

Comparison of Forecasting Algorithms for Type 1 Diabetic Glucose Prediction on 30 and 60-Minute Prediction Horizons

Richard McShinsky¹ and Brandon Marshall²

Abstract. Control of blood glucose (BG) levels is essential for diabetes management, especially for long term health improvement. Predicting both hypoglycemic events ($BG < 70$ mg/dl) and hyperglycemic events ($BG > 180$ mg/dl) is essential in helping diabetics control their long term health. In this paper we attempt to forecast future blood glucose levels, as well as analyze the efficiency of detecting both hypoglycemic events and hyperglycemic events. We do so by comparing Auto-Regressive Integrated Moving-Average, Vector Auto-Regression, Kalman Filter, Unscented Kalman Filter, Ordinary Least Squares, Support Vector Machines, Random Forests, Gradient Boosted Trees, XGBoosted Trees, Adaptive Neuro-Fuzzy Inference System (ANFIS), and Multi-Layer Perceptron in terms of Root Mean Squared Error, Mean Absolute Error, Coefficient of Determination, Matthews Correlation Coefficient, and Clarke Error Grid to compare their effectiveness in predicting future blood glucose levels, as well as predicting both hypoglycemic and hyperglycemic events.

1 Introduction

Blood glucose prediction has been an ongoing challenge within the medical field due to the near unpredictable variability of the many underlying factors influencing an individual's glucose levels. There has been a strong drive recently to create an artificial pancreas using artificial intelligence, which has necessitated the need to predict future blood glucose levels as well as the ability to accurately predict the onset of both hypoglycemic ($BG < 70$ mg/dl) and hyperglycemic ($BG > 180$ mg/dl) events [11].

Most predictive models for blood glucose encompass a physiological profile that includes a person's insulin, meal absorption, and past blood glucose levels [13]. Various machine learning methods that have been attempted to predict future blood glucose levels with regards to this profile include Auto-Regressive Integrated Moving-Average (ARIMA, see [3], [4], [13], and [15]), Support Vector Machines and Kernel Regression (SVM, see [3], [12], [13], and [15]), Random Forests (RF, see [8], [12], [13], and [15]), Gradient Boosted Trees (see [8] and [15]), and Artificial Neural Networks (see REFERENCES).

Comparing papers on the results, accuracy, and effectiveness of the models is near impossible due to different data sets being used between them. This paper seeks to offer a comparison of as many models as possible on a single data set.

In this paper, we compare the effectiveness of several models, namely ARIMA, Vector Auto-Regression Moving-Average with

Exogenous Regressor (VAR), Ordinary Least Squares (OLS), K-Nearest Neighbors (KNN), SVM, RF, Gradient Boosting, XGBoosting, Adaptive Neuro-Fuzzy Inference System (ANFIS), and Multi-Layer Perceptron. Additionally we attempt to use both the Kalman Filter and the Unscented Kalman Filter (UKF) to predict future blood glucose values. The Unscented Kalman Filter was chosen over the Extended Kalman Filter due to its ability to use state-space models to predict nonlinear functions. In comparing each of these model's effectiveness we use RMSE, MAE, the Matthew Correlation Coefficient (A commonly used metric for checking hypoglycemic and hyperglycemic events that roughly measures the quality of binary classifications) [4], and the Clarke Error Grid.

2 Data

2.1 OHIO T1DM

The data used for this comparison was the OhioT1DM data set, which was obtained as part of the second Blood Glucose Level Prediction Challenge [5]. This data set contains eight weeks worth of data for 12 people with type 1 diabetes. All contributors were on insulin pump therapy with continuous blood glucose monitoring (CGM). All pumps were of one of two brands, all life event data was reported via a custom smartphone app, and all psychological data was provided from a fitness band. The features themselves provided in the data set are: Date, Glucose Level, Finger Stick, Basal (Insulin), Basal Temperature, Bolus (Insulin), Meal (Carbohydrate Estimate), Sleep, Work, Stressors, Hypoglycemic Event, Illness, Exercise, Basis Heart Rate, Basis GSR, Basis Skin Temperature, Basis Air Temperature, Basis Steps, Basis Sleep, and Acceleration [5].

The train and test splits were given as part of the second Blood Glucose Level Prediction Challenge (see [5] for more details).

