

Deep Learning for Blood Glucose Prediction: CNN vs LSTM

Touria El Idrissi[1] and Ali Idri[2,3(✉)]

[1] Department of Computer Sciences EMI,
University Mohamed V, Rabat, Morocco
el.idrissit@gmail.com
[2] Software Project Management Research Team, RITC, ENSIAS,
University Mohamed V, Rabat, Morocco
ali.idri@um5.ac.ma
[3] CSEHS-MSDA, University Mohammed VI Polytechnic, Ben Guerir, Morocco

Abstract. To manage their disease, diabetic patients need to control the blood glucose level (BGL) by monitoring it and predicting its future values. This allows to avoid high or low BGL by taking recommended actions in advance. In this study, we propose a Convolutional Neural Network (CNN) for BGL prediction. This CNN is compared with Long-short-term memory (LSTM) model for both one-step and multi-steps prediction. The objectives of this work are: 1) Determining the best configuration of the proposed CNN, 2) Determining the best strategy of multi-steps forecasting (MSF) using the obtained CNN for a prediction horizon of 30 min, and 3) Comparing the CNN and LSTM models for one-step and multi-steps prediction. Toward the first objective, we conducted series of experiments through parameter selection. Then five MSF strategies are developed for the CNN to reach the second objective. Finally, for the third objective, comparisons between CNN and LSTM models are conducted and assessed by the Wilcoxon statistical test. All the experiments were conducted using 10 patients' datasets and the performance is evaluated through the Root Mean Square Error. The results show that the proposed CNN outperformed significantly the LSTM model for both one-step and multi-steps prediction and no MSF strategy outperforms the others for CNN.

Keywords: Convolutional Neural Network · Long-short-term memory network · Multi-step-ahead forecasting · Blood glucose · Prediction · Diabetes

1 Introduction

The diabetes mellitus disease occurs when the glucose metabolism is defected. Type 1 and Type 2 of diabetes mellitus (named respectively T1DM and T2DM) are the main types of diabetes. T1DM is due to a shortage in the insulin produced by the pancreas while T2DM is due to an inappropriate use of the produced insulin [1]. This chronic illness may cause serious health complications such as neuropathy, nephropathy, blindness and others [1]. Diabetic patients can prevent or delay the occurrence of these complications by managing their disease and maintaining their blood glucose level (BGL) within the normal range. This can be achieved by monitoring the BGL manually

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12250, pp. 379–393, 2020.
https://doi.org/10.1007/978-3-030-58802-1_28

via sticks or automatically via continuous glucose monitoring (CGM) sensors and then predicting the future values of BGL. If the predicted values tend to be outside the normal range, the diabetic patient can act in advance toward avoiding high or low BGL [1, 2].

Several data mining based prediction techniques were investigated for the BGL prediction problem counting machine learning and statistical techniques [2]. Nevertheless, machine learning and especially deep learning techniques are gaining more interest as they are achieving promising results [3].

The BGL prediction can be considered as a time series (TS) prediction problem where the past values are provided by a CGM device. The TS forecasting can be: 1) a one-step ahead forecasting (OSF) when the prediction concerns the next value or 2) a multi-steps ahead forecasting (MSF) when the prediction concerns the next H values where H is the prediction horizon [4].

In this work, we are interested in the application of deep learning techniques for BGL prediction especially LSTM and CNN in the context of OSF and MSF for 30 min ahead which is good enough to avoid likely increase or decrease of the BGL [5, 6].

In [5], the authors have conducted a comparative study between the five known MSF strategies using the LSTM technique. According to our belief, no similar study was done using the CNN technique and no comparison of the two techniques were performed in the OSF and MSF using the five known MSF strategies. These two points motivate this current study.

The CNN model that we propose in this study is a sequential one using a 1D convolutional layer, followed by a Flatten layer and 2 Dense layers. Regarding the LSTM model, we use the same architecture proposed by [3] as it outperformed significantly a previous LSTM model and an autoregressive model, this LSTM model contains one LSTM Layer tailed by 2 Dense layers.

The objective of the present work is threefold: 1) Getting a performant CNN based on the proposed architecture, 2) Determining the best strategy of MSF using the obtained CNN, and 3) Comparing the CNN and LSTM models for OSF and MSF.

