Analysis of the performance of Genetic Programming on the Blood Glucose Level Prediction Challenge 2020

David Joedicke[1,4], Oscar Garnica [2], Gabriel Kronberger[1], J. Manuel Colmenar[3]

Stephan Winkler[4,1], J. Manuel Velasco[2], Sergio Contador[2,3], J. Ignacio Hidalgo[2]

Abstract. In this paper we present results for the Blood Glucose Level Prediction Challenge for the Ohio2020 dataset. We have used four variants of genetic programming to build white-box models for predicting 30 minutes and 60 minutes ahead. The results are compared to classical methods including multi-variate linear regression, random forests, as well as two types of ARIMA models. Notably, we have included future values of bolus and basal into some of the models because we assume that these values can be controlled. Additionally, we have used a convolution filter to smooth the information in the bolus volume feature. We find that overall tree-based GP performs well and better than multi-variate linear regression and random forest, while ARIMA models performed worst on the here analyzed data.

1 INTRODUCTION

This paper describes our contribution to the Blood Glucose Level Prediction Challenge (BGLPC) for the Ohio2020 dataset described in [15]. We present a comparison among different algorithmic techniques related to linear regression applied to this glucose prediction problem, where we highlight four of them, based on tree-based Genetic Programming (GP) [14]: GP, GP with offspring selection [1] (GP-OS); and a single-objective as well as a multi-objective variant of Grammatical Evolution[16] denoted as GE and MOGE. In addition, we present three approaches based on classical methods. In particular, we consider Random Forest [2], denoted as RF, a multivariate linear regression, denoted as LR, and two ARIMA models [18], denoted as A-0 and A-1. All the methods will be briefly described in the following section, as well as the pre-processing of data we have performed. In data pre-processing several features where derived from exising data and added to the dataset. The experimental results will be analyzed in Section 3. We use root mean squared error (RMSE) and mean absolute error (MAE) as metrics to measure the accuracy of our results. Finally, the conclusions will be drawn in Section 4.

[1] Josef Ressel Center for Symbolic Regression, Upper Austria University of Applied Sciences. Emails: david.joedicke@fh-ooe.at, Gabriel.Kronberger@fh-hagenberg.at, stephan.winkler@fh-ooe.at.

[2] Universidad Complutense de Madrid. Emails: ogarnica@ucm.es, mvelascc@ucm.es, hidalgo@ucm.es.

[3] Rey Juan Carlos University. Email: josemanuel.colmenar@urjc.es.

[4] Johannes Kepler University Linz

2 ALGORITHMIC PROPOSAL

2.1 Data pre-processing

Data pre-processing proved to be challenging in this competition as the exact rules of the competition were rather opaque especially regarding usage of future information and the difference between the online and the offline case. The main pitfalls were: (i) the set of features is different for the six data contributors, (ii) different sampling rates for features, (iii) variance in the duration between sampling values (e.g. blood glucose values are usually sampled every five minutes but not always), (iv) some missing values are encoded as zeros (e.g. zero values for skin temperature).

In the ARIMA model we only used the glucose level. For all the other models we used the following data pre-processing steps. We prepared a Python script that we used for pre-processing training as well as testing data. We used only the set features which are available for all data contributors even though we built six individual models. Correspondingly, we only used the following features: glucose level, basal, bolus type, bolus dose, galvanic skin response (gsr), and skin temperature. We used numerical encoding to encode the categorical variable bolus type. For the skin temperature we removed all zeros values. For the basal value we replaced all missing values with zeros.

We incorporated lagged variables for our models (e.g. the glucose level five minutes ago). For this, we extended our dataset with lagged features, whereby we used a maximum lag of two hours. So, for each feature we produced 24 (120 min / 5 min) additional features. Hence, we require values at multiples of five minutes. This is not the case in the provided datasets. Therefore, we first prepared a intermediate larger dataset with one row for every minute (equidistant sampling). In this dataset, we had to fill missing values for glucose level, galvanic skin response, and skin temperature. For the training data we used linear interpolation to fill these gaps, for the test data we used the last known value, since future values should not be used to predict the glucose value. Using the sub-sampled and interpolated dataset we prepared the lagged features and finally we reduced the number of rows again by keeping only rows where we have a target glucose value (either 30 or 60 minutes ahead).

In our modelling efforts for GP and GP-OS, we assume that the basal value as well as the bolus type and dose can be controlled externally. This assumes an application of the model as part of a model-predictive controller for an insulin pump, whereby the goal is to optimize the automatic administration of insulin. Therefore, we have included "future information" for the blood glucose prediction. The variables that we assume to be controlled and known are: basal, bolus type, and bolus dose. For these variables we included forward look-

ing features up to the prediction horizon (6 features for 30 minutes and 12 features for 60 minutes).

Finally, we added features for smoothed bolus dose values using a convolution process. Even though the bolus dose is administered almost instantaneously, the effect is not immediate. Instead, the underlying dynamic uptake process has a longer-lasting diminishing effect. We used a convolution function (Bateman function) to produce smoothed features for the bolus dose. For this smoothed bolus dose we also prepared lagged features (backwards and forwards) using the same scheme as described above.

