

Blood Glucose Prediction Using Convolutional Long Short-Term Memory Algorithms

Redy Indrawan ^{1*}, Siti Saadah ¹, Prasti Eko Yunanto ²

Informatics Study Program

Telkom University

Bandung

²Informatics Department

Telkom University

Bandung

*redy.indrawan@gmail.com, gppras@telkomuniversity.ac.id

Abstract-Diabetes Mellitus is one of the preminent causes of death to date. Effective procedures are necessary to prevent diabetes and avoid complications that may cause early death. A common approach is to control patient blood glucose, which necessitates a periodic measurement of blood glucose concentration. This study developed a blood glucose prediction system using a convolutional long short-term memory (Conv-LSTM) algorithm. Conv-LSTM is a variation of LSTM algorithms that are suitable for use in time series problems. Conv-LSTM overcomes the lack in the LSTM algorithm because the latter algorithm cannot access the content of previous memory cells when its output gate has closed. We tested the algorithm and varied the experiment to check the effect of the cross-validation ratio between 70:30 and 80:20. The study indicates that the cross-validation using a ratio of 70:30 data split is more stable compared to one with 80:20 data split. The best result shows a measure of 21.44 in RMSE and 8.73 in MAE. With the application of conv-LSTM using correct parameters and selected data split, our experiment attains accuracy comparable to the regular LSTM.

Keywords: diabetes mellitus, blood glucose, recurrent neural network, long short-term memory

Article info: submitted May 30, 2021, revised July 21, 2021, accepted August 19, 2021

1. Introduction

Diabetes is a chronic disease with a gradual growth apparent through an increase in blood glucose. Diabetes is a disease that makes a poor impact on people's lives around the world. According to the World Health Organization (WHO), around 422 million adults live with diabetes in 2014 globally, up from 108 million in 1980. The rise in diabetes prevalence has a relation with the increasing number of people with the condition. Deaths due to diabetes were approximately 3.7 million in 2012, which included 1.5 million from the disease and 2.2 million from its complications [1].

There are two types of diabetes categorized as type 1 and type 2. Early diagnosis with diabetes can help prevent or minimize its complications. A systematic approach may prevent diabetes, its complications, and hence premature death [1]. Treatment can be ineffective if the disease is too severe.

The cause of each type is different. The root cause of type 1 diabetes is currently unidentifiable despite medical advancement. In other words, there are no cures. Therefore, predictions are made to create awareness and strengthen patients' alertness to the underlying problem. Type 2 diabetes stems from the body's lack of efficacy in the usage of insulin. Diabetes in any shape or form could be detrimental to one's health and quality of life. The probability of this circumstance happening is higher if the disease is left undiagnosed and untreated. The level of blood glucose is a key indicator in probing diabetes. For that reason, the prediction of blood glucose is important to carry out.

Various studies have been carried out to predict blood glucose with machine learning such as [2], [3], [4] including two studies using LSTM by [5] and [6]. Previous findings and this study focused on diabetes type 1, but the results are also applicable to other types of diabetes. The study of blood glucose prediction written by

[5] used the LSTM network, which consists of a singular LSTM layer, bi-directional LSTM layer, and several other fully connected layers. The latter study yielded better results compared to the ARIMA and SVR methods with Root Mean Square Error (RMSE) at the value of 21.747. Research for blood glucose prediction written by [6] states that LSTM-NN is superior with an RMSE value of 12.38 compared to autoregressive and LSTM.

The Long Short-Term Memory (LSTM) of the RNN can be used to predict blood glucose. This method is often used to predict other problems [6]. The memory structure of LSTM can capture and store complex data patterns. LSTM enables the neural network to perform tasks that were previously impossible [7]. Although LSTM has good performances in solving time series problems, it faces a problem where it can't access the content of its previous memory cell when its output gate is closed [8]. Convolutional LSTM (conv-LSTM) is a variation of LSTM that can solve the problem [9]. Conv-LSTM is a gate-based memory system that has a connection between the previous memory content and the gate. It allows the gate to access the previous memory content.

The performance of conv-LSTM outperforms the traditional LSTM, CNN-LSTM, and CNN in previous research [9]. In research studies [18], conv-LSTM has had good results compared to other methods in in diabetes classification problems. Therefore conv-LSTM was chosen and applied to predict blood glucose in the continuous glucose monitoring (CGM) time series data in this study.

2. Methods

a. LSTM Network

Long short-term memory (LSTM) is a form of recurrent neural network (RNN) that stores long-term information. It was developed by Johannes Hochreiter and Martin Schmidhuber in 1997 [10]. LSTM was designed to answer the problem of long-term dependency on RNN [11].