Toward these objectives, we consider the following research questions (RQ):

- (RQ1): what is the best configuration of the proposed CNN?
- (RQ2): Is there an MSF strategy that outperforms the others using the proposed CNN?
- (RQ3): Is the proposed CNN model more accurate than the LSTM model in OSF?
- (RQ4): Is the proposed CNN model more accurate than the LSTM model in MSF?
- (RQ5): Is the performance for OSF maintained for MSF?

This paper is organized into 7 sections. Section 2 states the time series problem and gives an overview of the CNN and LSTM techniques. Section 3 highlights the related work. In Sect. 4, the experimental design is detailed. Section 5 reports and then discusses the results. Threats to validity are presented in Sect. 6 and finally, conclusion along with future works are presented in Sect. 7.

2 Background

In this section, we define the TS problem and identify the strategies used for MSF. Then, we present an overview of the two techniques used in this study: LSTM and CNN. And finally, we present the Wilcoxon test and ranking method.

2.1 Time Series Prediction

Let y_1 to y_N be the N past values of a time series. The TS prediction can be performed for: 1) a single period by determining the next value y_{N+1} which is called one-step ahead forecasting (OSF), or 2) multiple periods by determining y_{N+1} to y_{N+H} which correspond to the H next values, this is called multi-step ahead forecasting (MSF) [4]. The MSF problem is more difficult comparatively to the OSF one. In fact, the former is confronted to the accumulation of errors, the decreasing of accuracy and the increasing of uncertainty [4, 5].

To perform the MSF, five strategies can be used [4, 5]. These strategies are: 1) Recursive, 2) Direct, 3) MIMO (Multi-input Multi-output), 4) DirREC: a combination of Direct and Recursive; and 5) DirMO: a combination of Direct and MIMO. Table 1 presents details about these five MSF strategies.

2.2 LSTM and CNN: An Overview

LSTM and CNN are neural networks (NNs) with special architecture allowing a deep learning. This latter is an emerging technique that have the ability to learn the data characteristics and select relevant features automatically [7]. In this subsection, we start by defining the LSTM then the CNN architecture.

LSTM: Hochreiter and Schmidhuber in [8] came up with a novel architecture of recurrent NNs (RNNs), called LSTM NNs, in order to solve the problem of vanishing or exploding gradient met in the traditional RNNs. The LSTM NNs contains memory cells that have a cell state preserved over time and a gate structure for controlling and regulating the information through the memory cell. With this structure, the LSTM NNs can catch long term dependencies and treat serial data [3, 8].

Table 2 gives details about the memory structure. The following notations are used:

t: The time or sequence number.
X_t: The input vector for t.
Y_t: The output vector for t.
h_t: The hidden vector for t.
C_t: The cell state for t.
W_i, W_f, W_o and W_c: The weight matrices corresponding to each component.
b_i, b_f, b_o and b_c: The bias vectors corresponding to each component.
i_t, f_t, and o_t: The results of the input, forget and output gates respectively.

Note that σ and tanh are respectively the sigmoid and the hyperbolic tangent used as activation functions.

Table 1. MSF strategies.

Strategy	Description	Number of models	Characteristics
Recursive (or iterative)	The prediction is performed iteratively using a OSF model. Each predicted value is used as part of input values to predict the next one	One model with single output	Intuitive and simple. Risk of errors' accumulation
Direct (or independent)	The prediction for each step is performed independently from the others	H models: a model with single output for each step	No errors' accumulation. Dependencies between the estimated values may not be apprehended.
MIMO	The prediction is performed by one model that returns the predicted values in a vector	One model with multiple output	The stochastic dependencies between the predicted values are preserved No prediction flexibility
DirRec	For each step, the prediction is done by a corresponding model based on the past values and the predictions of previous steps	H models with single output	Take advantages from the Recursive and Direct strategies
DirMO	The prediction horizon is divided in B blocks with the same size; each block is predicted based on a MIMO model	B models with multiple output	Take advantages from the Direct and MIMO strategies

Table 2. LSTM memory structure.

