A deep learning approach to adherence detection for type 2 diabetics

Conference Paper *in* Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference · July 2017

DOI: 10.1109/EMBC.2017.8037462

6 authors, including:

Ali Mohebbi
Technical University of Denmark
5 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Henrik Bengtsson
Novo Nordisk
5 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Marco Fraccaro
Technical University of Denmark
9 PUBLICATIONS **60** CITATIONS
SEE PROFILE

A Deep Learning Approach to Adherence Detection for Type 2 Diabetics

Ali Mohebbi[1], Tinna B. Aradóttir[1,2], Alexander R. Johansen[1],
Henrik Bengtsson[2], Marco Fraccaro[1], Morten Mørup[1]

***Abstract*— Diabetes has become one of the biggest health problems in the world. In this context, adherence to insulin treatment is essential in order to avoid life-threatening complications. In this pilot study, a novel adherence detection algorithm using *Deep Learning* (DL) approaches was developed for type 2 diabetes (T2D) patients, based on simulated Continuous Glucose Monitoring (CGM) signals. A large and diverse amount of CGM signals were simulated for T2D patients using a T2D adapted version of the Medtronic Virtual Patient (MVP) model for T1D. By using these signals, different classification algorithms were compared using a comprehensive grid search. We contrast a standard *logistic regression* baseline to *Multi-Layer Perceptrons* (MLPs) and *Convolutional Neural Networks* (CNNs). The best classification performance with an average accuracy of 77.5% was achieved with CNN. Hence, this indicates the potential of DL, when considering adherence detection systems for T2D patients.**

I. INTRODUCTION

Diabetes has become one of the biggest health problems in the world. It is categorized into two main types, type 1 diabetes (T1D) and the more common type 2 diabetes (T2D) with prevalence of approximately 95% [1]. Diabetes is a chronic condition where the body is incapable of producing enough insulin or where it cannot use it, causing elevated blood glucose (BG) levels. An estimated 415 millions adults are diagnosed with diabetes worldwide, while 193 millions are still undiagnosed. Prolonged elevated glucose levels can lead to life-threatening health complications [1]. Therefore, in T1D and progressed T2D, supplementation of long-acting insulin may be required to maintain the desired glucose levels. Hence, adherence to treatment and self-monitoring of blood glucose (SMBG) is essential for all diabetes patients. Unfortunately, adherence to insulin treatment is quite poor, as it is estimated to be only 57-67% and 47-51% in the UK and US respectively [2].

SMBG is the most widely used approach to manage BG levels, however improved Continuous Glucose Monitoring (CGM) technology has opened up exciting opportunities for T2D management. CGM devices are already in use in T1D and expansion of these towards treatment decisions is taking place [3], giving the ability of continuously monitoring BG levels with a predefined interval (usually between $5-15$ minutes).

In this pilot study, a large amount of simulated 24 hour T2D CGM signals were produced (with glucose level readings every 5 minutes). The simulation scenarios represented the different realities of T2D patients including adherence and non-adherence. The CGM signals were used to develop and evaluate different adherence detection models/algorithms. The purpose of this study was to investigate whether adherence detection was possible based on (simulated) CGM signals. In this context, we wanted to investigate if this goal could be achieved by simple linear or more complex approaches as in the case of Deep Learning (DL). Recently, DL methods have advanced and received state-of-the-art performance within Machine Learning (ML). This includes image and speech recognition problems, in addition to monitoring purposes as in e.g. Parkinson's Disease [4]–[8].

As a proof of concept, we investigated the possibility of building different classification models for adherence detection using CGM signals. The included models consist of logistic regression, and more complex models with Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs). Additionally, in order to achieve better predictive performances, *ensembling* was examined. In this context, Google's TensorFlow (r0.12) [9] was used as the platform to develop the algorithms, allowing straightforward configuration of the models [7]. A robust adherence detection algorithm is essential, considering effective insulin therapy, avoidance of hypo- and hyperglycaemia in addition to finding subject-specific optimal (insulin) doses. To the best of our knowledge, this study is the first of its kind.

II. METHODS

A. Simulating insulin-glucose dynamics in T2D

Kanderian et al. [10] proposed a physiological model for simulating 24 hour insulin-glucose dynamics in T1D patients along with identified parameter sets for 10 patients. The data simulator in the current work uses this model as a foundation for the T2D model.