The memory cell is the main component of LSTM. It coordinates with the three gate units, among them are the input gate, output gate and forget gate. The units can be used for various operations such as memory relay, write, and reset [7].

The output gate can block the output from the memory cell and sigmoidal nonlinearity exists in all gates [7]. Since the state unit can act as an additional input to other gating units, the LSTM architecture can easily solve the long-term dependency issues. [12]. Due to the blockage at the output gate, LSTM cannot access previous memory cell contents [13].

b. Convolutional LSTM Network

Convolutional LSTM allows the gate to take advantage of the previously stored content memory even after the output gate is blocked. The extra connection between each gate and the previous memory content can be added to solve the problem [9].

Conv-LSTM also has gates like LSTM, input gate, forget gate, and output gate. The illustration of conv-LSTM is shown in Figure 1.

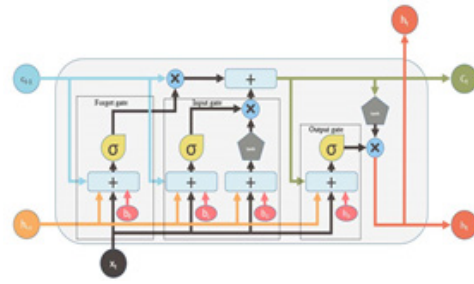


Figure 1 Conv-LSTM Structure [9]

The conv-LSTM implementation takes the previous memory content of the gate as input, which makes the previous memory cells usable even when the output gate is closed. This connection ensures the earliest input impact even when the input sequence is long. The implementation conv-LSTM is as follows [9]:

$$f_t = \sigma_g(w_f x_t + U_f h_{t-1} + u_f c_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma_g(w_i x_t + u_i h_{t-1} + u_i c_{t-1} + b_i) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \sigma_h(w_c x_t + u_c h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma_g(w_o x_t + u_o h_{t-1} + u_o c_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \tanh(c_t) \quad (5)$$

Here, the symbols σ_g and σ_h denote the sigmoid function and the hyperbolic tangent function respectively while b is bias, w , u , and u are the weight values used to avoid the gradient loss problem.

c. Prediction Model

Prediction models are built with Keras. Cross-validation is used with multiple data split consist of 80:20 and 70:30. The model is using a sequential model with one conv-LSTM layer. Equations (1), (2), (3), (4), (5) have already been built inside it. The conv-LSTM layer consisted of 256 filters with a reshape of (1, 1, 1, 1). The shaping is required for convolutional because the input needs to be 5-dimensional. The kernel in the conv-LSTM layer is using 1 width and 1 height to specify the height and width of the convolutional window. The conv-LSTM layer also uses reLU for its activation function for the recurrent step. The first layer is followed by a flattening layer to connect into one dense layer with an 8 unit and reLU activation function. Lastly, there is a one-unit dense layer with Adam optimizer as output layer to predicted blood glucose value. Model is trained with 100 epochs with 128 batch sizes.

d. Flowchart Design

After loading the time series dataset of continuous glucose monitoring (CGM), we need to choose a random patient and set the modelCount to 0. The modelCount is used to choose the best model from the training session for each patient. Whenever the modelCount is lower than 5, we will go to the next step of the training process.

First, we need to construct a training and test set with data split cross-validation depending on the variation that is used right now. There are two cross-validations in this study and both are using data split, it is 70:30 and 80:20 data split. Second, we will construct the conv-LSTM model as described in section 3. Next RMSE will be initialized with a big number and the training process will be started. After the model training is done, the model and the corresponding RMSE will be stored temporally to compare it with the initial RMSE or the previous RMSE model later on. If the new RMSE is greater than the previous one (or initial RMSE if it was the first run), the system will run the training process again. If the new RMSE is lower than the previous one, the current model and RMSE will be stored and the modelCount will be increased by one. After the modelCount is not lower than 5, the system will be outputting the best model for the patient. The process will be repeated until four models of a different patient are acquired for each cross-validation variation. For better understanding, the flow chart for this study algorithm is shown in Figure 2.