Component	Role	Equations
Input gate	Getting the information to be retained	$i_t = \sigma(W_i * [h_{t-1}, X_t] + b_i)$
Forget gate	Getting the information to be ignored	$f_t = \sigma\left(W_f * [h_{t-1}, X_t] + b_f\right)$
Output gate	Calculating the output and updating the hidden vector	$o_t = \sigma(W_o * [h_{t-1}, X_t] + b_o)$ $h_t = o_t * tanh(C_t)$
Cell state	Maintaining the information through cells	$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$ Where $\tilde{C}_t = tanh(W_c * [h_{t-1}, X_t] + b_c)$

CNN: The origins of the CNNs go back to the neocognitron model proposed in [9]. They are based on the concept developed in [10] related to the simple and complex cells which were inspired from the animal visual cortex. However, the first successful CNN that was trained by backpropagation was proposed by [11].

A CNN is a feed-forward NN whose main layer is a convolutional one performing a convolution operation. This latter consists of applying and sliding a filter over the input data through an elementwise multiplication [12–14]. The connection weights represent the kernel of the convolution.

The dimension of the filter depends on the type of input data. In fact, 1D is used for sequential data such as text and time series, 2D is used for images and 3D for videos [7]. Multiple filters can be used to be able to extract more useful features [13].

In the case of time series, the convolution is applied based on the following formula:

$$C_t = f(\omega * X_{t:t+l-1} + b) \, \forall t \in [1, T - l + 1] \tag{1}$$

where C_t is the t^{th} element of C which is the vector resulting from the convolution, X is the time series with length T, ω is a 1D filter with length l, b is the bias parameter and f represents the activation function [13].

2.3 Statistical Test and Ranking

The Wilcoxon statistical test is a non-parametric test used to assess if the difference between the performances of two models is significant. This test is performed considering the null hypothesis (NH) that there is no difference between the compared models. The p-value of the considered NH is calculated, if the p-value is less than a certain significance level α, the difference is considered statistically significant [15].

In the case we need to have a ranking of the models, the sum of ranking differences (SRD) is used. It calculates the ranking of the models by summing up the difference of their ranking and an ideal ranking for a certain number of cases. The ideal ranking can be a reference model or the best known model. If such a model does not exist, the ideal ranking can be defined based on the minimum, the maximum or the average of all the models for each case [16].

3 Related Work

Statistical methods and especially autoregressive models were widely used for the BGL prediction. However, a growing trend has been noticed for the use of the machine learning techniques including deep learning [2, 17]. These latter were successfully used in many fields such as image recognition, object detection, sequential data processing, their success is due to their ability to learn automatically the data representation from raw data and extract the relevant features [7].

In the context of BGL prediction, deep learning techniques especially LSTM and CNN were investigated and encouraging results were reached [5]. In fact, the LSTM

was used in [3, 17–19] and CNN used in [17, 20]. In [14] CNN and LSTM were combined.

Regarding the strategies of MSF, Direct strategy is the most used one according to [5] and to the best of our knowledge, no study focused on the comparison of LSTM and CNN in the OSF and MSF taking in consideration all the MSF strategies which motivated the current study. However, in [17] LSTM and CNN were compared with other autoregressive models considering only Direct and Recursive strategies, the study did not conclude on the best technique in all the cases and pointed that Direct strategy for LSTM outperformed the Recursive one. In [5], the authors conducted an exhaustive comparison of the five MSF strategies using the LSTM and concluded that there is no significant outperformance of a strategy over the others and noticed that non-recursive strategies tend to be better than recursive ones.

4 Experimental Process

In the first part of this section, the dataset and the performance measurement used for experimentation and evaluation are described. Later, the experimental design adopted for this study is presented.

4.1 Dataset and Performance Measurement

In the experiments, we considered Ten T1DM patients whose data were extracted from the dataset DirecNetInpatientAccuracyStudy available at the site [21]. This dataset was used by [3, 5] and contains the measurements of BGL every 5 min using a CGM device. Note that these patients are taken randomly and a pre-processing of data was required to remove redundant records and outliers between successive records. The dataset of the Ten patients is described in Table 3.

Table 3. Ten patients' data description. The BGL is in mg/dl.

Patients	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Number of records	766	278	283	923	562	771	897	546	831	246
Min BGL	40	57	103	40	50	62	42	43	40	72
Max BGL	339	283	322	400	270	400	400	310	400	189
Mean BGL	114.78	120.96	185.89	188.44	179.71	187.45	210.26	152.88	157.50	116.51

In order to assess the performance of the considered models, we consider the frequently used performance measurement RMSE (root-mean-square error) [2] which is calculated as follows:

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

Where y_i and $\hat{y}_i$ represent the measured and the estimated value respectively and n represents the size of the considered sample. The RMSE value ranges in the interval [0, +∞[and the performance is higher when the RMSE value tends to 0.