The non-linear MVP model consists of 6 compartments; subcutaneous and plasma insulin (I_{sc} and I_p), two meal compartments (D_1 and D_2), insulin effect on glucose (I_{eff}) and blood glucose (G). A block diagram is illustrated in Fig. 1. *EGP* and *GEZI* represent endogenous glucose production and insulin effectiveness at zero insulin. The underlying coupled differential equations can be found in [10]. The simulator furthermore uses a linear model of endogenous insulin production by Ruan et al. [11] and adjustments in a number of parameters to account for main physiological differences between T1D and T2D patients. This method

[1]Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

[2]Novo Nordisk A/S, DK-2880 Bagsværd, Denmark. This project is funded by Innovation Fund Denmark.

978-1-5090-2809-2/17/$31.00 ©2017 IEEE

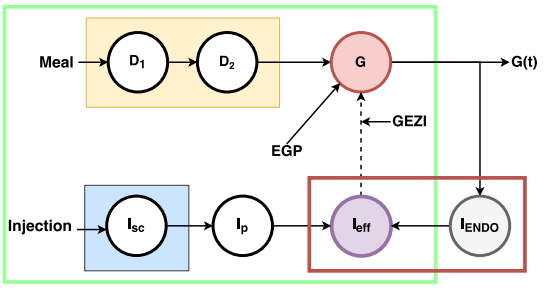

Fig. 1. Block diagram of the MVP model (green) [10] augmented with endogenous insulin production (red) [12] used for simulations in the current work. The output, $G(t)$, is the noisefree signal.

is described in [12]. The endogenous insulin production, defined by Eq. 1,

$$I_{ENDO}(t) = \frac{M_I(G(t) - G_{ENDO}) + M_0 G_{ENDO}}{MCR_I W}, \qquad (1)$$

is added to the plasma insulin compartment of the MVP model. The numerator expresses the posthepatic insulin secretion rate. This is illustrated in Fig. 1. The green square depicts the original MVP model for T1D patients, while the red square shows the T2D augmentation to the model. W is the body weight, MCR_I is the insulin metabolic clearance rate, G_{ENDO} is the fasting plasma glucose concentration threshold related to the endogenous insulin production, M_0 is the basal glucose sensitivity, M_I is the posthepatic glucose sensitivity and $G(t)$ is the plasma glucose concentration. The values for the parameters were chosen as the median values reported in [11] ($MCR_I, M_I = M_{I,f}, M_0 = M_{0,f}$). The G_{ENDO} threshold was set to 7 mmol/L.

B. Long-acting Insulin & Titration Algorithm

In this study, the pharmacokinetic profile of the long-acting insulin Tresiba®, or Insulin Degludec (IDeg) by Novo Nordisk A/S, is used. Based on clinical data, Tresiba has a half-life of > 25 hours, reaches steady state after 2-3 days and is detectable in the blood for approximately 5 days [13]. The MVP model parameters, published in [10] correspond to fast acting insulin infused by pumps. Since the current study considers long-acting insulin injections, a long-acting insulin injection profile is simulated by fast-acting insulin infusion which gives a similar activity profile. This is described in [12].

To find patient-specific optimal dose, the simulator uses a titration algorithm previously used in clinical trials on IDeg. The blood glucose concentration target is set to $4-5$ mmol/L, which ensures that an identified optimal dose at end of titration safely keeps glucose concentrations within desired levels. In the data set, a patient is either adherent or non-adherent to the prescribed optimal dose.

C. CGM Simulation & Data Generation

To simulate CGM noise, an autoregressive noise model proposed by Facchinetti et al. [14] was added on top of the simulation model output ($G(t)$ in Fig. 1). We wanted to simulate scenarios for T2D patients, which would imitate the reality of their BG levels in many different cases. In this context, considering the long half-life of IDeg, the simulation days were of 10 consecutive days. The first 5 days with the patient being adherent to the optimal insulin dose, followed by 5 days of non-adherence. The implementation of the titration algorithm and simulations of CGM signals were performed in MATLAB (The MathWorks, Inc.).

In order to produce a large amount of CGM data, parameters of 9 out of the 10 identified parameter sets in [10] were used. Subject number 10 was excluded due to unexplained unrealistic dynamics. A subset of parameters were varied within a pre-defined range, defined by typical differences between T1D and T2D patients seen in clinical trials [12], [13]. These include 10%, 20% and 30% increase in the body weight (W), basal glucose level (G_b) between $6-11$ mmol/L and a decrease in insulin sensitivity (S_I) by 30%, 50% and 70%. These adjustments are described in [12]. To the best of our knowledge, there is lack of evidence on correlation between the described parameters. Thus the change in a subset of parameters.

