-1$ folds are used for training the base models, and the meta-model is trained by using the remaining 1-fold [48]. The ensemble designed in this research is slightly different: the meta-model is not a naive algorithm, and k-fold cross validation is not used for training.

Figure 2 is a visual representation of the ensemble structure. The first level (level-1) contains the base models (model-1a and model-1b), and the second level (level-2) contains the meta-model (model-2). The level-1 models were first trained on the training data. Then, the obtained predictions were used to train the level-2 model. For a test sample, the predictions obtained from level-1 models were fed into level-2 model to obtain the final outcome.

The calibrated non-ensemble models were selected to be the constituent models of an ensemble. The best two models were selected as model-1a and model-1b, and model-2 was varied as the remaining models. Since the optimal parameters were already known for model-1a and model-1b from the previous step, the intermediate level-1 predictions in the ensemble were fixed. Thus, only model-2 required calibration. The model-2 was trained to estimate the final BGL value by using the BGL estimates calculated by model-1a and model-1b as input features.

IV. Results

The traditional non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) were calibrated to obtain the optimal values for hyperparameters. In the KNN model, the optimal hyperparameters were observed to be Manhattan distance metric with K as 5. In the SVR model, the train RMSE obtained by the gaussian kernel decreased monotonically for increasing values of C. The train RMSE obtained by the sigmoid kernel increased with increasing values of C. The train RMSE obtained by the polynomial kernel was nearly the same for degrees 6, 7 and 8. For each degree value, the train RMSE was observed to decrease monotonically for increasing values of C. The optimal hyperparameters were observed to be the gaussian kernel with C as 7000. In the DCT model, a minimum split of 2 achieved the best training results. For maximum depth greater than 20, the train RMSE was observed to be constant. Therefore, 20 was chosen as the optimal maximum depth. In the EXT model, a minimum split of 2 achieved the best training results. For maximum depth greater than 25, the train RMSE was observed to be constant. Therefore, 25 was chosen as the optimal maximum depth. The three MLP models obtained the best results at 200 epochs with regularization. The second MLP model with 4 layers obtained the lowest train RMSE among all the evaluated MLP models.

Table 1 presents the performance of traditional non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) and the optimal hyperparameters obtained by calibration. Table 2 presents the performance of novel Tree variants (Tree-LR, Tree-MLP, and Tree-KNN, and Tree-SVR). The results are presented in an increasing order of test RMSE. Figure 5 presents the plot of the actual BGLs and the predicted BGLs by the Tree-SVR ensemble model for all the testing samples.

TABLE I
PERFORMANCE AND OPTIMAL HYPERPARAMETERS FOR NON-ENSEMBLE MACHINE LEARNING MODELS

Model	Optimal hyperparameters	Train RMSE	Test RMSE
DCT	maximum depth = 20, minimum split = 2	0.202	2.207
EXT	maximum depth = 25, minimum split = 2	0.199	2.207
KNN	K = 5, distance metric = Manhattan	0.667	2.374
MLP	4 layers = [13,none], [32,tanh], [64,tanh], [1,relu]	2.050	2.507
SVR	kernel = gaussian, gamma = scale, C = 7000	2.160	2.642
LR	None	2.908	2.777

TABLE II
PERFORMANCE OF DESIGNED ENSEMBLE MODELS

Model-1a	Model-1b	Model-2	Train RMSE	Test RMSE
DCT	EXT	SVR	0.202	2.201
DCT	EXT	KNN	0.201	2.206
DCT	EXT	LR	0.199	2.207
DCT	EXT	MLP	0.224	3.139

V. DISCUSSIONS

The DCT model and the EXT model outperformed other models (LR, SVR, KNN, and MLP) by obtaining equal test RMSEs of 2.207. Because of the considerable gap in train and test RMSE, the DCT, EXT, and KNN models can be considered to have overfitting. The SVR and MLP models can be said to have little or no overfitting. On the other hand, the LR model can be considered to have underfitting as it had higher train RMSE as compared to test RMSE. It performed poorly with highest train and test RMSEs of 2.908 and 2.777, respectively. This was because the LR algorithm finds a linear relationship, whereas the BGLs are highly fluctuating (Fig. 1).

