Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

Blood Sugar Prediction with an Improved Mechanism for Diabetic using Recurrent Neural Networks

A K Saritha[1], I Jeena Jecob[2], Beena G Pillai[3], Madhurya J A[4],Dr.Dayanand Lal N[5]
[1,3,4,5]Assistant Professor, Department of CSE, GITAM School of Technology, Bengaluru, India
[2]Associate Professor, Department of CSE, GITAM School of Technology, Bengaluru, India
[1]savikkal@gitam.edu, [2]ijacob@gitam.edu, [3]bpillai@gitam.edu, [4]mambuja@gitam.edu,
[5]dnarayan@gitam.edu

Abstract:In the Current 2020 Covid-19 Situation, less expensive sensors measuring sugar levels in continuous intervals are essential. These, combined with easily usable devices plugged with SAAS based Deep Learning Solutions, allow for customized health and disease management grades. Deep Learning Systems scalable technologies are allowed by sensors, analytical tasks, and people. This brings us to the question of volume for creating such DL Solutions, which we can present for the current situation of the COVID-19 pandemic. The technique is based on scratch-trained RNN, requiring the amount of sugar of a patient in graded detail. The model predicts and estimates sugar and possible health issues accurately. There is no need for a reduction in this pre-processing system and function, which saves the calculation.

Keywords: Real-time Sugar Monitoring System, Recurrent Neural Networks, Blood Sugar

I. INTRODUCTION

Shortly the world will be recording the relevant quantifiable data, which leads to unprecedented temporal resolution. The broad adoption of real-time sugar monitoring systems (RSMS) provides a helpful system for closely capturing and reacting to their current blood sugar activities for type 1 diabetics (T1D). And are several factors that represent practices in which we participate and physiologically and medically related activities. Tables that depend on several different characteristics (proteins, insulin injections, exercise, hormones, sleep imbalances, body shape, etc.) comply with blood sugar levels. It makes predicting near term sugar changes a mammoth task and developing deep learning methods or algorithms. Improvements in sensor/hardware technology applied to the DL strategy.But, having domain knowledge, isn't easy to generate different hardware on a scale.

Many times natural, structured variables might be too complex to understand for end-users. Many approaches are available as deep learning technology advances to solve complex indicators such as attributes, classifiers, and predictors are continually learn—a scalable solution for cumbersome DL problems for accurate health solutions.

The assumption for our methods are:

- Predicting sugar levels from glucose levels alone

is viable.

- Non-experts can train functional model-related models with Engineering or advanced procedures for training
- In their forecasts, models can quantify uncertainties to warn users of the need for extra care or cautions auxiliary input.
- Functions of physiologically motivated failure increase the efficiency of forecasts.

We trained and tested our technique on the UCLA Blood Glucose Level Prediction Dataset;

II. BASIC APPROACH

For several years, recurrent neural networks (RNNs) have been the solution to most issues dealing with sequential data and Natural Language Processing (NLP) issues. Their variants, such as the LSTM, are still widely used in different state-of-the-art models to date. It covers the basic concepts of RNNs in this post and introduces a simple vanilla RNN model.

Traditional feedforward neural networks all take in a fixed amount of input data at the same time and produce a fixed output quantity each time. On the other hand, RNNs don't absorb all the input data at once. Instead, they take them one at a time and in a series. Before producing production, the RNN does a series of calculations at

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

each point. The output, known as the secret state, is then combined with the next input to produce another output.

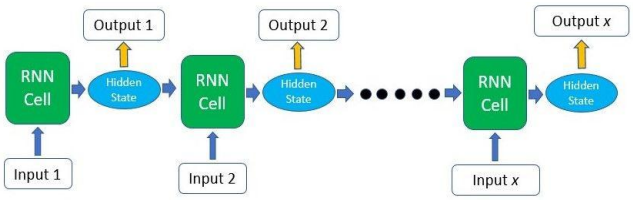

Figure 1: The LSTM network

LSTM network used in this work is a high-level the internal memory vector ci is modified by each cell, with current input information, and outputs a vector hi. Ci and hi are pass to the next cell. It is used as an input to a fully linked output layer. That implements a linear transformation and outputs the intended μ, σ2.