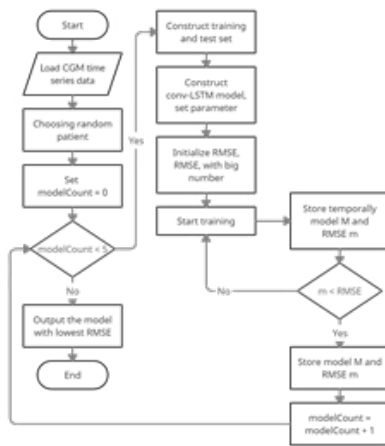


Figure 2 System Flowchart

e. Datasets

This study used the SENCE public dataset published in 2017 by the JAEB Center for Health Research. The dataset can be accessed in [14]. The data that is used from the SENCE dataset is the CGM. It was a data of glucose per 5 minutes for approximately 6 months to one year from a couple of patients. The data will be processed first into the desired shape as needed for the conv-LSTM layer in the prediction model. Only four patients out of all the patients will be used in this study. The dataset sample is shown in Table 1.

There are 4 columns in the dataset, RecID is a record id for each data, PtID indicates which patient has data on that row, DeviceDtTm has information about the date and time the data is recorded and the value is the blood glucose that got recorded.

Table 1 Dataset Sample

| RecID | PtID | DeviceDtTm | Value |
|----------|------|---------------------|-------|
| 5946141 | 38 | 2001-06-19 01:41:08 | 321.0 |
| 5946142 | 38 | 2001-06-19 01:46:08 | 319.0 |
| ... | .. | .. | .. |
| 8225193 | 38 | 2001-12-21 21:34:46 | 70.0 |
| 82251945 | 38 | 2001-12-21 21:39:46 | 73.0 |

f. Evaluation Methods

Two evaluation methods will be used in this research. We choose the two most commonly used ones, namely, root means square error and mean absolute error. Even more, root mean square error is usually used in blood glucose prediction research [15].

a. Root Mean Square Error (RMSE)

The variance of all errors rooted by the square is called the RMSE [16]. A given data set is computed by taking the difference between the target and the predicted data to obtain an absolute fit of a model. The relative measure of fit is called root squared, while the absolute measure of fit is called RMSE. It can be used to refer to the standard deviation of unexplained variance.

A better fit is represented by a lower RMSE value. RMSE is an evaluation method to know how good a model is at predicting the data. It's a good indicator of how accurate a model is at handling complex data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (6)$$

where y_i is the actual data and \hat{y}_i is the predicted glucose values.

b. Mean Absolute Error (MAE)

The mean absolute errors are known as MAE. It shows the gap between the actual value and the forecasted value. The value of the error from the forecast on average can be expected with MAE.

MAE is also the most popular error measure besides RMSE. MAE is given as:

where y_i is the actual target value for the test instance, \hat{y}_i while the predictive target value for the test instance is \hat{y}_i and the number of test instances is denoted by n .

$$\Sigma_{xx}(\theta) = \Lambda_x \Phi \Lambda_x' + \Theta_\delta, \quad (7)$$

3. Result

In this section, we explain two cross-validation variations investigated in this study. A model that automatically fits the samples in training data will get an overfitting result. Overfit is a condition in which the

model scores a perfect metric but fails to predict anything useful on the data that is not yet visible. Because of that, the use of cross-validation is necessary. We used two different ratios of time-series data split in this study, 70:30 and 80:20.

Many experiments use the two data split ratios of 70:30 and 80:20. The choice often gives pretty good results [5]. Another research conducted by [6] used a data split of 66:34, which is quite close to the data split used in our study.

In Figure 3 and Figure 4, the X-axis represents the timeline (date and time) whereas the Y-axis represents the blood glucose value. Three different data is shown by three different line colors. The blue line indicates the original data from the dataset. The green line is the training data from splitting a portion of the data from the original data. The last line, the red one, is the predicted outcome from the data test

Variation of cross-validation aims to find the best results from the training model. Based on [17] the bigger the training data, the better the model and the predictions, the better the results. However, the biggest size of training data does not mean it has the best results, so we vary the data to find better results.

a. Cross-Validation with Ratio 70:30

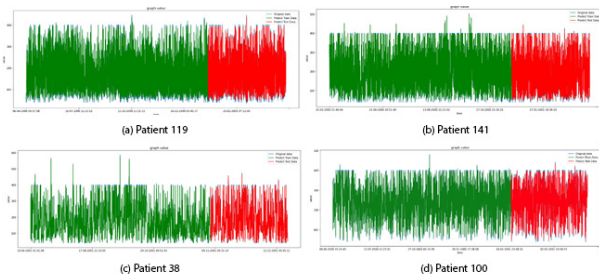


Figure 3 Result graph with 70:30 data split

This section shows the results of the cross-validation for four patients. The instrument that is used to derive results is cross-validation of data split 70:30. The purpose of cross-validation is to acquire more reliable results and to avoid the issue of overfitting.

