WAIT: Workshop on the Advances in Information Technology, Budapest, 2021

In-silico Simulation Based Evaluation of Insulin Prediction Method for Personalized Medical Treatment

Bálint Szabó[1], Ákos Szlávecz[1], Béla Paláncz[1], Geoffrey Chase[2], Balázs Benyó[1]

[1] Budapest University of Technology and Economics, Department of Control Engineering and Information Technology, Budapest, Hungary,
[2] Department of Mechanical Engineering, University of Canterbury, Christchurch, New Zealand

Abstract

Stress-induced hyperglycaemia is a frequent and serious issue in the intensive care causing elevated mortality rate. Insulin therapy is often applied in ICUs to normalize the patient's blood glucose level. This treatment method is generally referred to as Tight Glycaemic Control (TGC).

The most widely used TGC protocol is the STAR (Stochastic-TARgeted) protocol, which uses the patient's insulin sensitivity (SI) as a key parameter to describe the patient's actual state. STAR protocol uses the clinically validated ICING model to describe the human metabolic system and a stochastic model to predict the patient's future SI values.

In this paper, the evaluation of two new, artificial neural network based SI prediction methods is presented. The models were trained on a dataset collected during the STAR treatment. The models were evaluated by using a so-called in-silico validation, simulating the clinical interventions on virtual patients created from historical treatment data. The results proved that the new models could be applied in the SI prediction. The prediction accuracy was the same or even better in some aspects than the currently used model. The methods also support higher dimensional SI prediction, which is the field of recent research and resulted in improved personalized treatment based on the evaluation presented.

Categories and Subject Descriptors (according to ACM CCS): [Neural Networks, Simulation Evaluation]: machine learning, artificial intelligence, mixture density network, deep neural network, insulin sensitivity, tight glycaemic control, intensive care, STAR protocol, validation, in-silico validation

1. Introduction

Stress-induced hyperglycaemia is a frequent complication in the intensive therapy [1, 2]. The high absolute value and the high variability of the blood glucose level show significant correlation with the mortality rate in the intensive care.

Forcing the blood glucose (BG) level of these hyperglycaemic patients into the normal, so-called normoglycaemic range shows definite clinical benefits [3, 4, 5, 6]. This therapy is called in general as tight glycaemic control (TGC) that includes insulin therapy and occasionally moderation of the nutrition intake of the patient. In the intensive care there is tight control over these parameters. The insulin dosage is delivered in a form of infusion with well-known insulin effect characteristic in the main time. In some cases there is also bolus insulin dosage. Unlike the continuous dosage of the insulin, the bolus is an occasional correction. The nutrition intake happens in a parenteral or in an enteral way. These are standard methods in the intensive care.

Recently there are model-based TGC protocols for the glycaemic control that successfully implement safe and efficient patient treatment [7, 8, 9, 10, 11].

The STAR (Stochastic-TARgeted) TGC protocol is the most widely applied among them, it is used in four different countries [8]. STAR uses a clinically validated physiological model, called Intensive Control Insulin-Nutrition-Glucose (ICING) to describe the glucose-insulin dynamics, and a population-based stochastic model to manage patient-specific metabolic variability [12].

ICING is a pharmacokinetic-pharmacodynamic model [13] defining glucose-insulin kinetics and dynamics in the human body. Most of the parameters of the ICING model are chosen constant in the model calculation, except the key parameter

[14, 15], the insulin sensitivity (SI). This parameter is used to define the state of the patient. This parameter is not to be confused with the insulin sensitivity as the opposite of insulin resistance, which terminology is used in diabetology. The SI parameter of the model is influenced by multiple patient specific factor, including the diabetological SI too, so this parameter more like an abstract aggregate of the patient current state. Therefore this parameter can not be measured, it has to be identified by the protocol from the clinical treatment data (insulin dosing, nutrition intake and BG measurements) during the treatment.

STAR uses the patient-specific insulin sensitivity (SI) in the optimal treatment selection method. This method uses simulations with different treatment parameters and calculate the blood glucose (BG) levels at the end of these simulated treatments. From the results the protocol can recommend the optimal treatment parameters.

During the patients' STAR treatment the SI value, representing the current condition of the patient is identified in every hour. By processing the SI time series $\{SI(t);SI(t+1)\}$ data pairs can be created. These data pairs make up a distribution, which is called the 2D SI distribution. The logic behind the terminology is that the dimension refers to the dimension of the data points. In the 2D case $\{SI(t);SI(t+1)\}$ data pairs make up the distribution while in a 3D case for example $\{SI(t-1);SI(t);SI(t+1)\}$ triples can make it up. A prediction model is called higher dimensional if it works on data points with dimension higher than 2. This distribution can be transformed to a conditional distribution on the current SI value by kernel fitting [16, 17, 18]. Figure 1 shows the 2D SI distribution, the red curve is the conditional distribution of $SI(t+1)$ for a given $SI(t)$. Equivalently to the definition above, a model is called higher dimensional if it predicts a conditional distribution with more than one conditional variable. The conditional distribution is always one dimensional because all of the conditional parameters are set to a fixed value coming from the input parameters of the prediction. Fixing a conditional parameter reduces the dimension of the distribution, as it can be seen on Figure 1. By fixing $SI(t)$ to a concrete value the planar 2D distribution reduce into one dimensional that contains data points only with the fixed $SI(t)$ value. The currently used, stochastic model of the STAR protocol is based on kernel fitting [16]. In the STAR application supporting the implementation of the STAR protocol this means that there are calculated $SI(t+1)$ values at 5., 25., 50., 75., 95. percentile from the conditional distributions for some $SI(t)$ values at a defined step size.