2.2 Preprocessing

The glucose readings are in about 5-minute increments while other readings are every minute. Other readings reported by the patient are at arbitrary times not aligned with the glucose readings. To combine them into one data frame to use for predicting glucose, the most important predictor, glucose levels, was made the main index. All other values were merged to the closest glucose values within the previous 4 minutes. For values that were not in this tolerance they were dropped from the data frame.

Most of these values that were dropped were due to missing data. There are many gaps where the meter was not recording glucose values. This could be times between taking it off and putting it on, the

¹ Brigham Young University, USA, email: richard.mcshinsky@byu.net

² Brigham Young University, USA, email: brandon.marshall@byu.net

hour or more it takes for the meter to get set up, or a day where the user just did not put it on. Leaving these gaps often resulted in large jumps in the training and testing data. These discontinuities would be a problem in training the models. To fill them we couldn't use interpolation methods as we are unable to know the future while predicting these values. Therefore, our method to extrapolate values for these times was to use a moving average. For example, for the first extrapolated missing value, we would use the mean of the previous 2 values. For the second we would use the mean of the previous 4 values. For the tenth we would use the mean of the previous 20 values, including the ten we had just extrapolated before that. This would happen in five minute increments until we reach the next actual value in the data frame. The last predicted value would be dropped and the data frame would continue as normal until a difference of more than 6 minutes between values was detected and this rolling average would extrapolate the missing values. The rolling average would eventually converge to the average value of all the data, but maintains the nature of the recent data. For example, if the person has had high blood glucose levels for the day, the filled data would stay high, but eventually move towards the mean of the person when using several days for large gaps. This was done since after a few hours, guessing where the person's data was going to start is nearly random guessing. Since the actual glucose values are essentially normally distributed, it is better to guess more towards the mean of the glucose levels. Meanwhile, the discontinuities were reduced by maintaining the local rolling mean. This resulted in many of the extrapolations ending very close to where the data continues from the discontinuity for this data.

3 Methods

We intend to compare many methods used for classical and regressive time series analysis. Thus, even though some methods are known to not perform well with blood glucose levels for this type of problem, they give a baseline to compare each successive method. In addition to the classical models, we used some models described in other papers about predicting glucose levels for comparison and potentially better parameter choices. Further, we chose some methods like VAR and ANFIS in order to compare methods not seen in the research found. The following subsections explain choices in why specific methods, parameters, and architecture were chosen.

3.1 Classical Methods

3.1.1 ARIMA

Even though ARIMA itself is a linear combination of a trend component, a seasonal component, and a residual component, we chose to use this model due to its classical use within time series analysis. Additionally, ARIMA was chosen due to its ability to allow us to choose the order of p and q for both the AR and MA parts of the model. These hyperparameters p and q were chosen using `stats.models.ordersselect`, from which we found that $p=2$ and $q=2$ gave the lowest error. It should be noted that the data is nearly stationary to start, so a lag of 0 was used (as larger lags resulted in a worse error). The only data features used were the previous p blood glucose levels and the q corresponding error terms.

3.1.2 VAR

VAR is a vectored version of an AR model. This allows for more types of inputs to influence the prediction, rather than just simply

using the previous p blood glucose values. VAR used the same parameters used in the ARIMA model described above.

3.1.3 Unscented Kalman Filter (UKF)

Whilst the Extended Kalman Filter (EKF) works well for linear projections, blood glucose levels are nonlinear in nature. Generally EKF can be thought of as the extension of a Gaussian Random Variable (GRV) through a linear system [14]. In the nonlinear case however, the EKF produces approximations to the values x_k , y_k , and K_k (the state, observation, and covariance for the system) [14]. In other words, the Extended Kalman Filter propagates a GRV through a first-order linearization of the nonlinear system [14].

The Unscented Kalman Filter also uses a Gaussian Random Variable, but instead uses a minimal set of carefully chosen sample points for which to propagate this GRV [14]. This is done by applying the unscented transformation to the selected sample points and then propagating these carefully chosen points through the system. Doing so allows for approximations that are accurate to the third order of a Taylor series expansion [14].

To summarize, the Unscented Kalman Filter selects carefully chosen points, applies the unscented transformation to these points, then performs the time update and measurement update as is standard in the Kalman Filter [14].