4.2 Experimental Design

The empirical evaluation was conducted by considering the following steps: 1) Construction of the best configuration of the CNN model, 2) Development of MSF strategies for the CNN model and 3) Comparison between the CNN model and the LSTM model.

Step 1: Construction of the best configuration of the CNN model. The CNN model that we propose for the BGL prediction is a sequential one starting with a 1D convolutional layer, followed by a Flatten layer and 2 Dense layers. To determine the best configuration using this architecture, we consider a Search Grid (SG) concerning two hyper-parameters which are the kernel size and the number of filters. This SG is inspired from [3, 22] and presented in Table 4. This step is composed of 3 sub-steps which are the following:

Table 4. Search Grid parameters.

Parameter	Signification	Search space
Number of filters	Number of sliding windows	{2, 5, 10, 15, 20, 25}
Kernel size	Dimension of the sliding windows	{2,3,4,5,10}

- Step 1.1: It concerns the data preparation, in fact, the time series is decomposed into couples (X_i,y_i) where X_i is a vector of the input values and y_i is the output value which corresponds the next value following the values of the vector X_i.
- Step 1.2: For each value of the Number of Filters in the SG (Table 4) and each patient, the CNN model is trained and tested. Based on the SRD method, the best value of the Number of Filters is fixed.
- Step 1.3: For each value of the Kernel size in the SG (Table 4) and each patient, the CNN model with the Number of Filters fixed in Step 1.2 is trained and tested. Based on the SRD method, the best value of the Kernel size is fixed.

Step 2: Development of MSF strategies for the CNN model. This step contains 2 sub-steps which are the followings:

- Step 2.1: It concerns the data preparation. Let X = {s(t_i)} be a time series where s(t_i) is the BGL recorded at time t_i and d the sampling horizon, the time series X is divided to couples (X_i, y_i) where X_i and y_i are the input and the output data

respectively. Table 5 presents the decomposition done for each MSF strategy for 30 min ahead prediction.

Table 5. Data preparation for MSF strategies. HP: Horizon of Prediction; mn: minutes.

MSF strategy	Context	Decomposition
Recursive	HP = 5mn	$X_i = \{s(t_{i-d+1}),\ldots, s(t_i)\}$ $y_i = s(t_{i+1})$
Direct	HP = 30mn	$X_i = \{s(t_{i-d+1}),\ldots, s(t_i)\}$ $y_i = s(t_{i+6})$
MIMO	Multiple output for HP = 30mn	$X_i = \{s(t_{i-d+1}),\ldots, s(t_i)\}$ $y_i = \{s(t_{i+1}),\ldots, s(t_{i+6})\}$
DirRec	6 models M_j with HP = 5mn, j goes from 1 to 6	For each M_j: $X_i = \{s(t_{i-d-j+2}),\ldots, s(t_i)\}$ $y_i = s(t_{i+1})$
DirMO	Number of blocks = 2 So, 2 models are trained	For M_1: $X_i = \{s(t_{i-d+1}),\ldots, s(t_i)\}$ $y_i = \{s(t_{i+1}),\ldots, s(t_{i+3})\}$ For M_2: $X_i = \{s(t_{i-d+1}),\ldots, s(t_i)\}$ $y_i = \{s(t_{i+4}),\ldots, s(t_{i+6})\}$.

- Step 2.2: The performance is evaluated for each patient and each strategy using the RMSE. The models are trained on the training data which represents 66% of the dataset and evaluated on the test data which represents 34% of the dataset. If a difference between the performances is noticed, the statistical tests are applied to assess statistically the observed differences.

Step 3: Comparison between the CNN model and the LSTM model. The comparison is performed between our CNN model and the LSTM model proposed by [3] and used in [5]. The LSTM was developed for OSF and for the five MSF strategies using the same steps as for the CNN model namely data preparation and performance evaluation.

At this step, we compare first the performance of CNN and LSTM for OSF. Then, considering the MSF, we have to make 5 comparisons, in each one, the performances for each strategy with CNN and LSTM are compared. If a difference is noticed, it is assessed statistically by using the statistical tests.