The reason behind the percentile values is that the STAR protocol does not use the concrete predicted $SI(t+1)$ value but rather confidence interval that contains the $SI(t+1)$ value with 90% probability. To calculate this interval, the conditional distribution is approximated with normal distribution. With this approximation the 5. and the 95. percentile can be used as the border values of the interval. It has 90% probability that the $SI(t+1)$ will be between these two points, moreover the width of the interval is minimal.

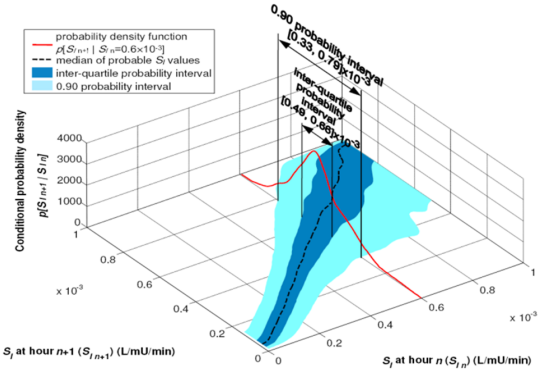

Figure 1: *Conditional density function defining the conditional probability distribution of $SI(t+1)$ for a given $SI(t)$*

Recently new artificial intelligence, especially neural network based models were created with the aim of replacing the currently used prediction method. There are studies that analysed the effects of involving additional parameters into the prediction [19, 20]. They showed clear benefits by developing models on the so called 3D SI distributions. These models can easily handle the involvement of additional prediction parameter that makes possible to create even higher dimensional models and make the treatment more personalized. These models went through a pre-validation step and showed promising results on some statistical metrics derived from the application criteria compared to the currently used model.

In this paper the results of the in-silico validation were presented. This validation simulates the clinical situation on virtual patients that were created from the historical treatment data. The models showed results that make them applicable in the ICU environment. The outcome of treatment based on the decisions of the new models was better in some aspects than the original. The results also confirmed our hypothesis about the benefits of using higher dimensional SI prediction: Including additional parameters into the prediction greatly increased the predictive performance. In the simulated clinical treatments the STAR protocol could recommend more accurate treatment parameters that resulted in a more steady blood glucose trend and in general a safer treatment.

2. Methods and Data

2.1. Patient Selection and SI Data Set Used

The patient's parameters under STAR treatment are collected in several studies. The dataset used in this study was collected between June 2016 and August 2019 and filtered by the following excluding rules:

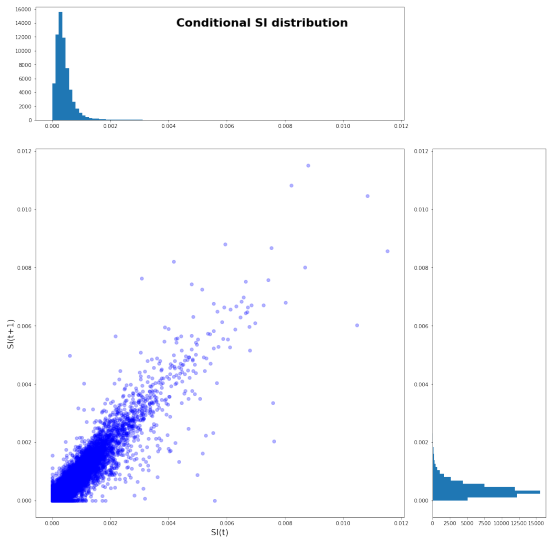

Figure 2: *Input data points defined by the $SI(t)$ and $SI(t+1)$ data pairs. The histogram of the values are shown on the top and right side of the figure.*

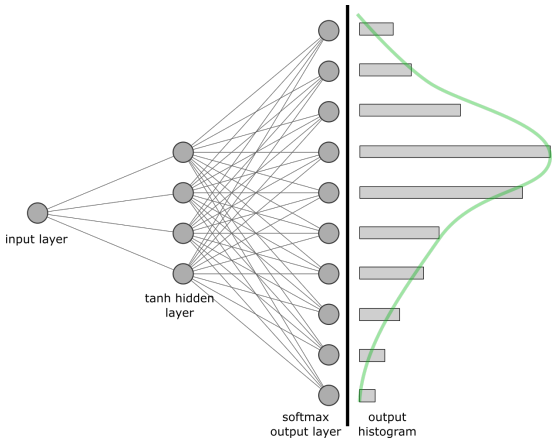

Figure 3: *Schematic structure of the 2D CDN network with the output histogram and the fitted normal distribution (green curve).*

- patients treated less than 10 hours by STAR;
- sections of treatments where the higher border of the BG target band was above 9 mmol/L;
- sections of treatments where lower border of the BG target band was above 6 mmol/L.

Data points used for generating the prediction models are created for each real BG measurements. The actual $SI(t)$ and $SI(t+1)$ values are identified using the ICING model and the treatment data. These data pairs make up the 2D SI distribution, which is shown in Figure 2. The histogram of the data points are shown on the top and right side of the Figure. The total number of data points was $65,052$.

Involving additional input parameters higher dimensional models can be created. In this research 2D, 3D and 6D models were created. The 2D models use the above mentioned data pairs. On top of these parameters the 3D models uses the $SI(t-1)$ input parameter. The 6D models use in addition the current blood glucose level and the nutrition and insulin intake in the last hour. The selection of the input parameters was based on correlation analysis.