The DCT and the EXT models performed the best among the non-ensemble models, therefore, they were chosen as level-1 models for the ensembles. The level-2 model was varied as LR, SVR, KNN and MLP. These novel ensembles were named the Tree variants (Tree-LR, Tree-KNN, Tree-MLP, and Tree-SVR). Because of the considerable gap in train and test RMSE, all the Tree variants can be considered to have overfitting. The Tree-KNN and the Tree-SVR models performed slightly better than the DCT and the EXT models. They obtained test RMSEs of 2.206 and 2.201, respectively. The Tree-SVR model performed the best with a test RMSE of 2.201. The improvement in results from the non-ensemble models to the novel ensemble is slight. However, it shows a promise that combination of traditional ML algorithms with each other can obtain better BGL predictions, thus provide value to T1D patients.

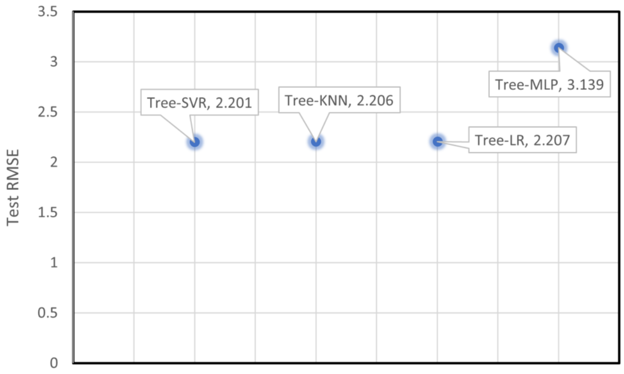

Fig. 3. Testing performance of designed ensemble models

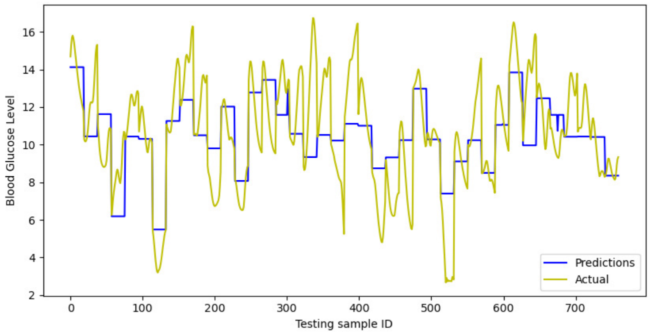

Fig. 4. Predicted and actual blood glucose levels for Tree-SVR ensemble model

The models were evaluated and tested on less amount of data, which is one of the limitations of this study. There were only 3120 training samples and 760 testing samples. The smaller number of samples did not allow validation data to be obtained. Therefore, hyperparameter tuning was performed by using only training data, which is not an ideal choice. The AIDA simulator is an educational program, thus, the use of virtual patients' data is another limitation of this research. However, the BGL values were oscillating. Therefore, the proposed Tree model can be called scalable, and it will be able to model similar data for real patients.

VI. Conclusions

Data-driven models for BGL prediction aim to assist in designing an effective regimen, thus, preventing serious health complications associated with diabetes. This study evaluated traditional non-ensemble models: linear regression (LR), multi-layer perceptron (MLP), support vector regression (SVR), k-nearest neighbors (KNN), decision tree (DCT), and extra tree (EXT) for generalized BGL prediction of type-1 diabetes (T1D) patients. Novel ensembles were designed, inspired by the stacking approach, where the predictions from the DCT and the EXT models were fed to a non-ensemble model for final BGL prediction. The AIDA simulator was used to generate 24-hour data of 40 virtual patients [31]. Eighty percent of each time-series was stitched together for training generalized models. The root mean squared error (RMSE) was used to gauge model performance.