3.2 Regression and Ensemble Methods

Since the OhioT1DM data set is time series based, regular regression methods are not immediately available for us to use when forecasting data. However, we can transform the data into a regression problem by first redefining how the data is presented. Instead of each row in the data representing a single time step of the nineteen features, we instead redefine the data on the last six rows of data (we used the last 30 minutes of known information of data). Thus each row in the new reformatted data set now contains the last six known time steps with the labels being the future blood glucose values we wish to predict at each time step. Each label is the next six or twelve blood glucose values following the current time step in the OhioT1DM data set for the 30-minute and 60-minute prediction horizons respectively. In summary, each time step is reformatted to have a 6x19 feature space with each label having 6 or 12 values. With the data reformatted the following algorithms can be run.

3.2.1 Ordinary Least Squares

While the data is nonlinear in nature, it is possible that within a sufficiently small subset of the data (that is, for a sufficiently small time interval), the data may be quasi-linear. As with ODEs (where one can essentially linearize a nonlinear system) we seek to do something similar by attempting to fit affine functions to a sufficiently small time domain. Ordinary Least Squares (OLS) seeks to do this, fit an affine function (with a constant and error term), to the data set. In addition to regular OLS, we also run OLS with regularization terms, namely Lasso (L1 regularization), Ridge (L2 regularization), and Elastic Net (L1 and L2 regularization) all with α values of 1 for the regularization terms. We note that Lasso regularization gives us the advantage of feature reduction, allowing us to analyze which lags are most important in determining future blood glucose levels.

3.2.2 Support Vector Machines

We believe Support Vector Machine regression may be a useful method due to its ability to alter the kernel being used, thus allowing us to alter our definition of distance with regards to the data. Support Vector Machine (SVM) regression seeks to fit a hyperplane to the data with an ϵ -margin. Points that fall within this ϵ -margin are known as support vectors and are used to help define the hyperplane used in the regression. Notions of distance to this hyperplane are defined using a kernel. We attempt to use an RBF-kernel (with a scaling γ value) and a Polynomial Kernel (with a scaling γ value, a constant term of 0 and a power of 3) in our regressions. Each SVM had an ϵ -margin of 0.1. The results for each of the SVMs are reported under RBF, Poly, and Sig respectively.

3.2.3 K-Nearest Neighbors

It is likely that previous patterns in the lags of blood glucose (and other features) may be similar to the current pattern in the lags of features, we believe KNN regression may also be a useful regression method. KNN uses a voting method to form the regression. Using a defined metric of distance, KNN regression finds the K closest neighbors to the given data point and then returns the average of the labels. We use five neighbors, along with Euclidean distance for this algorithm. The results for this algorithm are reported under KNN.

3.2.4 Random Forest Regression

Random Forest Regression is an ensemble method that combines weak decision-tree regressors to form a strong group regressor, Random Forests allow us to create a regressor that branches based on the features. This is included here due to its use in other papers attempting blood glucose prediction (see [8], [12], [13], and [15]). To limit run-time to a reasonable length, a max-depth of four was imposed on each forest.

3.2.5 Gradient Boosting

Another ensemble method that combines weak decision-tree regressors to form a strong group regressor, Gradient Boosting instead seeks to optimize the gradient of the loss function for each regressor. As this can perform well with the correct hyperparameters, we include this to see if the algorithm can outperform any of the aforementioned algorithms. In addition to using regular Gradient Boosted Trees, we also use an optimized version of this algorithm known as Extreme Gradient Boosted Trees (XGB). For Gradient Boosting a least-squares loss function, along with a learning rate of 0.1, and 100 estimators were used. For XGB a grid search was performed to find the optimal hyperparameters. Respectively, the results for these algorithms are reported under Grad and XGB.

3.3 Neural Networks

Much work has already been done implementing neural networks in many different forms, including CNN, CRNN, DCNN, LSTM, Jump neural Networks, and Echo State (see [1], [2], [3], [4], [6], [8], and [15]). Much of this work came from the Blood Glucose Level Prediction Challenge (BGLP) in 2018 using the OHIO T1DM data set.