Traditional feedforward neural networks all take in a fixed amount of input data at the same time and produce a fixed output quantity each time. On the other hand, RNNs don't absorb all the input data at once. Instead, they take them one at a time and in a series. At each point, the RNN performs a series of calculations before generating output. Then the output, known as the hidden state, is combined with the next input in the series to produce another output until this process begins.

$$\begin{aligned} \boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\ \boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\ \boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\ \hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}) \end{aligned}$$

With the gradient we just obtained, we can change the model weights accordingly so that future computations with the input data can generate more accurate results. The weight here refers to the weight matrices multiplied with the input data and hidden states during the forward transfer. Back-propagation is called this whole method for measuring the gradients and modifying the weights. Back-propagation is looped repeatedly in conjunction with the forward transfer, allowing the model to become more precise with its outputs each time as the weight matrices values are adjusted to pick out the data patterns.

[μ, σ2] = Wlht + bl.

Though it can seem as if each RNN cell uses a different weight, as seen in the graphics, all the weights are the same as the reuse of the RNN cell in the process. Therefore, only the input data and hidden state transmitted are similar in and of the time. Though it can seem as if each RNN cell uses a different weight, as seen in the graphics, all weights are the same as the reuse of the RNN cell in the process. In and step of the time, only the input data and hidden state transmitted are similar.

It centered on the Gaussian probability density equation, the negative log-likelihood (NLL) loss function.

$$L = \frac{1^{\Sigma k}}{k_{i=0}} - \log N(yi|\mu i, \sigma\, i$$

If yi is the data-target value, and μi, given the xi input sequence, σi is the network output. This method of prediction modeling makes it easier to base decisions on predictions.

Physiological loss feature: The model has been trained with a loss function unique to glucose[Favero et al. A measure that incorporates the mean square error with a penalty word for forecasts leading to clinically risky therapies, 2012]

II. OBSERVATIONS AND RESULT

our references note the training values of glucose captured by recording it by 0.01

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

Parameter identificationwas made by identifying sample patient data of 900 individuals of UCI Diabetes Dataset for Diabetes prediction forecast, and train on 70:20:10 as training, testing, and validation. We are training four models with a different set of models of M-RNN and Fast-RNN. The base model of RNN is LSTM, with different layers randomly selected 8,16,32 and 128. Each model trained for 300Epochs yielded greater accuracy has depicted in the figure. These results were based on the data collected for 30 mins,45mins, and 60mins intervals Before fasting and post fasting. The problem of selecting an actual model plays a key role in future work.

Final Models: Using a glucose level background of 30 minutes, the final models equipped for projections 30 and 60 minutes into the future, respectively.The final training setup was training on the first 80 percent of the glucose level training data for each patient and validating the last 20 percent.A low learning rate of 10-5, a maximum number of 10,000 epochs, and an early stop patient of 256 were given to the final models to allow them more time to converge. Those final models were then the only models that worked on the test data provided.Note that there are values for which no predictions exist in the test data.

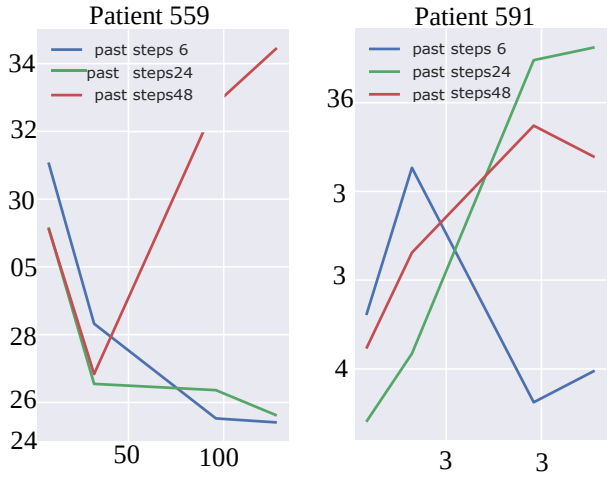

Figure 2: On the validation results, the RMSE score is shown to select the number of LSTM states and the number of previous blood glucose time (history) measures to estimate future value.