The DCT and the EXT models outperformed the other non-ensemble models (LR, SVR, KNN, and MLP). The ensemble (Tree-SVR) of DCT, EXT and SVR outperformed all the non-ensemble models and the Tree variants (Tree-LR, Tree-KNN, and Tree-MLP) evaluated in this research.

This research has various implications for T1D patients. The patients' inability to interpret data often compromises diabetes management [3]. The models proposed in this research can help individuals in predicting future BGLs without expert knowledge about the model and glucose metabolism. Neural networks (NNs) are widely used in research studies for diabetes and BGL prediction. However, NNs often require large datasets to give promising results, and deep learning can be expensive [49]. The state-of-the-art NNs are often trained on data containing thousands, or even millions of observations [49]. This study has used only traditional ML algorithms, which can work well even with datasets containing smaller number of samples[49]. Few studies have combined different traditional ML algorithms with each other for BGL prediction; but this study has developed an ensemble approach with these algorithms. Majority of the available studies have been patient-specific, and the developed models calculated predictions on an individual basis. However, this study has developed generalized models that were shown to predict BGLs of multiple patients.

Moving forward, this research can be conducted by using lesser number of input features. The models can be evaluated on larger T1D datasets with real patients. Motivated by the improvement shown by ensemble models, more traditional ML algorithms can be combined with each other using several ensemble approaches for BGL prediction.

Acknowledgment

This research was supported by a research grant from RxDataScience, Inc., USA, to the Indian Institute of Technology Mandi, India. The project was supported by grants (award: IITM/CONS/RxDSI/VD/33) to Varun Dutt.

References

[1] IDF Diabetes Atlas, 9th edition (2019). http://www/diabetesatlas.org

[2] Iván Contreras, Silvia Oviedo, Martina Vettoretti, Roberto Visentin, and Josep Vehí. 2017. Personalized blood glucose prediction: A hybrid approach using grammatical evolution and physiological models. PLOS ONE 12 (11 2017), e0187754. https://doi.org/10.1371/journal.pone.0187754

[3] Gavin Robertson, Eldon Lehmann, William Sandham, and David Hamilton. 2011. Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-Concept Pilot Study. J. Electrical and Computer Engineering 2011 (05 2011). https://doi.org/10.1155/2011/681786

[4] E. D. Lehmann and T. Deutsch. 1992. A physiological model of glucose-insulin interaction in type 1 diabetes mellitus. Journal of biomedical engineering 14 3 (1992), 235–42.

[5] S. M. Lynch and B. W. Bequette. 2001. Estimation-based model predictive control of blood glucose in type I diabetics: a simulation study. Proceedings of the IEEE 27th Annual Northeast Bioengineering Conference (Cat. No.01CH37201) (2001),79–80.

[6] T. Hamdi, J. Ben Ali, N. Fnaiech, V. Di Costanzo, F. Fnaiech, E. Moreau, and J.Ginoux. 2017. Artificial neural network for blood glucose level prediction. In 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C). 91–95. https://doi.org/10.1109/SM2C.2017.8071825

[7] Scott Pappada, Brent Cameron, and Paul Rosman. 2008. Development of a neural network for prediction of glucose in type I diabetes patients. Journal of diabetes science and technology 2 (09 2008), 792–801. https://doi.org/10.1177/193229680800200507

[8] Sandham WA, Nikoletou D, Hamilton DJ, Patterson K, Japp A, MacGregor C. Blood glucose prediction for diabetes therapy using a recurrent artificial neural network. In: Proceedings, EUSIPCO-98, IX European Signal Processing Conference, Rhodes Is-land, Greece, 1998; Vol. 11: pp. 673-676.