3.3.1 ANFIS

ANFIS is a neural network that includes fuzzy logic principles. Fuzzy logic is about partial truths. Most neural networks have a true/false form in selections. Fuzzy logic models uncertainties. Some examples of this are what one considers warm/cold, fast/medium/slow, or high/low. Rather than just picking one or the other, a draw from a distribution can give a weighted random nature to the choices. ANFIS is designed to approximate nonlinear functions like glucose values. This was chosen due to the extremely accurate predictions in the referenced paper on chaotic systems. [9]

3.3.2 Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) is a fully-connected, feed-forward neural network. This neural network can often find higher-order terms without having to create these higher-order terms. This reduces feature engineering of the data. Our MLP consists of three hidden layers, each with 100 nodes, and ReLu activation functions. The output layer for the regression is merely the output of the last affine function. Results are reported under MLP.

4 Metrics

The following metrics were used when evaluating the efficiency and accuracy of the algorithms:

4.1 Root Mean Square Error

The root mean square error (RMSE) is defined as $\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$ where \hat{y}_i is the predicted value and y_i is the actual value. RMSE has the advantage of an easily defined gradient, easy interpretability, and taking the square root of the squares transforms the error back to the original function space (that is, the RMSE value is in the same units as our label). This is the first metric used in evaluating the accuracy of the regression models.

4.2 Mean Absolute Error

The mean absolute error (MAE) is defined as $\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$. This error function is easy to define, is fairly robust against outliers, and will be in the same units as our label. However, the gradient is not always easy to define (and may not exist). This is the second metric used in evaluating the accuracy of the regression models.

4.3 Coefficient of Determination

The coefficient of determination (R^2) is defined as

$$1 - \frac{\sum_{i=1}^n \epsilon_i^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

where y_i is the actual value, \hat{y}_i is the predicted value, $\epsilon_i = y_i - \hat{y}_i$ and is defined as the i^{th} residual, and \bar{y} is the sample mean. The coefficient of determination gives a measure of how much variance is explained by the model. Values near 1 indicate nearly all variance is explained by the model, while values near 0 indicate the variance may be caused by other factors. We note that negative values are possible, and for this paper indicate poor performance from the model.

4.4 Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) is defined as $\frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$ where TP, FP, FN, TN stand for the true positive, false positive, false negative, and true negative rates respectively [4]. This metric gives a general idea of how well an algorithm does in predicting glycemic events. Values near 1 show the predictions correlate with the actual glycemic events. Values near 0 indicate the algorithm does no better than random guessing. Values near -1 indicate negative correlation (that is the predictions correlate with the opposite of the glycemic event). This metric is commonly used by many articles that attempt to predict blood glucose levels (see [4] for one such example), and as such is used here.

4.5 Clarke Error Grid

The Clarke Error Grid plots the actual blood glucose values against the predicted blood glucose values and is used as an indication of the potential results that may occur for a given prediction. The grid is split into 5 zones A-E. Predictions in Zone A and B are generally considered safe predictions and would not result in any negative effects on the patient. Predictions in Zone C would result in unnecessary treatment. Predictions in Zone D indicate a potentially dangerous failure to detect a glycemic event. Predictions in zone E would confuse treatment of hypoglycemia for hyperglycemia and vice versa (see [1]). Points in Zone E are considered extremely dangerous, as treatment due to these results could result in the patient's death. For this paper, in addition to MCC we use the percentage of points within each zone to evaluate the accuracy of a model's predictions.

5 Results

The following tables describe the average of the metric scores from the 6 patients. Each of these metrics are described above, namely RMSE, MAE, MCCs, and R^2 . The abbreviation definitions and explanations can be found in the Methods section above.

Table 1. Metric Averages for 30-minute Prediction Horizon

Method	RMSE	MAE	MCC _t	MCC _h	R ²
OLS	20.53	14.14	0.34	0.79	0.86
Lasso	20.58	14.22	0.32	0.79	0.85
Ridge	20.52	14.13	0.35	0.79	0.86
Elastic	20.56	14.20	0.31	0.79	0.86
RBF	24.89	16.96	0.14	0.74	0.79
Poly	31.73	22.51	-0.00	0.70	0.66
KNN	24.57	17.07	0.30	0.73	0.79
RF	23.00	16.27	0.16	0.76	0.82
Grad	21.37	14.87	0.17	0.78	0.84
XGB	24.62	17.29	0.34	0.74	0.79
Kalman	24.08	24.08	0.40	0.74	0.78
UKF	29.88	20.65	0.30	0.67	0.69
ARIMA	23.73	16.68	0.12	0.75	0.81
VAR	25.25	17.05	0.36	0.74	0.79
ANFIS	24.56	16.52	0.26	0.76	0.80
MLP	20.85	14.30	0.30	0.78	0.85