2.2. SI Prediction Based on Deep Neural Network

The basic idea of this method is to translate the confidence interval prediction problem to a classification problem. To apply the classification deep neural network (CDN) for the prediction of the $SI(t+1)$ distribution the codomain of SI was divided into 100 equal size intervals. Each interval was associated with one output class. The output layer of the CDN consists of 100 nodes, associated with the classes defined above. The CDN nodes will define for each output class the probability that the SI domain associated with the given class includes the predicted $SI(t+1)$ value. Thus we can calculate the confidence interval by combining the subintervals based on the probability. Schematic structure of the 2D CDN network is shown on Figure 3. The actual number of layers and their sizes can be found in Table 1 for the 2D model and in Table 1 for the 3D model.

To create the training data, the dataset was preprocessed. Each data point was given with a class label by which interval contains the $SI(t+1)$ value. After that the class label was one-hot encoded and used in the training instead of the actual $SI(t+1)$ value. One-hot encoding a class label means creating a bit vector with zero values at all position except the one related to the class label, which will be one. It is a standard method to construct the training data for a classification networks with softmax output layer. Also, the input parameters were individually normalized by the so called Standard Scaler that transforms the data as it was coming from standard normal distribution (0 mean, 1 standard deviation).

The CDN uses a standard multiclass classification architecture. The size of the input layers depends on the number of input variables. In the hidden layers the tangent hyperbolic activation function is used. In the output layer the softmax activation function was employed, which is standard in the multiclass classification. As an optimizer, ADAM [21] is found to be the best in our experiments.

Several network topologies have been tested in the initial phase of our study, in Table 1 the best performing 2D CDN topology is defined.

The deep neural network used for SI prediction was implemented in Python using TensorFlow and Keras [22] . 80%

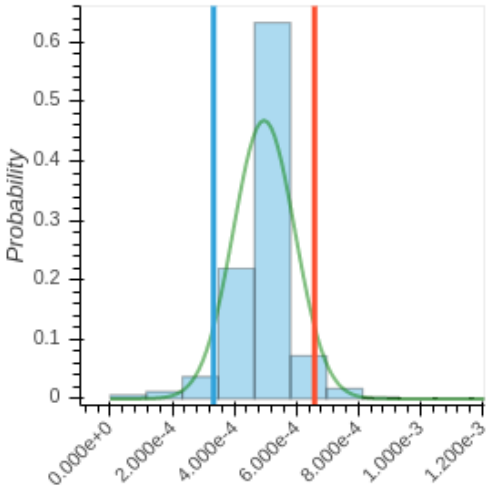

Figure 4: *The output of the 2D CDN network prediction for a given $SI(t)$ value. The fitted Gaussian distribution (green) and the 5% and 95% percentile values are also shown.*

of the input data set was used for training. The training consisted of 30 epochs.

The final output of the SI prediction is calculated by fitting a Gaussian distribution to the output of the deep neural network which is considered in this case as a histogram (see Figure 3). This calculation is illustrated in Figure 4. In this figure the output of the deep neural network prediction is shown as a blue histogram. The fitted Gaussian distribution is the green line. The mean value of the Gaussian is considered to be the actual $SI(t+1)$ value. The 5% and 95% percentile values are also shown by blue and red lines. Between these values are the 90% confidence interval so this will be used as the result of the prediction. It is important to note that the result of the Gaussian fitting and percentile computation are in the class label domain, thus they need to be transformed back to the $SI(t+1)$ domain. There are also some tweaking options like translating the mean value to the midpoint of the slice interval, or tweaking the deviation to force the predicted interval width to be at least the width of the slice interval.

layer type	size	activation function
input	1	-
hidden	10	tanh
hidden	20	tanh
hidden	30	tanh
output	100	softmax

Table 1: *Classification deep neural network topology definition for 2D SI prediction*

layer type	size	activation function
input	2	-
hidden	20	tanh
hidden	30	tanh
hidden	40	tanh
output	100	softmax

Table 2: *Classification deep neural network topology definition for 3D SI prediction*

We can extend the method to higher dimensional SI prediction by connecting a new input parameter, for example in the 3D case with the $SI(t-1)$ parameter. This modification involves changes in the pre-process stage and in the network topology, but not in the post-process stage. The Gauss fitting stage will remain the same because the network output will also remain the discrete distribution discussed before. Table 2 shows the topology of the extended model for 3D prediction.

2.3. Creation of higher dimensional models

Higher dimensional models could be created by including additional input parameters into the prediction. The higher dimensional CDN models differ only in the dataset and in the size of the input layer. Other stages of the prediction method are invariant to the additional parameters as you can see on Table 2, because the interval separation and the class labelling only depends on the $SI(t+1)$ input parameter and the output of the network is always a one dimensional histogram, so any stage behind the network operation will always use the same type of input.

In the Mixture Density Network (MDN) models this post-process invariance is still true. The output layer of the networks only depends on the number of subdistributions. Moreover, the implementations of the post-process stages at both methods can handle arbitrary number of classes in the CDN case and arbitrary number of subdistributions in the MDN case.

2.4. SI Prediction based on Mixture Density Network

The Mixture Density Network [23] method assumes that the conditional distribution of $SI(t+1)$ can be modelled as mixture of more Gaussian distributions where the parameters (means, standard deviations, weights) depend on $SI(t)$ and additional input parameters. Therefore, the output of an MDN network is a vector containing the parameters of a specified number of Gaussian distribution. The number of the subdistributions has to be specified.

The network architecture is a standard regression architecture with semantically defined neurons as the parameters of the subdistributions at the output (see Figure 5). In the

training, the loss is calculated by the similarity of the input distribution with the output distribution sampled from the output parameters.