[9] W. A. Sandham, D. J. Hamilton, A. Japp and K. Patterson, "Neural network and neuro-fuzzy systems for improving diabetes therapy," Proceedings of the 20th Annual Inter-national Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), Hong Kong, China, 1998, pp. 1438-1441 vol.3, doi: 10.1109/IEMBS.1998.747154.

[10] John Martinsson, Alexander Schliep, Björn Eliasson, and Olof Mogren. 2020. Blood Glucose Prediction with Variance Estimation Using Recurrent Neural Networks. Journal of Healthcare Informatics Research 4 (03 2020). https://doi.org/10.1007/s41666-019-00059-y

[11] Muhammad Asad, Usman Qamar, Babar Zeb, Aimal Khan, and Younas Khan. 2019. Blood Glucose Level Prediction with Minimal Inputs Using Feedforward Neural Network for Diabetic Type 1 Patients (ICMLC '19). Association for Computing Machinery, New York, NY, USA, 182–185. https://doi.org/10.1145/3318299.3318354

[12] Taisa Kushner, Marc D. Breton, and Sriram Sankaranarayanan.Multi-Hour Blood Glucose Prediction in Type 1 Diabetes: A Patient-Specific Approach Using Shallow Neural Network Models.Diabetes Technology & Therapeutics.Dec 2020.883-891.

[13] Mario Munoz-Organero. 2020. Deep Physiological Model for Blood Glucose Prediction in T1DM Patients. Sensors 20 (07 2020), 3896. https://doi.org/10.3390/s20143896

[14] Rabby, M.F., Tu, Y., Hossen, M.I. et al. Stacked LSTM based deep recurrent neural network with kalman smoothing for blood glucose prediction. BMC Med Inform Decis Mak 21, 101 (2021). https://doi.org/10.1186/s12911-021-01462-5

[15] Kim, Dae-Yeon &Choi, Dong-Sik & Kang, Ah & Woo, Jiyoung & Han, Yechan & Chun, Sung Wan & Kim, Jaeyun. (2022). Intelligent Ensemble Deep Learning System for Blood Glucose Prediction Using Genetic Algorithms. Complexity. 2022. 1-10. 10.1155/2022/7902418.

[16] Zhu, T., Li, K., Chen, J. et al. Dilated Recurrent Neural Networks for Glucose Forecasting in Type 1 Diabetes. J Healthc Inform Res 4, 308–324 (2020). https://doi.org/10.1007/s41666-020-00068-2

[17] Ning Li, Jianyong Tuo, Youqing Wang, Menghui Wang, Prediction of blood glucose concentration for type 1 diabetes based on echo state networks embedded with incremental learning, Neurocomputing, Volume 378, 2020, Pages 248-259, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.10.003

[18] W. Wang, M. Tong and M. Yu, "Blood Glucose Prediction With VMD and LSTM Optimized by Improved Particle Swarm Optimization," in IEEE Access, vol. 8, pp. 217908-217916, 2020, doi: 10.1109/ACCESS.2020.3041355.

[19] Zhu, T; Yao, X; Li, K; Herrero, P; Georgiou, P; (2020) Blood glucose prediction for type 1 diabetes using generative adversarial networks. In: Bach, K and Bunescu, R and Marling, C and Wiratunga, N, (eds.) Proceedings of the 5th International Workshop on Knowledge Discovery in Healthcare Data co-located with 24th European Conference on Artificial Intelligence (ECAI 2020). (pp. pp. 90-94). : Santiago de Compostela, Spain

[20] Khaoula Assadi, Takoua Hamdi, F. Fnaiech, J. M. Ginoux, and E. Moreau. 2017. Estimation of blood glucose levels techniques. 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C) (2017), 75–80.