Each day of CGM data includes either 2 or 3 main meals with varying carbohydrate (CHO) intakes and several snacks. Sizes and timings of CHO intake were chosen at random within predefined realistic ranges [15]. This resulted in a large amount of CGM data for the 9 patients, representing different realities of T2D patients. Based on the variation in parameters in addition to repetition of the simulations, 10800 days of CGM data ($3W \times 3S_I \times 6G_b \times 10$ days $\times$ 20 different meal scenarios) were produced for each patient, resulting in a total of 97200 simulated CGM days. Simulated CGM data produced for patient 1-6 were used for training purposes of the models, whereas the data for the remaining 3 patients were reserved for evaluation of the developed models, see also Fig. 2.

D. Deep Learning

Deep learning builds on neural network theory, and performs automatic *feature extraction* rather than relying on expert-dependent features [7]. Here, the designed models are characterized by producing outputs which are the class probabilities of either adherence and non-adherence. In other words, the probability of belonging to each of the classes given a day of CGM data as input. As function for training error, we use the *cross-entropy error function* defined by,

$$E = -\sum_{n=1}^{N} \Big(y_n \; ln \; \hat{y}_n + \big(1 - y_n\big) \; ln \; \big(1 - \hat{y}_n\big) \Big), \qquad (2)$$

where y_n is the the desired output and $\hat{y}_n$ is the predicted class probability for a day of CGM (denoted as $\mathbf{x}_n$, $n = 1,..,N$). The predicted class probability is defined with a *logistic sigmoid activation function,*

$$\hat{y}_n = \hat{y}(z_n) = \frac{1}{1 + e^{-z_n}} \; . \qquad (3)$$

For logistic regression z_n is a linear parametric function, i.e. $z_n = \sum_i w_i x_{ni} + b$, whereas for MLPs and CNNs we make the model more powerful by adding multiple layers of non-linear

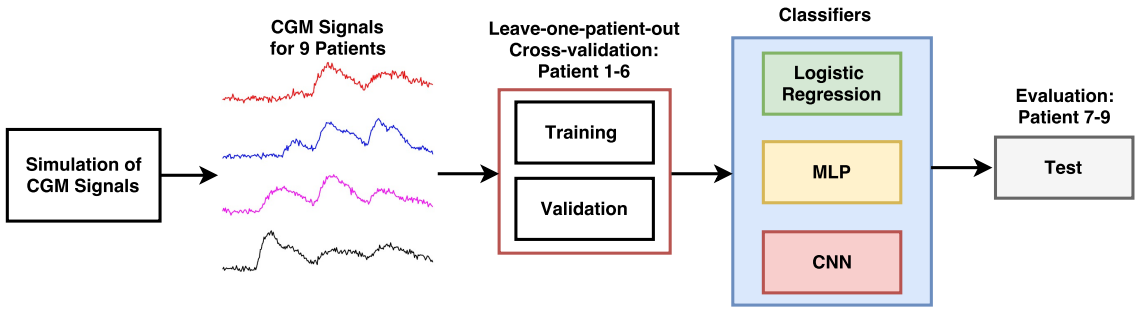

Fig. 2. Schematic representation of the simulations followed by classification. Once the CGM signals were produced for all 9 patients, they were divided into training, validation and test set. Based on leave-one-patient-out cross-validation the training and validation sets, including patient 1 to 6, were used on one of the classifiers. Best performing classifiers were then applied on the test set (patient 7-9), presented to the models for the first time, resulting in different performances.

parametric functions.

The optimal values for the parameters of the model, represented for logistic regression by the weights w_i and the bias term b, are found by minimizing the cross-entropy error function. While this function is convex for linear regression, for MLPs and CNNs the non-linearities in the model lead to a *non-convex* problem with several local minima. We find the minimum with gradient-based optimization, using the *Adam* optimizer [16] for stochastic error functions. The framework used to develop the different classification algorithms is based on the state-of-the-art library Tensorflow (r0.12) [9] developed by Google, which has gained considerable attention since its release in the fall of 2015.