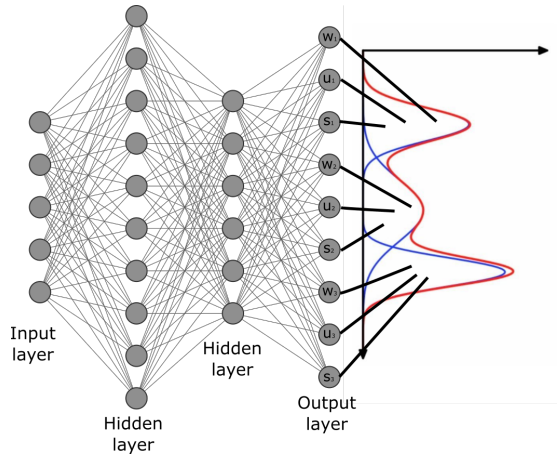

Figure 5: *Schematic structure of the 6D MDN network with the mixture Gaussian distribution.*

The MDN network was also implemented in TensorFlow with Keras and an additional library named keras-mdn-layer [24]. For the numerical stability, the usage of normalization with the so called Min Max Scaler is recommended. This normalization maps the values between the given minimum and maximum value.

Postprocessing step is needed for the confidence interval definition which is calculated by the 5th and 95th percentile of the distribution. The percentile calculation of the mixture distribution lacks canonical implementation at the time of writing this paper, thus a self-developed method was used. Eq 1 shows the PDF (probability density function) of the mixture gaussian distribution:

$$f(x) = \sum_{i=1}^{n} w_i \phi(x, \alpha_i, \beta_i^2) \tag{1}$$

where $f(x)$ is the PDF of the mixture distribution, w_i is the weight of the i-th subdistribution, ϕ is the PDF of the normal distribution, α_i is the mean of the i-th subdistribution, β_i^2 is the variance of the i-th subdistribution and n is the number of the subdistributions.

Based on Eq 1, the CDF (cumulative distribution function) function are calculated as it can be seen in Eq 2:

$$F(x) = \sum_{i=1}^{n} w_i \mathcal{N}(x, \alpha_i, \beta_i^2) \tag{2}$$

where $F(x)$ is the CDF of the mixture distribution, w_i is the weight of the i-th subdistribution, $\mathcal{N}$ is the CDF of the normal distribution, α_i is the mean of the i-th subdistribution, β_i^2 is the variance of the i-th subdistribution and n is the number of the subdistributions.

The percentile function also called inverse CDF can be calculated by inverting the CDF. This can be done by numerical root finding. It is important that the CDF of the mixture gaussian distribution is strictly increasing, because it is sum of individually strictly increasing functions. Therefore, it can be inverted without special conditions. For the sake of guaranteed convergence a bracketing based method was used for the root finding. This method needs a starting interval that contains the root. In order to find an interval the mixture distribution was fitted with a single normal distribution. The parameters of the fitted distribution can be calculated as seen in Eq 3 and 4:

$$\mu = \mathrm{E}[X] = \sum_{i=1}^{n} w_i \, \mathrm{E}[\mathcal{N}(x, \alpha_i, \beta_i^2))] = \sum_{i=1}^{n} w_i \alpha_i \tag{3}$$

$$\sigma^2 = \mathrm{E}[(X-\mu)^2] = \mathrm{E}[X^2] - \mu^2 = \sum_{i=1}^{n} w_i(\beta_i^2 + \alpha_i^2 - \mu^2) \tag{4}$$

where μ is the mean of the mixture distribution, σ^2 is the variance of the mixture distribution, E is the expected value function, w_i is the weight of the i-th subdistribution, $\mathcal{N}$ is the CDF of the normal distribution, α_i is the mean of the i-th subdistribution, β_i^2 is the variance of the i-th subdistribution and n is the number of the subdistributions.

Based on these values, the interval of the bracketing root finding can be defined as $[\mu - 6\sigma, \mu + 6\sigma]$. There is a minimal probability that this interval will not contain the root. In this case the interval can be heuristically widened to fit the criteria.

2.5. In-silico validation

A new version of the STAR protocol using the proposed SI prediction methods has to go through a virtual trial, called in-silico validation, before it can be used in real clinical environment. The trial simulates the clinical environment from historical patient data, by creating virtual patients and then treat them based on the decisions of the protocol under test. The details of in-silico validation procedure can be seen on Figure 6 [25].

The theory behind the trial is that the insulin sensitivity of the patient is independent from the nutrition and insulin dosage, it depends only on the state of the patient. As the patient goes through the treatment and heals, his/her SI values follows this change in the state. Therefore, we assume that the historical SI values of the patients remain relevant even with modified insulin and nutrition dosage.

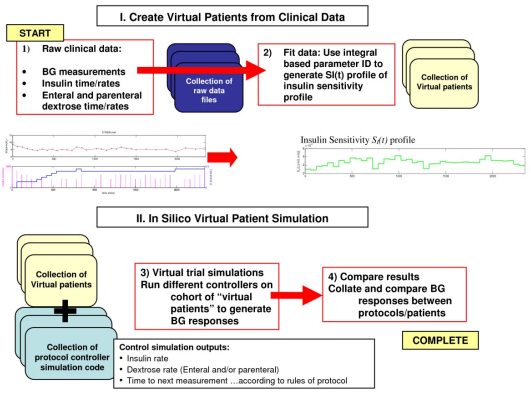

Figure 6: *The process of the in-silico validation.*

During the in-silico validation the patient's treatment datasets are processed one by one. The first recommendation of the protocol is a rule based one using the current blood glucose, because at this point there is no known SI value of the patient. The first blood glucose value comes from the historical data of the patient, after that the new value is computed by the ICING model based on the recommended nutrition and insulin dosage of the protocol and the historical SI value. After that there are also available SI information, so for the next recommendations the protocol can use the SI prediction method.