Missing data: This is a widely used method for handling null values. If it has a null value for a particular function, we either delete a particular row and a specific column if it has more than 70-75 percent of the missing values. Only when there are enough samples in the data set is this approach recommended. One has to ensure that there is no added bias after we have deleted the data. Removing the data would result in data loss that does not offer the expected results when predicting the data.

For example, a categorical attribute would have a definite number of possibilities, such as gender. They have a definite number of classes sincewe may allocate another class for the missing values. Here, the Cabin and Embarked features have missing values replaced with a new category, for example, U for 'unknown'. This method adds more details to the dataset that contribute to the variance change. We need to find because they are categorical, one-hot encoding to convert it to a numeric type for the algorithm to understand them.
Computational Requirements: The model training in our experimental setup performed on a commodity laptop.The model is small enough to fit and can use on mobile devices (e.g., mobile phones, blood glucose monitoring devices, etc.). Training may initially be carried out offline; training would be light enough to train either on the devices or offline.

III. RESULTS

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

The results given in Table 1 are the models' root mean squared error (RMSE) when trained with the loss function of mean squared error (MSE) and the loss function of negative log probability (NLL). The findings indicate the efficiency of the model.It is comparably equivalent when trained with NLL and MSE but with the additional benefit of estimating the forecast variance.

	30 min horizon			60 min horizon		
Patient ID	NLL	MSE	t_0	NLL	MSE	t_0
559	19.5	19.5	23.4	34.8	34.4	39.7
570	16.4	16.5	19.0	28.8	28.6	31.9
588	19.3	19.2	21.8	32.5	33.1	35.8
563	19.0	19.0	20.8	30.8	29.9	34.0
575	24.8	24.2	25.6	38.4	37.3	39.7
591	25.4	22.0	24.4	36.0	36.0	38.6
μ	20.7	20.1	22.5	33.6	33.2	36.6
σ	±3.2	±2.5	±2.2	±3.2	±3.1	±3.0

As seen in Table 1, where patient 570's RMSE is 16.3 and patient 570's RMSE is 16.3, the glucose level of patient 900 is more difficult to calculate than patient 900's glucose level.

Table 1: The results in the measurement of glucose levels with intervals of 30 and 60 min are seen individually per patient and on average.The table shows the predictions of the root mean square error (RMSE) when the LSTM retrained with the negative log-likelihood (NLL) loss function and the mean square error (MSE) loss function. To estimate the last value of t0., it refers to the naive baseline.

Figure 3: The projection (red) and standard deviations shownFor 570 (top) and 900 (bottom) patients (shaded orange) comparedto ground truth (green). Notice the much greater confusion for the patient

For Patient 591, the RMSE is 24.6.Abb. Fig. 3 shows that by assigning a larger variance to the predictions for patient 591 than for patient 570, the model can learn this.In the figure, the pink shaded field indicates the standard deviation. In Fig.4, the Clarke error grid plots further illustrate this. The Clarke error grid plots demonstrate this further in Fig.4. We can see that most 570 patient predictions are in region A, which is considered a clinically secure region.For patient 591, we can see that in region B, which is considered non-important, there are more estimates and region D, which is more important. In the estimates, the variance of the error is greater for patient 591 than for patient 570. The model has a difficult time predicting hypoglycemic events, in particular.

IV. DISCUSSION

We concentrate on unique experiences we have learned during the system development as the competition will provide benchmarking.

Minimalistic ML: By using the holdout method, as described above, we computed a point estimate of our model generalization precision.Indeed, a confidence interval around this estimate may not only be more informative and useful in particular applications, but our point estimate may be very sensitive to the unusual training/test split (i.e. suffering from high variance), nested cross-validation, which may be the most common approach to model evaluation for hyperparameter tuning or recognition processes.