Table 2. Metric Averages for 60-minute Prediction Horizon

Method	RMSE	MAE	MCC _t	MCC _h	R ²
OLS	33.42	24.65	0.02	0.61	0.62
Lasso	33.41	24.67	0.02	0.61	0.62
Ridge	33.41	24.65	0.02	0.61	0.62
Elastic	33.40	24.67	0.02	0.61	0.62
RBF	36.76	26.53	-0.00	0.55	0.54
Poly	39.16	29.31	-0.00	0.53	0.48
KNN	38.11	28.01	0.15	0.53	0.50
RF	35.20	26.08	0.09	0.58	0.58
Grad	33.98	24.96	0.08	0.58	0.61
XGB	39.78	26.97	0.15	0.53	0.46
Kalman	22.77	15.28	0.41	0.75	0.81
UKF	29.78	20.65	0.30	0.66	0.69
ARIMA	36.39	26.93	0.01	0.56	0.54
VAR	35.06	19.56	0.16	0.70	0.54
ANFIS	36.87	26.53	0.12	0.59	0.56
MLP	35.59	25.81	0.06	0.59	0.57

6 Analysis

In an attempt to first analyze the accuracy of these predictions we first analyze the RMSE and MAE for both the 30-minute and 60-minute prediction horizons (Tables 1 and 2). As a general guideline we will first analyze which model we believe is performing best among the patients. Once this is done we will then analyze general trends we have noticed while analyzing this data.

6.1 30-Minute Prediction

We note that in terms of the above defined metrics OLS, Lasso, Ridge, and Elastic Net Regression perform nearly identical. Thus, since the differences between OLS, Ridge, Lasso, and Elastic Net regression yield minimally different results, we consider Lasso to be the best model for the 30-minute blood glucose predictions. Lasso regression offers a natural form of feature selection which allows us to analyze which lags are most important for predicting future blood glucose levels. A further analysis of the feature relevancy can be found under section 6.4.

Even though we have identified Lasso regression as the best performing algorithm among those tested for the 30-minute prediction horizon, this means little if this "best" algorithm still yields subpar results. As such, we analyze Lasso regression both in terms of MCC and the Clarke Error Grid to determine if these results are "sufficiently adequate" for blood glucose prediction. To see general trends for the prediction we analyze the results for actual and predicted values across time for patients 540 and 584.

Note the Clarke Error Grid for patients 540 and 584 for the 30-minute prediction horizon (figure 2). The closer the points fall onto the bottom left to top right diagonal the better the predictions are considered. Analyzing these plots visually does not raise any immediate concerns for the predictions. Most values appear to fall within zones A, B, and C. Analyzing the zones percentages (table 3) shows that Lasso has 96% accuracy for patient 540 and about 99% accuracy for patient 584. The major concern however is that the rest of these predictions fall within zones D-E, indicating these predictions may result in potentially dangerous care if acted on for the patient. Considering the high accuracy for each patient though, these results are considered "sufficiently accurate" for the 30-minute prediction horizon.

Analyzing the MCC for Lasso regression for the 30 minute horizon shows that the MCC tends to be about twice as high for hyper-

glycemic events than for hypoglycemic events. Given that the data tends to have many more values in the hyperglycemic range than the hypoglycemic this reflects more on the class imbalance more than the algorithm. This is seen due to all the algorithms having this trend. Further, this bias is reflected in the algorithm's predictions, as valleys in the predictions do not reach as low as the valleys in the actual data (see figure 1). Because of this, we note that the algorithms are less likely to predict hypoglycemic events as they are hyperglycemic events, a result that occurs due to the higher number of blood glucose values in the data.

6.2 60-Minute Prediction

Looking at the results for the 60-minute prediction horizon for the RMSE and MAE we find the surprising result that the Kalman Filter (not the Unscented Kalman Filter), performs best out of all the algorithms. Several explanations are possible as to why this occurs. One of these is that the Kalman filter seemed to dampen the predictions. Most of the other algorithms would keep predicting upwards for the hour predictions if the trend was going up beforehand. The Kalman filter seems to mainly shift the prediction horizon over (so the difference between the last known glucose value and the prediction for an hour later is minimal). Since it keeps the results in the typical ranges of glucose values it may avoid the poor scores from unusually strong spikes of predicted values. The scores may be the best, but they may still be very poor predictors for an hour out.