During the simulation the validator stores the recommendations of the protocol and follows the trend of the glucose values. It collects the clinically important data, like hours spent in each BG region or the count of hypo events and at the same time it also compares them to the in-use protocol.

3. Results

In the in-silico validation the historic treatment data comes from 112 randomly selected patients different from those used to train the neural networks. The simulation collects various kinds of data, mostly values that the protocol collects during the treatment. It collects the measured BG levels, the added insulin and nutrition and makes statistics from them. The most important statistics is the BG statistics. The validation sorts the BG measurements into ranges based on the effect or severity. The validation calculates how much time the patient spent in different BG regions. The simulation also creates the CDF function of the BG measurements to visually compare the new model to the original one.

3.1. Classification Deep Network

The results of the in-silico validation on the CDN models can be seen in Table 3. The most important parameter from the aspect of safety is the number of episodes with $BG < 2.22$ and $BG < 4.0$. These are the dangerous hypo events, and low BG levels that have to be avoided. The table shows that all the CDN models successfully reduced the number of these events from 2 to 1. In the 2D case there is only one dangerous hypo event and the low level events also reduced to 36. These values were 2 and 44 in the original protocol version. In the 3D case, that one hypo events seems to be shifted into the low BG level category, that reduced further in the 6D case.

Cohort Directory	original	cdn2D	cdn3D	cdn6D
BG median [IQR] (mmol/L):	5.68	5.94	5.82	6.01
BG mean (geometric) (mmol/L):	5,9295	6,1548	6,0537	6,2241
BG StDev (geometric) (mmol/L):	1,2830	1,2743	1,2809	1,2616
Num episodes <4.0 mmol/L	44	36	37	34
Num episodes <2.22 mmol/L	2	1	0	0
% BG <2.22 mmol/L	0,0335	0,0166	0	0
% BG <4.0 mmol/L	2,6448	1,7453	1,9657	1,4184
% BG <4.4 mmol/L	5,7750	3,7068	4,5644	3,4816
% BG within 4.4 - 6.5 mmol/L	66,2203	63,0319	64,7343	61,5893
% BG within 4.4 - 7.0 mmol/L	73,3847	72,2241	72,8969	72,0986
% BG within 4.4 - 8.0 mmol/L	83,2775	83,4441	83,2584	83,6396
% BG within 8.0 - 10 mmol/L	7,1811	8,9262	8,1459	9,0909
% BG >10 mmol/L	3,8333	3,9894	4,0980	3,8524

Table 3: *Results of the in-silico validation on the CDN models*

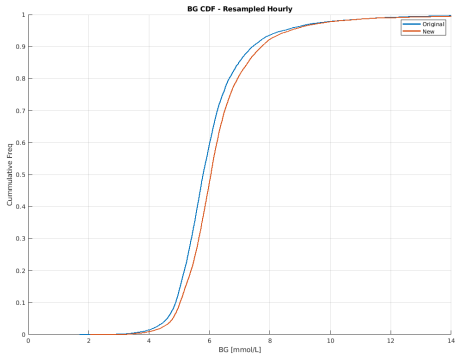

Figure 7: *Cumulative Density function of the BG values based on 2D CDN (red) and the currently used model (blue).*

The mean BG level increased which could be beneficial as higher energy intake potentially makes the healing process faster. This increase also affects the number of episodes with high BG level ($BG > 10$), but in the 6D case this means 0.02%, which is not that significant.

BG CDF is a convenient method to visually compare the models. The CDF of the BG measurements for the different CDN models are displayed on Figure 7, 8 and 9.

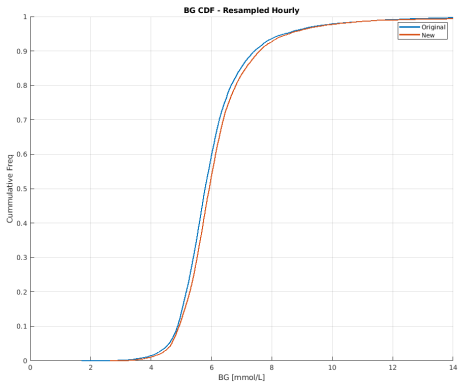

Figure 8: *Cumulative Density function of the BG values based on 3D CDN (red) and the currently used model (blue).*

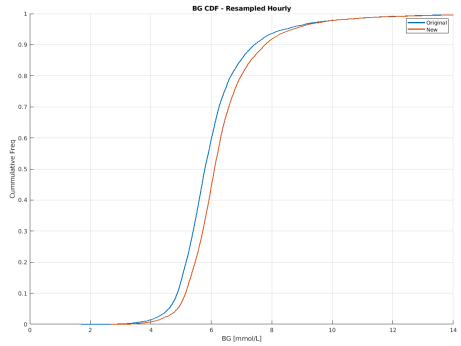

Figure 9: *Cumulative Density function of the BG values based on 6D CDN (red) and the currently used model (blue).*

At the higher BG this CDF begins to rise the better treatment result is represented by the CDF, if the rise still falls within the normal glycaemic range of the CDF. Thus, the CDF should also reach its maximum within the normal glycaemic range. The strategy to evaluate these graphs is to visually compare the new function curve to the original one to see how much it is shifted to the higher values, and simultaneously, how the endpoint is related to the original one. This visualization also helps to compare the models similarity.