5 Results and Discussion

In this section, we present the results of each step defined in the experimental design. Thereafter, we discuss the empirical results.

For the sake of the experimentations, we developed a tool under Windows 10 using Python-3.6 and the framework Keras 2.2.4 with, as backend, Tensorflow 1.12.0.

5.1 Results

This section presents the empirical results according to each step.

Step1: Construction of the best configuration of the CNN model. After fixing the architecture of our CNN model which is sequence of a convolutional layer (1D), a Flatten layer and 2 Dense layers, we prepared the data to fit the required input and output. Then, we carried out a set of experiments to answer the RQ1 by varying the two hyper-parameter: 1) the Filters' number in sub-step 1.2, and 2) the kernel size in sub-step 1.3.

Figure 1 shows the results of sub-step 1.2 which corresponds to the RMSE obtained for each patient and for each value of Filters' number from the Table 4. We can observe that with 20 Filters, we have the best RMSE. Furthermore, we used the SRD by considering the ideal ranking as the minimum performance. The results of the SRD are presented in Table 6, it shows that 20 Filters achieves the best ranking. At the end of this sub-step, we fixed the number of Filters to 20.

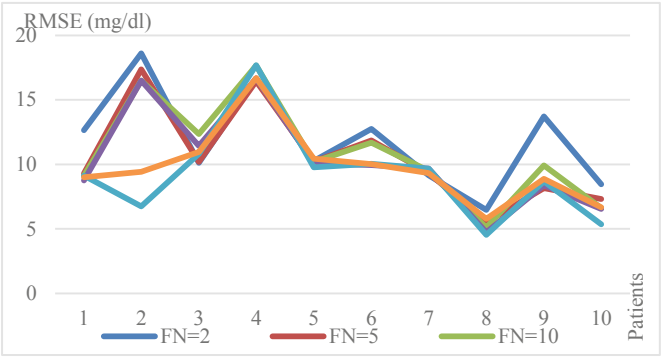


Fig. 1. Performance of CNN with Filters' number (FN) variations.

In sub-step 1.3, the same experiments were conducted by varying the Kernel size with respect to the values determined in Table 4. The results of these experiments are shown in Fig. 2. As there is no clearly dominant value, we use the SRD to rank the obtained models. Table 7 gives the ranking of the models based on SRD method. We can see that a Kernel size equals to 2 had the best ranking.

To summarize and answer the RQ1, 20 Filters and a kernel size equals to 2 give the best configuration of our CNN. This is the configuration of the CNN that will be used in the next steps of our experimental process.

Step 2: Development of MSF strategies for the CNN model. At that step and for each of the five MSF strategies, we started by preparing the data in accordance to the decomposition detailed in Table 5 then we train and validate the corresponding CNN

Table 6. SRD with the variation of Filters' number (FN).

Patients	FN = 2	FN = 5	FN = 10	FN = 15	FN = 20	FN = 25	Min
P1	5	4	3	0	2	1	1
P2	5	4	2	3	0	1	1
P3	0	1	5	4	2	3	1
P4	1	0	4	2	5	3	1
P5	4	3	2	1	0	5	1
P6	5	4	2	0	2	1	1
P7	0	2	3	5	4	1	1
P8	5	3	2	1	0	4	1
P9	5	0	4	1	2	3	1
P10	5	4	2	1	0	3	1
SRD	35	25	29	18	17	25	10

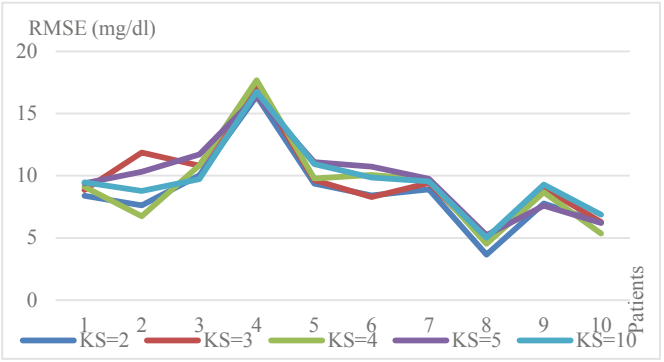


Fig. 2. Performance of CNN with Kernel size (KS) variations.

Table 7. SRD with the variation of Kernel Size (KS).