Considering the aforementioned problems with the Kalman filter, we analyze the "second" best algorithm. Since the general trends discussed in the 30-minute prediction horizon section still hold for the 60-minute prediction horizon (when we disregard the Kalman Filter), we conclude Lasso regression to be the next best algorithm to use. However, analyzing the difference between the 30-minute prediction horizon and the 60-minute prediction horizon raises several concerns with using Lasso regression for the 60-minute prediction horizon.

We noted earlier that Lasso regression tends to underfit with regards to hypoglycemic events. This problem is only exacerbated when the prediction horizon is extended to 60 minutes (see table 2). Here we notice the hypoglycemic MCC has reduced to near 0, indicating that Lasso prediction does no better than random guessing as to whether a hypoglycemic event is occurring. This is far from ideal for any diabetic patient. As well, we note that for the 60-minute prediction horizon, the accuracy of safe predictions degrades by about 2-3% (see table 3). While 94-97% accuracy is still fairly good, given that this reduction in accuracy results in 2-3% more dangerous predictions, and considering the fact that Lasso regression is unable to predict hypoglycemic events better than random guessing, we do **not** consider these predictions to be "sufficiently accurate" for the 60-minute prediction horizon. As such, our recommendation is to use the 30-minute prediction horizon.

6.3 Overall Trends

The biggest trend that we notice is that the models tend to underfit in regards to hypoglycemic events. That is, the predicted values do not reach as low as the actual blood glucose values do. This is noted in the hypoglycemic MCC for the 30-minute prediction horizon (see table 1) which gives on average a score at about 0.3. This indicates a general correlation in predicting hypoglycemic events, but not a strong one. Given that the average blood glucose levels on the test data were 159.42 mg/dl, 158.51 mg/dl, 134.92 mg/dl, 143.41 mg/dl,

172.71 mg/dl, and 148.23 mg/dl for patients 540, 544, 552, 567, 584, and 596 respectively the most likely reason that the MCC for hypoglycemic events is so low is due to class imbalance within the glucose levels. Since most glucose levels are generally high for the patients, the model overfits for higher glucose levels, and as such struggles to predict hypoglycemic events. A potential solution could be to upsample by "jittering" the smaller imbalanced class (adding small random perturbations to the existing smaller imbalanced class in order to create for data). See [7] and [10] for such an example.

6.4 Feature Relevancy

As stated earlier, one important benefit of Lasso regression is the ability to identify features important to glucose prediction. As seen in Table 4: glucose level, bolus, meal, and exercise are significant in predicting glucose levels (finger sticks are potentially significant, but they may be linearly dependent on glucose level). The Weights column is the sum of all 6 people's weight scores. The problem with the weights is the huge variability in the number of recorded data points. In an attempt to normalize the data, we created an Adjusted Weight. This is made by dividing the weights of each person by the number of recorded values for each person and summing all 6 of them together. This was multiplied by 1000 so the values would be about the same magnitude as the original weights. The lack of enough data for exercise is demonstrated here. Only 3 of the 6 people had values for exercise and one of them had only 4 values. This person in the Adjusted Weights had a score of 32 while the other two were about 1.5 and 2. More data points for these other categories would reduce the variance and more clearly identify what features are important.

7 Conclusion

We found that Lasso regression performed best out of the algorithms used for both the 30-minute prediction horizon and the 60-minute prediction horizon. While the results were adequate for the 30-minute prediction horizon, these quickly degraded for the 60-minute horizon. We found in general that the regression algorithms perform fairly well for predicting hyperglycemic events, but struggle for predicting hypoglycemic events. It is our opinion that further research should be done with regards to improving the prediction horizon for blood glucose prediction. Specifically, further research should be investigated into the effects of the volume of data on the prediction horizon. If an artificial pancreas is to become a reality, stable prediction horizons beyond 30-minutes are needed.

Furthermore, analyzing the coefficients of the Lasso model shows that glucose level, bolus, meal, and exercise are the most relevant features in producing forecasts for blood glucose levels. However, problems with sparsity among certain features reduce the relevancy of these features. As such, future research should include handling sparse features in a more robust way.