3.2. Mixture Density Network

The results of the in-silico validation on the MDN models can be seen in Table 4 . The same rules apply to the evaluation as for the CDN case. The most conspicuous difference from the CDN models is that the 3D version of the MDN method has more low BG level values than the original, and there is a hypo event unlike in the 3D CDN case. So the 3D variant of the MDN method requires some further analysis.

Cohort Directory	original	mdn2D	mdn3D	mdn6D
BG median [IQR] (mmol/L):	5.68	5.83	5.68	6.25
BG mean (geometric) (mmol/L):	5,9295	6,0159	5,8954	6,3630
BG StDev (geometric) (mmol/L):	1,2830	1,2824	1,2875	1,2595
Num episodes <4.0 mmol/L	44	42	48	31
Num episodes <2.22 mmol/L	2	1	1	0
% BG <2.22 mmol/L	0,0335	0,0169	0,0169	0
% BG <4.0 mmol/L	2,6448	2,4777	2,7265	1,2882
% BG <4.4 mmol/L	5,7750	5,2082	6,3844	3,0113
% BG within 4.4 - 6.5 mmol/L	66,2203	63,2395	65,3683	54,3478
% BG within 4.4 - 7.0 mmol/L	73,3847	71,7849	72,9213	67,5684
% BG within 4.4 - 8.0 mmol/L	83,2775	83,0440	82,5910	83,6715
% BG within 8.0 - 10 mmol/L	7,1811	7,9386	7,1465	9,4686
% BG >10 mmol/L	3,8333	3,8766	3,9458	3,9130

Table 4: *Results of the in-silico validation on the MDN models*

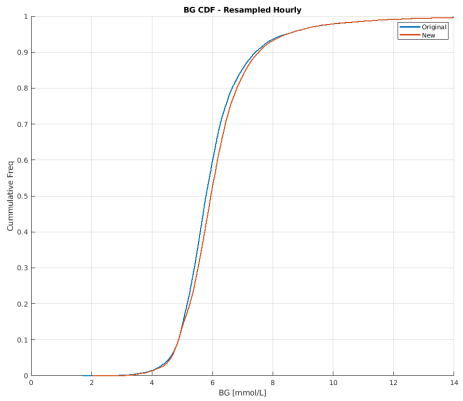

Figure 10: *Cumulative Density function of the BG values based on 2D MDN (red) and the currently used model (blue).*

The 6D variant of this method produced the best validation results. It showed only 31 episodes with $BG < 4.0$ without dangerous hypo event. Its only disadvantage is the increased number of high BG level measurement.

The CDF of BG belonging to MDN models can be seen on Figure 10, 11 and 12.

The same CDF based evaluation rules can be applied to the MDN cases, as well. At the higher BG the rise of this CDF begins the better result is represented by the CDF, if the rise still falls within the normal glycaemic range . On the other hand, the CDF should also reach its maximum within the normal glycaemic range.

The visualization mirrors the significant differences in the

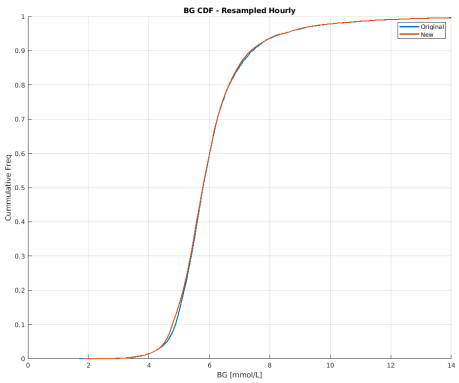

Figure 11: *Cumulative Density function of the BG values based on 3D MDN (red) and the currently used model (blue).*

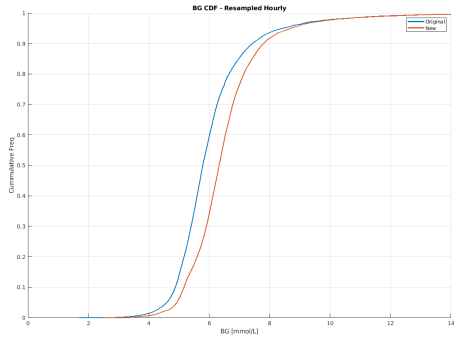

Figure 12: *Cumulative Density function of the BG values based on 6D MDN (red) and the currently used model (blue).*

6D case (Figure 12). The most similar model to the currently used is the 3D MDN, based on the CDF (Figure 11).

4. Discussion

In-silico simulation applied in the second phase of validation is a more complex procedure. The validation measures the time that the patient spends in different BG regions. The top priority from the aspect of safety is to avoid the dangerous hypoglycaemic events ($BG < 2.2$ mmol/L) because hypoglycaemia can cause serious conditions, like coma, in a very short time. Beyond this, avoiding BG events below 4.4 mmol/L is also important because they can easily turn into hypo events by the time and also the lower BG level make the healing process slower. The 4.4-6.5 mmol/L range is the ideal BG domain. This is the healthy range in which the patient has comfort and heals steadily. Over that range the BG level have to be decreased. The BG level over 8.0 mmol/L is also avoidable. This case is not as dangerous as the hypo event but can cause complications in the long run. This case can be tolerable if it does not last for several hours continuously. Keeping this in mind the different models can be compared as follows.

In the CDN case applying additional input parameters resulted in better success rate but widened BG prediction ranges were experienced. In some cases the 6D CDN network resulted in better success rate with average interval width smaller than the one of the currently used model. In the 2D and 3D cases widened intervals are more common. In these cases a decision has to be taken if the additional success rate has bigger benefits than the disadvantages arised by the widened intervals. For this trade off two endpoints have to be defined. The first endpoint can be derived directly from the requirements: the success rate have to be at least 90%. The second endpoint relates to the interval widening that is derived indirectly from the requirements. There is no canonical value to the threshold of the upper limit but the successful results of the in-silico validation can indicate if the upper limit was sufficient.