Patients	KS = 2	KS = 3	KS = 4	KS = 5	KS = 10	Min
P1	0	1	2	3	4	1
P2	1	4	0	3	2	1
P3	1	2	3	4	0	1
P4	1	3	4	0	2	1
P5	0	1	2	4	3	1
P6	1	0	3	4	2	1
P7	0	1	3	4	2	1
P8	0	2	1	4	3	1
P9	1	3	2	0	4	1
P10	2	3	0	1	4	1
SRD	7	20	20	27	26	10

model (s) based on the configuration that we obtained in the previous step. Figure 3 presents the performance in term of RMSE of the considered MSF strategies.

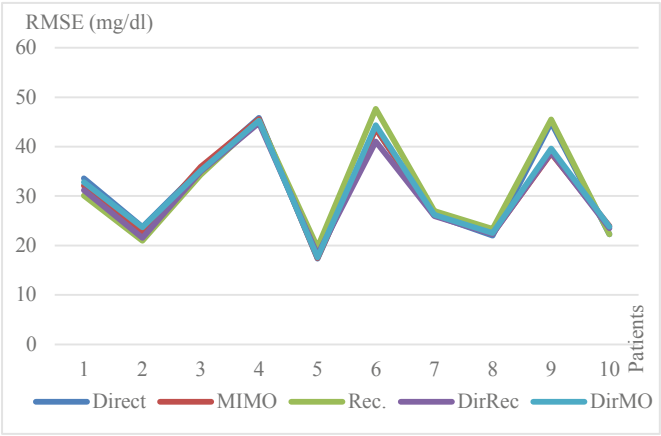


Fig. 3. Performance of CNN using the five strategies.

Figure 3 shows that there is no remarkable difference between the performances of the five strategies. Furthermore, considering the average for RMSE, the values are 31.48, 30.84, 31.60, 30.22 and 31.10 respectively for Direct, MIMO, Recursive, DirRec and DirMO. At that point, there is no need to perform a statistical test.

Thus, the answer for RQ2 is the following: there is no MSF strategy that outperforms the others using the proposed CNN. Besides, the five strategies are giving similar performances.

Step 3: Comparison between the CNN model and the LSTM model. To be able to compare the performances of the CNN and the LSTM models, we performed the experiments on the same datasets. We started by comparing the performance of the two models for OSF, and then the performances of the 30 min' forecasting using each of the MSF strategies are compared. The results are shown in Fig. 4. Figure 4.A represents the results for OSF for both CNN and LSTM, while figures Fig. 4.B, Fig. 4.C, Fig. 4.D, Fig. 4.E and Fig. 4.F represent the results for CNN and LSTM using Direct, MIMO, Recursive, DirRec and DirMO strategies respectively.

Figure 4 shows that CNN outperforms LSTM in OSF and MSF for all the strategies. To assess these findings statistically, we used the Wilcoxon test. Toward this, we formulate 6 NHs which are the followings:

- NH1: The CNN model does not outperform the LSTM model for OSF.
- NH2: The CNN model does not outperform the LSTM model for MSF using Direct strategy.
- NH3: The CNN model does not outperform the LSTM model for MSF using MIMO strategy.

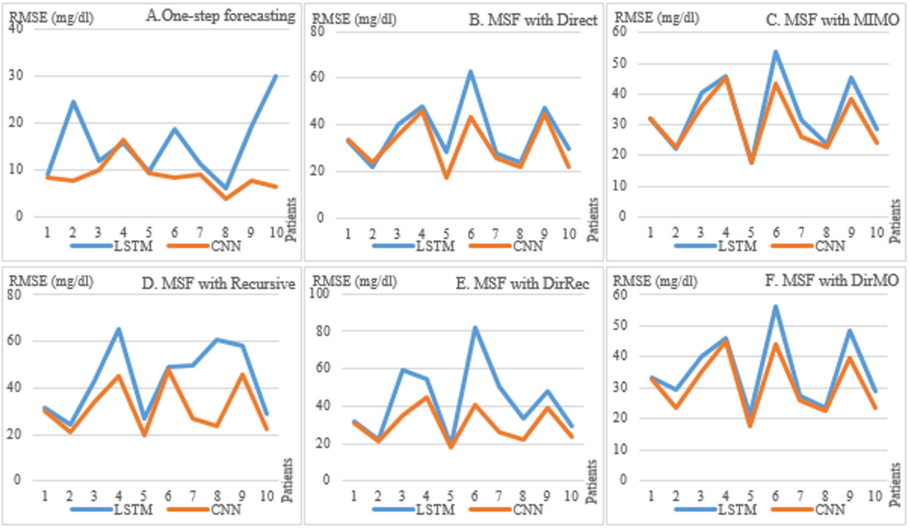


Fig. 4. CNN vs LSTM for one-step ahead forecasting (A) and multi-steps ahead forecasting (B to F).