8 Additional Material

For those wishing to compare or reproduce work found in this paper, the related code can be found at <https://github.com/marshallb95/BloodGlucosePrediction/blob/master/Master.ipynb>.

REFERENCES

- [1] A. Aliberti, I. Pupillo, S. Terna, E. Macii, S. Di Cataldo, E. Patti, and A. Acquaviva, 'A multi-patient data-driven approach to blood glucose prediction', *IEEE Access*, **7**, 69311–69325, (2019).
- [2] J. Chen, K. Li, P. Herrero, T. Zhu, and G. Pantelis. Dilated recurrent neural network for short-time prediction of glucose concentration. Paper presented at the Third International Workshop on Knowledge Discovery in Healthcare Data at the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, 2018.
- [3] S. Fiorini, C. Martini, D. Malpassi, R. Cordera, D. Maggi, A. Verri, and A. Barla. Data-driven strategies for robust forecast of continuous glucose monitoring time-series. Paper presented at the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017.
- [4] 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*, **24**, 603–613, (2019).
- [5] C. Marling and R. Bunescu. The ohio1dm dataset for blood glucose level prediction: Update 2020. In The 5th International Workshop on Knowledge Discovery in Healthcare Data, Santiago de Compostela, Spain, June, 2020, 2020.
- [6] J. Martinsson, A. Schliep, B. Eliasson, C. Meijner, S. Persson, and O. Mogren. Automatic blood glucose prediction with confidence using recurrent neural networks. Paper presented at the Third International Workshop on Knowledge Discovery in Healthcare Data at the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, 2018.
- [7] M. Mayo, L. Chepulis, and R. Paul, 'Glycemic-aware metrics and over-sampling techniques for predicting blood glucose levels using machine learning', *PLoS ONE*, **14**, 0225613–0225632, (2019).
- [8] C. Midroni, P. J. Leimbiger, G. Baruah, M. Kolla, A. J. Whitehead, and Y. Fossat. Predicting glycemia in type 1 diabetes patients: Experiments with xgboost. Paper presented at the Third International Workshop on Knowledge Discovery in Healthcare Data at the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, 2018.
- [9] A. Miranian and M. Abdollahzade, 'Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction.', *IEEE Transactions on Neural Networks and Learning Systems*, **24**, 207–218, (2013).
- [10] N. Nnamoko and I. Korkontzelos, 'Efficient treatment of outliers and class imbalance for diabetes prediction', *Artificial Intelligence in Medicine*, **104**, 101805–101817, (2020).
- [11] S. M. Pappada, M. H. Owais, B. D. Cameron, J. C. Jaume, A. Mavarez-Martinez, R. S. Tripathi, and T. J. Papadimos, 'An artificial neural network-based predictive model to support optimization of inpatient glycemic control', *Diabetes Technology & Therapeutics*, **22**, 1–12, (2020).
- [12] I. Rodríguez-Rodríguez, J.V. Rodríguez, I. Chatzigiannakis, and M.A. Zamora, 'On the possibility of predicting glycaemia 'on the fly' with constrained iot devices in type 1 diabetes mellitus patients', *Sensors*, **19**, 4482–4496, (2019).
- [13] I. Rodríguez-Rodríguez, I. Chatzigiannakis, J.V. Rodríguez, M. Maranghi, M. Gentili, and M.A. Zamora, 'Utility of big data in predicting short-term blood glucose levels in type 1 diabetes mellitus through machine learning techniques', *Sensors*, **19**, 4538–4557, (2019).
- [14] E. A. Wan and R. V. D. Merwe. The unscented kalman filter for nonlinear estimation. Adaptive Systems for Signal Processing, Communications, and Control Symposium, 2000.
- [15] J. Xie and Q. Wang. Benchmark machine learning approaches with classical time series approaches on the blood glucose level prediction challenge. Paper presented at the Third International Workshop on Knowledge Discovery in Healthcare Data at the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, 2018.