Between the endpoints, mentioned above, there is a method that can help evaluate the options. Using the normal distribution approximation and rounding the currently used protocol success rate to 90%, it can be claimed that to improve the success rate the only solution is to widen the intervals. For example, if we want to improve the success rate of the prediction to 92% we have to use the 0.04 and 0.96 percentile value instead of 0.05 and 0.95. In order to calculate the necessary interval rate for 92% success, the percentile value of 0.96 has to be divided by the percentile value of 0.95. This ratio shows how the width of the confidence interval defined by a single normal distribution changes in order to achieve the better success rate. In the example above the calculated ratio is 1.06434 resulting in 6% widening that is a good compromise for additional 2% of success rate because statistically it can not be achieved with less widening.

Based on this method it can be claimed that the CDN networks make good compromises to optimize on success rate. This nature of the CDN network depends on the number of the classes. With more classes the improvement of the success rate is not so significant, however, the widening is more restrained.

The in-silico validation showed improvements in the distribution of the patients blood glucose labels. In the 2D case there was only one dangerous hypo event ($BG < 2.2$ mmol/L) instead of two and in the 3D and 6D cases there were zero dangerous hypo event. The number of the episodes with low glucose level ($BG < 4.0$ mmol/L) has also decreased from 44 to 34. In general, BG values were shifted to a higher range. This could be beneficial in general, because the patients are further away from the dangerous hypo range, but there are cases when the BG level goes higher than the normal range and causes hyperglycemia. The hyperglycemia

is not as dangerous as the hypoglycemia, but also has to be avoided.

In the MDN case the significant improvement is shown in the narrower average interval ratio. This architecture is a good option to improve the interval ratio while maintaining the success rate. The additional parameters have also some improvement on the success rate. Because of that, this network type can be used without trade off considerations based on the data from the pre-validation.

The in-silico validation showed that the 2D MDN network decreased the dangerous hypo events from two to one, and the number of episodes with low BG ($BG < 4.0$ mmol/L) from 44 to 42. Unexpectedly, the number of these episodes increased in the 3D case to 48. This increment appeared also in the CDN case, thus this case needs further analysis. On the contrary, the 6D MDN prediction showed significant improvements in this feature. There was no hypo event in this prediction case and the number of low BG episodes decreased to 31, which makes this aspect better than the 6D CDN network. The trade off is that this network based prediction caused slightly more hyperglycaemic events than the CDN network but this 0.06% increment worth the trade probably.

It also worth to mention that both methods decreased the standard deviation of the BG values, which means the BG level of the patients was more stable during the treatment.

In general, both methods proved that they are able to replace the currently used model and perform better in some respects. If the goal was to pick the most similar model to the currently used one the choice would be the 3D MDN model. The BG CDF diagram of the 3D MDN model shows similarity to the currently used model. There are only some diversity at lower values but they are in the normal range.

Our experiments also validated the improved SI prediction results using higher dimensional inputs experienced by other researches [19, 20], as well. The 6D models trained on the explicit patient data resulted in serious improvements, for example the 6D MDN model avoided all the hypo events compared to the currently used model and even halved the time spent in the low BG ($BG < 4.0$ mmol/L) range. Additionally, there are assumptions that other derived patient and treatment data can be used to achieve higher prediction performance.

5. Conclusions

In this research two different neural network based methods were presented and evaluated to replace the currently used model of the STAR SI prediction.

For the evaluation the so-called in-silico validation was used, a standard, simulation based method, that simulates the clinical treatment on virtual patients. Higher dimensional models were created for both of the newly developed SI prediction methods to make the prediction more accurate and personalized. The additional parameters resulted in significant improvement as in the pre-validation statistics, so in the results of the in-silico validation, but the field of the improvement was different in each method.

The in-silico validation on virtual patients correlates with the statistical metrics of the pre-validation. With the additional input parameters, the higher dimensional models can avoid the dangerous hypoglycaemic events but the BG CDF shifted to a slightly higher BG range in both methods. The higher BG level sometimes ends in hyperglycemia, which is to be avoided, so in the model selection it also involves a trade off consideration.

In conclusion both methods proved to be able to replace the currently used model, and based on the in-silico validation they are ready to the real clinical trial.

5.1. Further Work

The hyper parameter optimisation can be the next step of the research. An optimal number of classes in the CDN case and subdistributions in the MDN case can be found based on the presented evaluation method.

Another option is to improve the post-process stages. There are many options to calculate the endpoints of the confidence interval in both architectures. In the CDN case a new method can be developed that relies more on the histogram than the normal fitting while in the MDN case the numerical root finding method can be optimized.

Acknowledgements

The research was supported by the Hungarian National Scientific Research Foundation, Grant No. K116574, by the BME-Biotechnology FIKP grant of EMMI (BME FIKP-BIO), by the EFOP-3.6.1-16-2016-00017 project, and by H2020 MSCA-RISE DCPM (#872488) grant.

References

1. Karen C McCowen, Atul Malhotra, and Bruce R Bistrian. Stress-induced hyperglycemia. *Critical care clinics*, 17(1):107–124, 2001.

2. Naeem A Ali, James M O'Brien Jr, Kathleen Dungan, Gary Phillips, Clay B Marsh, Stanley Lemeshow, Alfred F Connors Jr, and Jean-Charles Preiser. Glucose variability and mortality in patients with sepsis. *Critical care medicine*, 36(8):2316, 2008.