- NH4: The CNN model does not outperform the LSTM model for MSF using Recursive strategy.
- NH5: The CNN model does not outperform the LSTM model for MSF using DirRec strategy.
- NH6: The CNN model does not outperform the LSTM model for MSF using DirMO strategy.

The statistical tests were Two-tailed with 0.05 as a significance level. The calculated p-values are presented in Table 8. As we can see, all the p-values are under the significance level, thus the CNN outperforms significantly the LSTM for OSF and MSF which answers the RQ3 and RQ4.

Table 8. p-value for the NHs.

NH	NH1	NH2	NH3	NH4	NH5	NH6
p-value	0.00932	0.02202	0.02852	0.00512	0.00512	0.00512

Concerning the RQ5, we remark from Fig. 4, that CNN is maintaining its performance over LSTM for OSF and all the MSF strategies.

5.2 Discussion

In the current study, we propose a CNN model composed of one 1D convolutional layer, one Flatten layer and 2 Dense layers. Our first objective was shaping the best

configuration of the suggested CNN model through the tuning of the two hyper-parameters: kernel size and the number of filters. Figures 1 and 2 show that each of these parameters has an influence on the obtained accuracy. Therefore, a careful choice of these parameters is crucial for building a CNN model and other models [5, 14]. Our CNN model performed well relatively to the performances found in literature and reported in [2]. Indeed, the minimum, maximum and mean RMSE for our CNN are respectively 3.66, 16.4 and 8.68. Furthermore, these results show that the CNN can perform well in time series prediction and sequential data in general even though it was traditionally conceived for image processing [7].

Concerning the prediction for the 30 min ahead, five strategies for MSF were developed and compared using the RMSE, the CNN model gives similar performances using these five strategies. These results are not consistent with the finding of the previous work using LSTM [5] where it was noticed that non-recursive strategies tend to be better than recursive ones even though there was no strategy significantly better than the others. Thus, further experiments should be carried out in order to refute or confirm this result for CNN. In the case of confirmation, the MSF strategy to consider should be Direct as it needs to train just one model with no iterations.

The third concern of this study is the comparison between CNN and LSTM. In fact, the performances of the CNN and the LSTM were compared in both the OSF and MSF, the results show that CNN outperforms significantly the LSTM in OSF and MSF using the five MSF strategies. This can be explained by the followings: 1) the CNN are using a small number of parameters since it uses shared weights [7], 2) There is no recurrent connections in the CNN contrarily to the LSTM which makes the process of training faster [14], and 3) The use of multiple filters in CNN helps to extract more useful features [13].

Finally, the performance of the CNN over the LSTM in OSF is maintained for MSF using the five strategies. This persistence is important as it gives some confidence in case we enlarge the prediction horizon.

6 Threats to Validity

We describe in this section the four threats to our study validity. These threats are:

- Internal validity: This threat to validity takes in consideration the evaluation approach. This latter should be appropriate so as the findings of the study are valid. Toward that aim, all the models were trained and tested in different datasets. In fact, 66% of each dataset was used in the training phase while the remaining 34% was used for the evaluation phase.
- External validity: This concerns the perimeter of the study and its ability to be generalized. To ensure this, we took in a random way the datasets of ten patients from a public dataset. These datasets have different sizes. In fact, the records' number varies from 246 to 923. Note that these datasets were previously used by [3, 5].
- Construct validity: The performance is the main criterion to compare the considered models, thus it is essential to use a performance measurement that indicates how far

the models are performant. In our study, we used the RMSE since it is a performance measurement used commonly in the BGL prediction [2].
- Statistical validity: In this study, we aim at determining the best model among the proposed ones through their performance's comparison. When a difference between two models is observed, it is essential to assess statistically this difference. Therefore, we used the Wilcoxon statistical test. Besides, the SRD method is used for ranking.