We can see in Figure 3 that the prediction result or the red line is almost matched the original data. It means the prediction has a small error.

b. Cross-Validation with Ratio 80:20

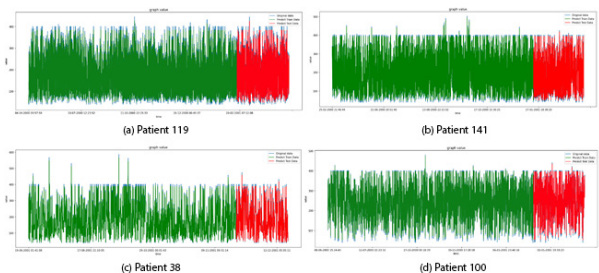


Figure 4 Result graph with 80:20 data split

This section shows the results of the cross-validation for the 80:20 data split of the data for four patients. The graph can be seen in Figure 4. We can compare them with the other cross-validation result, the prediction result or the red line are not covering up the blue line as much as in Figure 3.

From the results of the comparison, it can be seen that the error in the 80:20 cross-validation is greater. These results will be evaluated in the discussion section.

4. Discussion

The data in this study is a time-series data of blood glucose. Time series is well-ordered data so that the kernel size is 1-dimensional convolutional. After reshaping the dataset into 5-dimensional, the algorithm will project the input through the kernel. There is no direct relationship between the kernel size and the result accuracy for the time series because of the 1-dimensional size. Whereas filter parameter affects the accuracy of the model, selecting filter 256 increases the accuracy higher than filters 32, 64, and 128.

Table 2 RMSE and MAE Cross-Validation 70:30

| Evaluation Method | | Patient ID | | | |
|-------------------|-------|------------|-------|-------|-------|
| | | 119 | 141 | 38 | 100 |
| RMSE | Train | 21.44 | 23.99 | 22.62 | 21.48 |
| | Test | 21.26 | 23.11 | 21.58 | 22.98 |
| MAE | Train | 9.44 | 9.71 | 8.73 | 9.02 |
| | Test | 9.55 | 9.53 | 9.03 | 9.40 |

Using the RMSE and MAE evaluation methods from formula (6) and (7), the results in Table 2 and Table 3 were obtained in the scenario of cross-validation 70:30 and 80:20 respectively.

We can see from Table 2 and Table 3, calculation for patient 119 gets the best RMSE result from the 70:30 data split because it has the lowest RMSE value. Inversely, in terms of MAE, patient 38, patient 38 gets the best MAE result from the 70:30 split data because it has the lowest MAE value.

Table 3 RMSE and MAE Cross-Validation 80:20

| Evaluation Method | | Patient ID | | | |
|-------------------|-------|------------|-------|-------|-------|
| | | 119 | 141 | 38 | 100 |
| RMSE | Train | 21.61 | 23.94 | 22.68 | 21.70 |
| | Test | 22.06 | 23.05 | 21.97 | 22.75 |
| MAE | Train | 10.32 | 9.78 | 9.38 | 9.00 |
| | Test | 10.66 | 9.52 | 9.80 | 9.20 |

Apart from that, we can see that there is some variance in the individual errors from comparing the RMSE and MAE values of each patient. The greater the difference of the value, the greater the inconsistency of error. The differences in the value of our results are on average 11

to 13. That number is not that big but it is not that small either. So it is not possible to have a very large error in the results.

Overall although sometimes the 80:20 scenario gives better results for some results, the 70:30 scenario gives a more stable good result. Here we can see even though the 80:20 ratio of split data has larger training data sizes, the results are no better than the 70:30 ratio where the size of the training data used is smaller.

The result of RMSE in this study is acceptable when we recall the dependent variable because that is the deciding factor for the RMSE threshold. The RMSE in this study is comparable with the result in [5], but the result from [6] is still has a better result with 12.38 RMSE. The difference in the RMSE between the study is affected by the dataset and the model used.

Comparison with LSTM, conv-LSTM have comparatively good performances in solving time series problems. It also solves the problem of LSTM where it can't access the content of its previous memory cell when its output gate is closed. Although conv-LSTM is usually used for multiple-dimensional data, in the study conv-LSTM was used for one-dimensional cases and still yielded good results.

4. Conclusion

This paper describes the use of a deep neural network as a method of predicting blood glucose. The deep network has a sequential model that consists of one conv-LSTM layer, one flattened layer, one fully connected layer, and one last layer for the output.