3. Van Den Berghe et al. Intensive insulin therapy in critically ill patients. *New England journal of medicine*, 345(19):1359–1367, 2001.

4. James Stephen Krinsley. Effect of an intensive glucose management protocol on the mortality of critically ill adult patients. In *Mayo Clinic Proceedings*, volume 79, pages 992–1000. Elsevier, 2004.

5. Reed et al. Intensive insulin protocol improves glucose control and is associated with a reduction in intensive care unit mortality. *Journal of the American College of Surgeons*, 204(5):1048–1054, 2007.

6. J Geoffrey Chase, Christopher G Pretty, Leesa Pfeifer, Geoffrey M Shaw, Jean-Charles Preiser, Aaron J Le Compte, Jessica Lin, Darren Hewett, Katherine T Moorhead, and Thomas Desaive. Organ failure and tight glycemic control in the SPRINT study. *Crit Care*, 14(4):R154, 2010.

7. Benyo et al. Pilot study of the SPRINT glycemic control protocol in a Hungarian medical intensive care unit. *Journal of diabetes science and technology*, 6(6):1464–1477, 2012.

8. Stewart et al. Safety, efficacy and clinical generalization of the STAR protocol: a retrospective analysis. *Annals of intensive care*, 6(1):24, 2016.

9. Dubois et al. Software-guided versus nurse-directed blood glucose control in critically ill patients: the LOGIC-2 multicenter randomized controlled clinical trial. *Critical Care*, 21(1):212, 2017.

10. Aaron J Le Compte, Adrienne M Lynn, Jessica Lin, Christopher G Pretty, Geoffrey M Shaw, and J Geoffrey Chase. Pilot study of a model-based approach to blood glucose control in very-low-birthweight neonates. *BMC pediatrics*, 12(1):117, 2012.

11. Schultz et al. Adoption and implementation of the original strict glycemic control guideline is feasible and safe in adult critically ill patients. *Minerva anestesiologica*, 78(9):982–995, 2012.

12. Alicia Evans, Aaron Le Compte, Chia-Siong Tan, Logan Ward, James Steel, Christopher G Pretty, Sophie Penning, Fatanah Suhaimi, Geoffrey M Shaw, Thomas Desaive, et al. Stochastic targeted (STAR) glycemic control: design, safety, and performance. *Journal of diabetes science and technology*, 6(1):102–115, 2012.

13. Jessica Lin, Normy N Razak, Christopher G Pretty, Aaron Le Compte, Paul Docherty, Jacquelyn D Parente, Geoffrey M Shaw, Christopher E Hann, and J Geoffrey Chase. A physiological Intensive Control Insulin-Nutrition-Glucose (ICING) model validated in critically ill patients. *Computer methods and programs in biomedicine*, 102(2):192–205, 2011.

14. Chase et al. Tight glycemic control in critical care–the leading role of insulin sensitivity and patient variability: a review and model-based analysis. *Computer methods and programs in biomedicine*, 102(2):156–171, 2011.

15. Suhaimi et al. What makes tight glycemic control tight? The impact of variability and nutrition in two clinical studies. *Journal of diabetes science and technology*, 4(2):284–298, 2010.

16. Timothy Lonergan, Aaron Le Compte, Mike Willacy, J Geoffrey Chase, Geoffrey M Shaw, Xing-Wei Wong, Thomas Lotz, Jessica Lin, and Christopher E Hann. A simple insulin-nutrition protocol for tight glycemic control in critical illness: development and protocol comparison. *Diabetes technology & therapeutics*, 8(2):191–206, 2006.

17. Balazs Benyo, Attila Illyés, Noémi Szabó Némedi, Aaron J. Le Compte, Attila Havas, Levente Kovacs, Liam Fisk, Geoffrey M. Shaw, and J. Geoffrey Chase. Pilot study of the sprint glycemic control protocol in a hungarian medical intensive care unit. *Journal of Diabetes Science and Technology*, 6(6):1464–1477, 2012. PMID: 23294794.

18. Jessica Lin, Dominic Lee, J Geoffrey Chase, Geoffrey M Shaw, Aaron Le Compte, Thomas Lotz, Jason Wong, Timothy Lonergan, and Christopher E Hann. Stochastic modelling of insulin sensitivity and adaptive glycemic control for critical care. *Computer methods and programs in biomedicine*, 89(2):141–152, 2008.

19. Uyttendaele et al. A 3D insulin sensitivity prediction model enables more patient-specific prediction and model-based glycaemic control. *Biomedical Signal Processing and Control*, 46:192–200, 2018.

20. Vincent Uyttendaele, Jennifer L. Knopp, Shaun Davidson, Thomas Desaive, Balazs Benyo, Geoffrey M. Shaw, and J. Geoffrey Chase. 3D kernel-density stochastic model for more personalized glycaemic control: development and in-silico validation. *BioMedical Engineering OnLine*, 18(1):102, 2019.

21. Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

22. Antonio Gulli and Sujit Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

23. Christopher M. Bishop. Mixture density networks. Workingpaper, Aston University, 1994.

24. Charles Martin and Douglas Duhaime. cpmpercussion/keras-mdn-layer v0.3.0. Nov 2019.

25. J. G. Chase, F. Suhaimi, S. Penning, J. C. Preiser, A. J. Le Compte, J. Lin, C. G. Pretty, G. M. Shaw, K. T. Moorhead, and T. Desaive. Validation of a model-based virtual trials method for tight glycemic control in intensive care. *Biomed Eng Online*, 9:84, Dec 2010.