Quantifying uncertainty: An estimation of the forecast variance is also given by our model, thus measuring forecast uncertainty. This is a beneficial function of a scheme to make decisions about insulin administration and caloric intake by continuous glucose monitoring customers.We believe that the large-scale processing of multiple user data boosts the output further. In this quantification of ambiguity, the results in Fig.3 show the two ends of the spectrum.
Thecentral problem is that it is difficult to differentiate between intra-patient variation and sensor errors.
A significant research issue is methods that can detect sensor deterioration over time or recognize defects through long-term physiological comparison of sensors for the same patient; it is uncertain whether the regularly smoothed sensor-generated data is adequate for that.

Loss feature: We randomly divide our available data into two subsets: training and testing. Our work-around is to set aside test data to cope with imperfections in non-ideal settings, such as limited data and resources, and the inability to obtain further delivery data.Here, our learning algorithms test set interpret new, unseen data; it is essential that we only touch the test set once to ensure that we do not introduce any

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

bias when we estimate the generalization accuracy.Usually, we distribute 2/3 of the training set and 1/3 of the test set performance. 60/40, 70/30, 80/20, or even 90/10 are other common training/test splits.

Patient 570 Clarke Error Grid Reference Concentration (mg/dl).
Figure 4 : For patttient 570 (top) and patient 591 (bottom), we present the Clarke error grid plots. Notice that for patient 591, the predictions error variance is more significant than for patient 570.

Model selection: A forecast output model illustrates how well it performs on new data. In machine learning applications or designing new algorithms, predicting future data is still the

TABLE II: PREDICTION RESULTS OF DIFFERENT METHODS

PH = 15 minutes				
Methods	***RMSE***	***CC***	***Time Lag***	***Fit***
ARIMA	12.256	0.972	10.192	76.425
SVR	11.694	0.973	9.808	77.565
LSTM	**11.633**	**0.974**	**9.423**	**77.714**
PH = 30 minutes				
Methods	***RMSE***	***CC***	***Time Lag***	***Fit***
ARIMA	22.924	0.903	22.885	55.923
SVR	22.135	0.904	20.769	57.644
LSTM	**21.747**	**0.909**	**20.385**	**58.523**
PH = 45 minutes				
Methods	***RMSE***	***CC***	***Time Lag***	***Fit***
ARIMA	32.588	0.806	37.885	37.463
SVR	30.628	0.812	34.423	41.595
LSTM	**30.215**	**0.818**	**32.692**	**42.563**
PH = 60 minutes				
Methods	***RMSE***	***CC***	***Time Lag***	***Fit***
ARIMA	40.841	0.698	52.885	21.694
SVR	37.422	0.709	47.885	28.893
LSTM	**36.918**	**0.722**	**46.346**	**30.079**

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

TABLE III: PREDICTION PERFORMANCE OF LSTM METHOD WITH DIFFERENT PRE-TRAIN EPOCHS (FOR PH=30 MINUTES)

Epochs	*RMSE*	*CC*	*Time Lag*	*Fit*
100	22.649	0.902	21.731	56.749
200	22.509	0.904	21.539	57.077
300	22.317	0.904	21.346	57.448
400	22.140	0.906	20.577	57.778
500	22.021	0.907	20.962	58.004
600	22.014	0.907	20.577	58.010
700	22.064	0.906	20.769	57.918
800	22.035	0.907	20.962	58.039
900	21.828	0.908	20.577	58.338
1000	22.113	0.905	20.769	57.760
1100	22.137	0.907	20.962	57.813
1200	22.082	0.907	20.962	57.933
1300	21.747	0.909	20.385	58.523
1400	22.025	0.906	20.962	57.967
1500	21.931	0.908	20.577	58.173
1600	22.008	0.907	20.577	58.070
1700	21.804	0.908	20.385	58.429
1800	21.850	0.908	20.577	58.343
1900	21.848	0.907	20.769	58.315
2000	21.982	0.907	20.769	58.113

basic problem we want to overcome.Machine learning typically requires a great deal of experimentation, but internal knobs, the so-called hyperparameters, are calibrated for learning algorithms. Running a learning algorithm over a training dataset with multiple hyperparameter settings will result in different models.Since we typically want to pick from this set the best-performing model, we need to find a way to estimate their performance to score them against each other. Going one step beyond mere fine-tuning algorithm, we normally not just play with the single algorithm that we think would be the "right solution" under the given circumstances. We want to compare different algorithms more often than not with each other, often in predictive and computational performance.