The model has successfully attained a good result of RSME and MAE even with two different data split cross-validations. The 70:30 cross-validation gives a more stable result. Since the average range of RMSE and MAE is between 11 to 13, they are not significantly larger or smaller. Thus, the model can minimize errors in the prediction.

The result is affected by problems such as missing data from the patient. Missing data may occur, for example, because the patient takes off the CGM device that records the patient blood glucose.

Reference

- [1] WHO NCD Management-Screening, Diagnosis, and Treatment, "Global Report on Diabetes", 2016. Isbn, 9789241565257. Available online at: <https://www.who.int/publications/i/item/9789241565257>
- [2] Q. Wang, S. Harsh, P. Molenaar, and K. Freeman, "Developing personalized empirical models for Type-I diabetes: An extended Kalman filter approach," 2013. American Control Conference, Washington, DC, 2013, pp. 2923-2928, DOI: 10.1109/ACC.2013.6580278.
- [3] Mhaskar, Hrushikesh & Pereverzyev, Sergei & Van der Walt, Maria, "A Deep Learning Approach to Diabetic Blood Glucose Prediction," 2017. *Frontiers in Applied Mathematics and Statistics*. 3. 10.3389/fams.2017.00014.
- [4] Novara, Carlo & Pour, Nima & Vincent, Tyrone & Grassi, Giorgio, "A Nonlinear Blind Identification Approach to Modeling of Diabetic Patients" 2015. *IEEE Transactions on Control Systems Technology*. 19. 1-1. 10.1109/TCST.2015.2462734.
- [5] Q. Sun, M. V. Jankovic, L. Bally, and S. G. Mougiakakou, "Predicting Blood Glucose with an LSTM and Bi-LSTM Based Deep Neural Network," 2018. 14th Symposium on Neural Networks and Applications (NEUREL), Belgrade, 2018, pp. 1-5.
- [6] T. El. Idriss, A. Idri, I. Abnane, and Z. Bakkoury, "Predicting Blood Glucose using an LSTM Neural Network," 2019. *Federated Conference on Computer Science and Information Systems (FedCSIS)*, Leipzig, Germany, 2019, pp. 35-41.
- [7] Hochreiter, Sepp & Schmidhuber, Jürgen, Long Short-term Memory. *Neural computation*, 1997, 9. 1735-80.
- [8] Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*; MIT Press: Montreal, QC, Canada, 2015; pp. 802–810.
- [9] Rahman, Md. M.; Siddiqui, Fazlul H., "An Optimized Abstractive Text Summarization Model Using Peephole Convolutional LSTM", 2019 *Symmetry* 11, no. 10: 1290
- [10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] Chung, H., & Shin, K. S, "Genetic algorithm-optimized long short-term memory network for stock market prediction," 2018. *Sustainability*, 10(10), 3765
- [12] Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; pp. 373–418
- [13] Schmidhuber, J.; Hochreiter, S. Long short-term memory. *Neural Comput.* 1997, 9, 1735–1780.
- [14] Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* 1999, 12, 2451–2471.
- [12] Kim, Y.; Roh, J.H.; Kim, H. Early Forecasting of Rice Blast Disease Using Long Short-Term Memory Recurrent Neural Networks. *Sustainability* 2017, 10, 34.

- [13] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., Woo, W., “Convolutional LSTM network: a machine learning approach for precipitation nowcasting” 2015. *Adv. Neural Inf. Process. Syst.* 2015 (January), 802–810 Jun
- [14] JAEB Center for Health Research. Available online at: <https://public.jaeb.org/datasets/diabetes>
- [15] T. El Idrissi, A. Idri, and Z. Bakkoury, “Systematic map and review of predictive techniques in diabetes self-management”, *International Journal of Information Management*, 2019; vol. 46, pp. 263-277
- [16] E. Daskalaki, A. Prountzou, P. Diem, and S. G. Mougiakakou, “Real-Time Adaptive Models for the Personalized Prediction of Glycemic Profile in Type 1 Diabetes Patients,” *Diabetes Technol. Ther.*, vol. 14, no. 2, pp. 168–174, 2012.
- [17] Medar, Ramesh & Rajpurohit, Vijay & Rashmi, B. (2017). Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning. 1-6. 10.1109/ICCUBEA.2017.8463779.
- [18] Rahman, Motiur et al. “A deep learning approach based on convolutional LSTM for detecting diabetes.” *Computational biology and chemistry* vol. 88 (2020): 107329. doi:10.1016/j.combiolchem.2020.107329