E. Grid Search & Tuning of Hyper-parameters

As mentioned earlier, DL has shown great performance in different classification problems. However, the performance is highly dependent on the problem, the amount of available training samples and the architecture of the models. Additionally, it relies on a set of *hyper-parameters*, which have to be thoroughly chosen. Among these we find for example the number of iterations/epochs, the mini-batch size (BS) and the learning rate (LR) [17]. Incorrect tuning of these can cause suboptimal performance. In general, DL is a highly empirical and explorative process when investigating new data and model architectures. We therefore performed a *grid search* for each model in order to tune the hyper-parameters in a more principled way.

F. Experiments & Choice of Hyper-parameters

The experimental pipeline is presented in Fig. 2. For the simulated CGM signals of each patient, the data was separated into a training, validation and test set. In this context, *leave-on-patient-out cross-validation* was performed when evaluating the classifier models of logistic regression, MLP and CNN under training. Here, patient 1 to 6 were used for training purposes. The average cross-validation performances were used to tune the hyper-parameters when training the models. Once the best models were found, based on a comprehensive grid search, each model was evaluated on the test set including patient 7 to 9. In order to investigate the robustness of the models to the initialization of the parameters, 10 additional restarts were performed, giving a total of 11 runs.

In the case of logistic regression, the LR was set to either $10^{-2} - 10^{-4}$ with BS 150 and 250. The best performance was obtained with a BS of 250 and LR = 10^{-4}. Henceforth, this BS was used for MLP and CNN.

Under MLP, a larger grid search was performed by including LR between $10^{-2} - 10^{-5}$, 3 different numbers of *fully-connected* hidden layers (1, 2 or 3) of either 10, 50 and 100 hidden units. The output in the hidden layers used *rectified linear units* (ReLUs) as activation function, as commonly used in different DL tasks [8], as it does not suffer from vanishing gradients like the sigmoid or tanh activation function,

$$f_{ReLU} = max(0,x) \ . \tag{4}$$

Additionally, several combinations of layers with different numbers of hidden units were performed for larger models with and without *dropout* of different probabilities (0.7 and 0.9). Dropout is a *regularization* method where you randomly turn off some of the units in the layers to control and avoid *overfitting* to the training data [18].

Considering CNN, 3 different convolutional layers with ReLUs, followed by *max-pooling* layers were applied [4]. The LR range was restricted between $10^{-3} - 10^{-4}$ based on performances from previous models. Hyper-parameters related only to CNN include *filter size* (in both convolutional and max-pooling layer), *number of filters* and *stride length*. In our case, stride length and filter size of the max-pooling layers were fixed to 2 and $[1 \times 2]$, respectively. Given this setup, the number of hidden units in the fully-connected layer before the output layer were investigated with 10 and 50 units. For the feature extraction, $8, 16, 32$ and 64 number of filters and filter sizes of $6, 12$ and 18 were investigated. The filter size represents the part of the signals which is taken into account at a time. Thus, e.g. a filter size of 6 corresponds to 30 minutes of the simulated CGM signal. Each filter searches for a specific feature of the whole sequence.

III. Results & Discussion

The average performances of the best models are shown in Fig. 3. In MLP the combination of LR = 10^{-3} and one

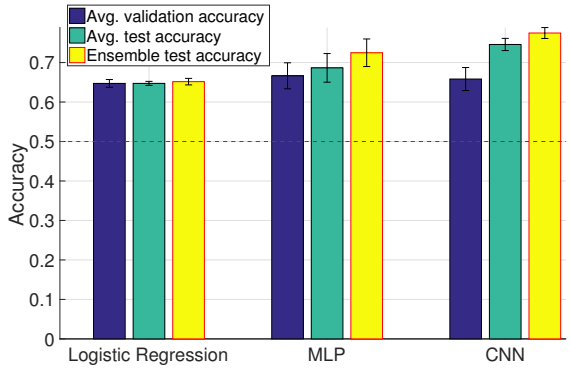

Fig. 3. Accuracy performances of the best models, including average validation, average test and ensemble test accuracy. The standard deviations across the 11 restarts are indicated with error bars.

hidden layer with 10 hidden units was the best performing model. In CNN, the best model was the combination of LR = 10^{-4}, 2 convolutional layers, FS = 18, number of filters = 8 and 10 units in the fully-connected layer. The best performing models did not include dropout. We show average validation, test and ensemble test accuracies (including error bars), based on the 11 runs. Ensembling is obtained by taking an average of the probabilities given by different models, before making a prediction. We ensemble the output probabilities given by the 6 models obtained during the cross-validation step. On the other hand, to compute the average test accuracy we average, instead of the probabilities, the accuracies obtained by each of the 6 models.