7 Conclusion and Future Work

This study proposed a CNN model having a 1D convolutional layer, a Flatten layer and two Dense layers. First we had to determine the best configuration by considering the two hyper-parameters: number of filters and kernel size. Then, we developed the MSF strategies using the CNN model to determine the strategy that may outperform the others. And finally, a performance comparison was conducted considering the CNN and LSTM models for one-step ahead forecasting and multi-steps ahead forecasting using the five identified MSF strategies.

The main outcomes of this work were: 1) No MSF strategy outperforms the others for CNN, besides they have similar performances, 2) The proposed CNN outperformed significantly the LSTM model for both one-step and multi-steps prediction.

These propitious results motivate further researches in the use of the CNN model taking in consideration different points such as: tuning other hyper-parameters, considering other input data such as activities and medication and exploring a larger prediction horizon.

References

1. Bilous, R., Donnelly, R.: Handbook of Diabetes. Wiley, Chichester (2010)
2. El Idrissi, T., Idri, A., Bakkoury, Z.: Systematic map and review of predictive techniques in diabetes self-management. Int. J. Inf. Manage. **46**, 263–277 (2019). https://doi.org/10.1016/j.ijinfomgt.2018.09.011
3. El Idrissi, T., Idri, A., Abnane, I., Bakkoury, Z.: Predicting blood glucose using an LSTM neural network. In: Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS, vol. 18, pp. 35–41. IEEE, Leipzig (2019)
4. Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Syst. Appl. **39**(8), 7067–7083 (2012)
5. El Idrissi, T., Idri, A., Kadi, I., Bakkoury, Z.: Strategies of multi-step-ahead forecasting for blood glucose level using LSTM neural networks: a comparative study. In: Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020), vol. 5, HEALTHINF, pp. 337–344. SCITEPRESS, Valletta (2020)
6. Fox, I., Ang, L., Jaiswal, M., Pop-Busui, R., Wiens, J.: Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1387–1395. ACM, London (2018)

7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Fukushima, K., Miyake, S.: Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recogn. **15**(6), 455–469 (1982)
10. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**(1), 106–154 (1962)
11. LeCun, Y., et al.: Handwritten digit recognition with a back-propagation network. In: Proceedings of Advances in Neural Information Processing Systems, pp. 396–404. MIT Press (1989)
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Min. Knowl. Discov. **33**(4), 917–963 (2019)
14. Li, K., Liu, C., Zhu, T., Herrero, P., Georgiou, P.: GluNet: a deep learning framework for accurate glucose forecasting. IEEE J. Biomed. Health Inform. **24**(2), 414–423 (2020)
15. Idri, A., Abnane, I., Abran, A.: Missing data techniques in analogy-based software development effort estimation. J. Syst. Softw. **117**, 595–611 (2016)
16. Héberger, K.: Sum of ranking differences compares methods or models fairly. TrAC Trends Anal. Chem. **29**(1), 101–109 (2010)
17. Xie, J., Wang, Q.: Benchmark machine learning approaches with classical time series approaches on the blood glucose level prediction challenge. CEUR Workshop Proc. **2148**, 97–102 (2018)
18. Sun, Q., Jankovic, M.V., Bally, L., Mougiakakou, S.G.: Predicting blood glucose with an LSTM and Bi-LSTM based deep neural network. In: 2018 14th Symposium on Neural Networks and Applications (NEUREL), pp. 1–5. IEEE, Belgrade, Serbia (2018)
19. Mirshekarian, S., Bunescu, R., Marling, C., Schwartz, F.: Using LSTMs to learn physiological models of blood glucose behavior. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2887–2891. IEEE, Seogwipo, South Korea (2017)
20. Zhu, T., Li, K., Herrero, P., Chen, J., Georgiou, P.: A deep learning algorithm for personalized blood glucose prediction. In: KHD@ IJCAI, pp. 64–78 (2018)
21. DirecNet: Diabetes Research in Children Network, http://direcnet.jaeb.org/Studies.aspx. Accessed 01 Apr 2019
22. Hosni, M., Idri, A., Abran, A.: Investigating heterogeneous ensembles with filter feature selection for software effort estimation. In: Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, pp. 207–220. ACM, Gothenburg, Sweden (2017)