Table 3. Clarke Error Grid percentages

Zone	30 min		60 min	
	540	584	540	584
Zones A-B	0.96	0.99	0.935	0.97
Zone C	0.0	0.00	0.001	0.01
Zones D-E	0.04	0.01	0.064	0.02

Table 4. Lasso Significant Values Totals

Feature	Number Recorded	Weights	Adjusted Weights
glucose_level	77563	15.4654	1.2062
basis_gsr	39542	0.2272	0.03560
skin temperature	39540	0.2418	0.0295
acceleration	39542	0	0
finger stick	1669	0.54	2.4504
basal	428	0	0
temp basal	208	0	0
bolus	1994	9.4944	23.4776
meal	957	3.5682	31.6974
stressors	2	0	0
exercise	65	0.2312	36.2337

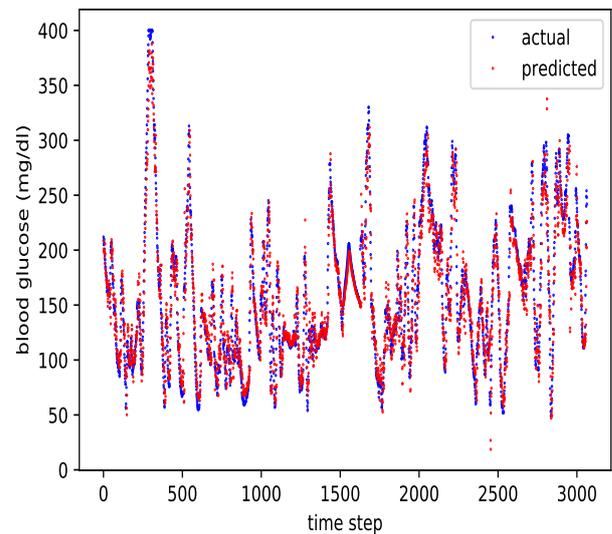


Figure 1. Patient 540 prediction results for 30 min PH with Lasso regression

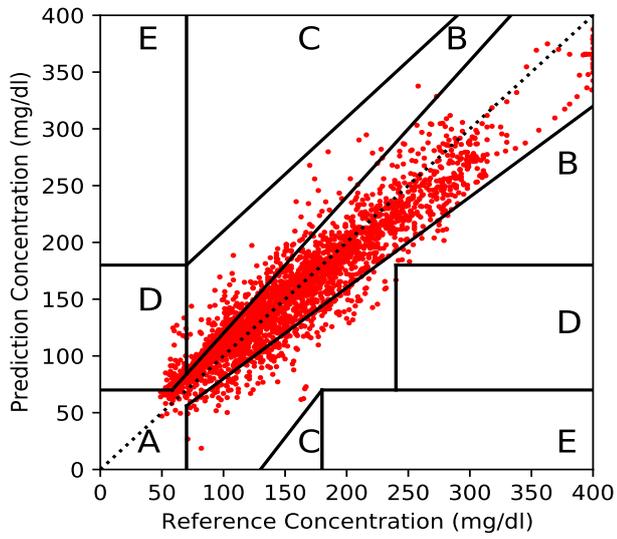


Figure 2. Patient 540 Clarke Error Grid for 30 min PH with Lasso regression

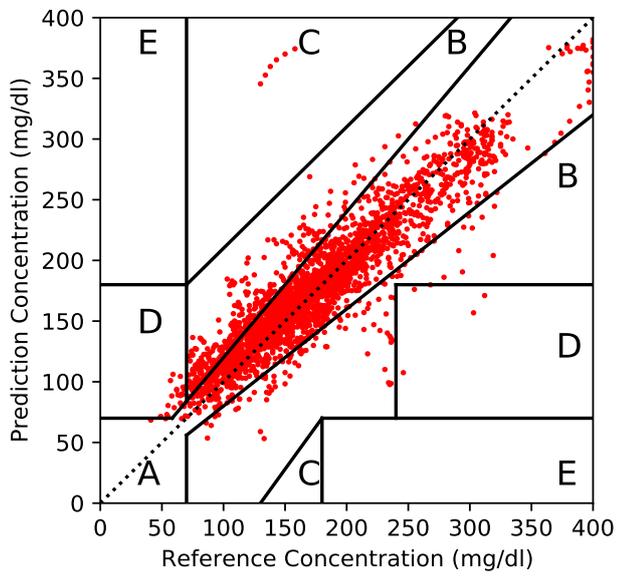


Figure 3. Patient 584 Clarke Error Grid for 30 min PH with Lasso regression