Substitution: Inarguably, the holdout technique is the most accessible form of model evaluation. We take and divide our labeled dataset into two components: a training set and a test set. Then, we tailor a model to the specifics of the training and predict the test set labels. And the fraction of accurate predictions is our prediction accuracy measurement; we retain the known test labels during prediction, of course, on the same training dataset (this is called resubstitution evaluation).We do not want to test and evaluate our framework even though it would introduce a very positive bias because of overfitting. In other words, we can't say whether or not the model merely memorized the training data, or whether new, unknown data is well generalized. (We can quantify this so-called optimism bias on a side note as the difference between the accuracy of the planning and the precision of the test.)

V. CONCLUSION

Data science area is looking for massive data and protocols for assessing. Open sourcing the data from different hospital organizations ensure that more data and different geography help for evaluation. We are having unbalanced data of sensors, which helps to assess different factors. Analysis of sensor errors would lead to the prediction of a more accurate model. Another issue is pre-processing in the sensors, which should be a need for intellectual property rights. For instance, in Genetic analysis or DNA Probing, all the analyses are performed in vendors software, which fine-tuned the result for understanding all metrics.

REFERENCES

[1] Kochanek KD, Murphy SL], Xu J, et al. Deaths: Final data for 2014. National vital statistics reports; vol 65 no 4. Hyattsville, MD: National Center for Health Statistics. 2016.

[2] Chrvala, C. A., et al. (2016)]. Diabetes self-management education for adults with type 2 diabetes mellitus: A systematic review of the effect on glycemic control. in Patient Education and Counseling journal

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1394 - 1400
Received 20 January 2021; Accepted 08 February 2021.

[3] Schellenberg], E. S., Dryden, D. M., Vandermeer, B., Ha, C., & Korownyk, C. (2013). Lifestyle interventions for patients with and at risk for type 2 diabetes: a systematic review and meta-analysis. Annals of internal medicine, 159(8), 543-551

[4] Hood, M., Wilson, R.], Corsica, J., Bradley, L., Chirinos, D., & Vivo, A. (2016). What do we know about mobile applications for diabetes self-management? A review of reviews. Journal of behavioral medicine, 39(6), 981-994.

[5] Broadbent, E., Kumar, V., Li, X., Sollers 3rd], J., Stafford, R. Q., MacDonald, B. A., & Wegner, D. M. (2013). Robots with display screens: a robot with a more humanlike face display is perceived to have more mind and a better personality. PloS one, 8(8), e72589

[6] Tang, T. S., Funnell, M. M., Brown,] M. B., & Kurlander, J. E. (2010). Self-management support in "real-world" settings: an empowerment-based intervention. Patient education and counseling, 79(2), 178-184

[7] Cole-Lewis, H., & Kershaw, T. (2010).] Text messaging as a tool for behavior change in disease prevention and management. Epidemiologic Reviews, 32(1), 56-69

[8] Edwin V Bonilla, Kian Ming Adam Chai,] and Christopher KI Williams. 2008. Multi-task Gaussian process prediction. In Advances in Neural Information Processing Systems. Curran Associates, Inc., Red Hook, NY, USA, 153–160.

[10] Razvan Bunescu, Nigel Struble, Cindy Marling, Jay Shubrook, and Frank Schwartz]. 2013. Blood glucose level prediction using physiological models and support vector regression. In Proceedings of IEEE International Conference on Machine Learning and Applications, Vol. 1. IEEE Press, Piscataway, NJ, USA, 135–140