It is easily observed that for all 3 models, adherence detection is higher than 50% chance (dashed red line). Interestingly, the average validation accuracy is similar for all 3 models. On the other hand, the average and ensemble test accuracies seem to perform much better. An explanation of this could be the fact that the patients in the training data were too difficult to train, compared to the patient data in the test set. This indicates and necessitates individual examination of each patient, when it comes to real world application of such adherence detection system. Additionally, a remarkable point here is the obvious effect of ensembling, when taking the average of the class probabilities before performing predictions. In all 3 cases, the ensemble test accuracy does better compared to the average test accuracy. An overall comparison presents CNN being the best preforming model with $77.5 \pm 1.4\%$, followed by $72.5 \pm 3.5\%$ in MLP and $65.2 \pm 0.8\%$ in logistic regression.

It should be emphasized that DL seem to have great potential when it comes to adherence detection for T2D patients. Both MLP and CNN indicate promising performance, given the setup and grid search presented in this study. However, based on the results, there is an indication of the models performing differently on each patient. Hence, patient-specific classification algorithms should be further investigated using the different DL approaches.

IV. Conclusion

As the main objective of this study, we showed that it is possible to use CGM signals for adherence detection for T2D patients, in addition to DL being a promising approach for this task. A large amount of CGM signals were produced, imitating the reality of T2D patients. Here, the presented models were investigated by a comprehensive grid search, using the simulated CGM signals. Even though the standard logistic regression model achieved an accuracy better than random of $65.2 \pm 0.8\%$, the best performing models were obtained with DL, giving a $77.5 \pm 1.4\%$ accuracy with CNN, and $72.5 \pm 3.5\%$ with MLP. Considering future perspective, once large amount of real CGM data is available, the possibility of patient-specific detection systems, based on DL models, should be examined.

References

[1] Atlas, IDF Diabetes. "Brussels, Belgium: International Diabetes Federation; 2015. International Diabetes Federation."
[2] Davies, M. J., et al. "Realworld factors affecting adherence to insulin therapy in patients with Type 1 or Type 2 diabetes mellitus: a systematic review." Diabetic Medicine 30.5 (2013): 512-524.
[3] Castle, Jessica R., and Peter G. Jacobs. "Nonadjunctive use of continuous glucose monitoring for diabetes treatment decisions." Journal of diabetes science and technology 10.5 (2016): 1169-1173.
[4] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.
[5] He, Kaiming, et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." Proceedings of the IEEE international conference on computer vision. 2015.
[6] Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal Processing Magazine 29.6 (2012): 82-97.
[7] Eskofier, Bjoern M., et al. "Recent machine learning advancements in sensor-based mobility analysis: Deep learning for Parkinson's disease assessment." Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the. IEEE, 2016.
[8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," (White Paper, November 9, 2015), 2015.
[10] Kanderian, Sami S., et al. "Identification of intraday metabolic profiles during closed-loop glucose control in individuals with type 1 diabetes." Journal of Diabetes Science and Technology (2009): 1047-1057.
[11] Ruan, Yue, et al. "Modelling endogenous insulin concentration in type 2 diabetes during closed-loop insulin delivery." Biomedical engineering online 14.1 (2015): 19.
[12] Aradóttir et al. "A Model for Simulating Fasting Glucose in Type 2 Diabetes and the Effect of Adherence to Treatment." The 20th World Congress of the International Federation of Automatic Control (Accepted).
[13] Haahr, Hanne, and Tim Heise. "A review of the pharmacological properties of insulin degludec and their clinical relevance." Clinical pharmacokinetics 53.9 (2014): 787-800.
[14] Facchinetti, Andrea, et al. "Modeling the glucose sensor error." IEEE Transactions on Biomedical Engineering 61.3 (2014): 620-629.
[15] American Diabetes Association. "Nutrition recommendations and interventions for diabetes." Diabetes care 31.Supplement 1 (2008): S61-S78.
[16] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[17] Pandey, Gaurav, and Ambedkar Dukkipati. "To go deep or wide in learning?" AISTATS. 2014.
[18] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15.1 (2014): 1929-1958.