[21] Enric Monte-Moreno. 2011. Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques.Artificial intelligence in medicine 53 (06 2011), 127–38. https://doi.org/10.1016/j.artmed.2011.05.001

[22] Takoua Hamdi, Jaouher Ben Ali, Véronique Di Costanzo, Farhat Fnaiech, Eric Moreau, and Jean-Marc Ginoux. 2018. Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm. Biocybernetics and Biomedical Engineering 38, 2 (2018), 362 – 372. https://doi.org/10.1016/j.bbe.2018.02.005

[23] Eleni Georga, Vasilios Protopappas, Diego Ardigò, Michela Marina, Ivana Zavaroni, Demosthenes Polyzos, and Dimitrios Fotiadis. 2012. Multivariate Prediction of Subcutaneous Glucose Concentration in Type 1 Diabetes Patients Based on Support Vector Regression. IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society 17 (09 2012). https://doi.org/10.1109/TITB.2012.2219876

[24] K. Plis, Razvan C. Bunescu, C. Marling, J. Shubrook, and F. Schwartz. 2014. A Machine Learning Approach to Predicting Blood Glucose Levels for Diabetes Management. In AAAI Workshop: Modern Artificial Intelligence for Health Analytics.

[25] R. Bunescu, N. Struble, C. Marling, J. Shubrook, and F. Schwartz. 2013. Blood Glucose Level Prediction Using Physiological Models and Support Vector Regression. In 2013 12th International Conference on Machine Learning and Applications, Vol. 1. 135–140. https://doi.org/10.1109/ICMLA.2013.30

[26] Natalia Mordvanyuk, F. Torrent-Fontbona, and B. López. 2017. Prediction of Glucose Level Conditions from Sequential Data. In CCIA. .

[27] Maged, Youssef & Atia, Ayman. (2022). The Prediction Of Blood Glucose Level By Using The ECG Sensor of Smartwatches. 406-411. 10.1109/MIUCC55081.2022.9781730.

[28] Kyriaki Saiti, Martin Macaš, Lenka Lhotská, Kateřina Štechová, Pavlína Pithová, Ensemble methods in combination with compartment models for blood glucose level prediction in type 1 diabetes mellitus,Computer Methods and Programs in Biomedicine, Volume 196, 2020, 105628, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2020.105628

[29] Ma, Ning et al. "Online Blood Glucose Prediction Using Autoregressive Moving Average Model with Residual Compensation Network." KDH@ECAI (2020)

[30] Xie, Jinyu & Wang, Qian. (2020). Benchmarking Machine Learning Algorithms on Blood Glucose Prediction for Type I Diabetes in Comparison With Classical Time-Series Models. IEEE Transactions on Biomedical Engineering. PP. 10.1109/TBME.2020.2975959.

[31] AIDA, http://www.2aida.org/

[32] James Moody. What does RMSE really mean?,

[33] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. 2012. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12 (01 2012).

[34] Keras webpage https://keras.io/guides/sequential_model/

[35] Wikipedia contributors. 2020. Hyperparameter (machine learning) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Hyperparameter_(machine_learning)&oldid=984957886

[36] Scikit-learn webpage https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[37] Scikit-learn webpage https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

[38] Towards Data Science webpage https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms

[39] Scikit-learn webpage https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[40] Wikipedia webpage https://en.wikipedia.org/wiki/Multilayer_perceptron

[41] Machine learning mastery webpage https://machinelearningmastery.com/neural-networks-crash-course/

[42] Keras webpage https://keras.io/api/optimizers/adam/

[43] Keras webpage https://keras.io/api/layers/initializers/

[44] Keras webpage https://keras.io/api/layers/regularizers/

[45] Dr. Saed Sayad. Decision Tree - Regression, https://www.saedsayad.com/decision_tree_reg.htm

[46] scikit-learn developers. sklearn.tree.DecisionTreeRegressor, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[47] scikit-learn developers. sklearn.tree.ExtraTreeRegressor, https://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeRegressor.html

[48] Medium webpage https://medium.com/@supun.setunga/stacking-in-machine-learning-357db1cfc3a

[49] George Seif. 2018. https://towardsdatascience.com/three-reasons-that-you-should-not-use-deep-learning-15bec517b622