2.2 Algorithms

After pre-processing the data as described above, we used machine learning methods to find models that describe future values of glucose after 30 minutes, $\hat{g}_{t+30}$ and after 60 minutes, $\hat{g}_{t+60}$, as a function of basal value (bv), bolus dose (bd), basis GSR value (gsr), basis skin temperature (sk), bolus type (bt) and glucose level (gl):

$$\hat{g}_{t+30/t+60} = f(bv(t-60...t), bd(t-60...t), ...) \quad (1)$$

We used seven different algorithms to model the function described in Equation (1). Linear Regression (LR) and Random Forest (RF) are well known methods that are used as benchmarks for our models. Additionally, we used two GP, two GE algorithms, and two ARIMA models to predict the glucose value. Next, we detail our proposals[5].

2.2.1 Genetic Programming

Symbolic regression (SR) is a specific method of regression analysis, where the model is represented as a closed-form mathematical expression [14]. A unique characteristic of SR is that the model structure does not have to be pre-specified. Instead, a SR solver (i.e. GP) automatically constructs mathematical expressions from the set of input variables (with their respective allowed time offsets) as well as mathematical operators and functions.

We use genetic programming (GP), an evolutionary technique that iteratively produces solutions for a given optimization problem. GP is specifically designed to find programs that solve given tasks; when applied to SR, these programs are formulas that are based on of mathematical operators, variables, and constants. Being an evolutionary algorithm, GP initially creates a randomly set of formulas and then, over many generations, produces new formulas by means of crossover and mutation operators. The improvement of these formulas is reached by selection operators: in each generation the parents for the new solution candidates are selected, and new individuals can be inserted into the next generation either automatically or only if they are selected by some kind of offspring selection. We used the GP implementation in HeuristicLab [6] and created models with a maximum size of 100 nodes and ten levels. We used GP in two different variants:

- Standard GP (GP): 1000 individuals, tournament selection as parent selection mechanism, elitism, termination criterion: 1000 generations.
- Offspring selection GP (OSGP): 1000 individuals, random parents selection, strict offspring selection (i.e., individuals are sent to the next generation if they are better than their parents [1]), elitism, termination criterion: maximum selection pressure 200 (i.e., as soon as the number of individuals that have to be created so that 1000 successful ones are found in one generation has reached 200000).

2.2.2 Grammatical Evolution

Grammatical Evolution (GE) [16] is a variant of GP which uses chromosomes to encode the information of the individuals (trees). In GE, a grammar is applied to perform the decoding process that generates the trees which, in this case, will be the mathematical expressions that represent prediction models of glucose values. Given that this method uses chromosomes, it allows the application of classical genetic operators such as crossover or mutation directly at the chromosome level, instead of the tree level, as happens in GP. We evaluate two GE proposals:

- Standard GE: we follow the same implementation and grammars of [12].The GE approach only considers one objective function, which will be either RSME or MAE. We present here only the results with RMSE, since they are significantly better with the parameters used.
- Multi-Objective GE (MOGE): we propose a multi-objective implementation of GE where the underlying algorithm is the well-known NSGA-II [9]. The MOGE approach considers RSME as one of the objective functions and a custom objective function called F_{CLARKE} as the second objective. F_{CLARKE} is based on the Clarke Error Grid (CEG) metric, and was defined as shown in Equation (2). In the expression, $|E|$ represents the number of points that belong to zone E of CEG, which is the most dangerous one for the patient, $|D|$ corresponds to the second most dangerous zone, D, and $|C|$ corresponds to zone C. Zone B was not included in the formula because it represents a not very dangerous zone, and A corresponds to the safe zone. A more detailed explanation of F_{CLARKE} can be found in [7].

$$F_{CLARKE} = 100 \cdot |E| + 10 \cdot |D| + |C| \quad (2)$$

Prediction models with GE use information of the previous 60 minutes while MOGE models can use data from the previous two hours. Additional configurations will be explored and presented at the workshop. In all the experiments, both GE and MOGE, we perform 10 runs with 400 individuals over 1000 generations, random initialization of the population (half-ramped) allowing a maximum number of 5 wrappings using a crossover probability of 0.7 and a mutation probability of 0.1. Executions were run on our Pancreas Model Tool described in [13]. Unlike the GP description above, with the two GE variants we only use information of the past and present. We did not use all the generated features, but only those of every 15 minutes before. So, we use historical data from 120, 105, 90, 75, 60, 45, 30 and 15 minutes ago for MOGE and 60, 45, 30 and 15 for GE. We only consider the glucose level, basal, bolus type, bolus dose, galvanic skin response, and skin temperature variables. We would like to highlight that recent papers that combines GE with other techniques, such as, data augmentation [17], random GE and bagging [11] or clustering [8] achieved better results than the GE configurations studied in this paper. We limit GE in order to follow the instructions of the Challenge.

[5] Source files are available under request at absys@ucm.es https://drive.google.com/drive/folders/1TOGvl55iR10aqRFO8GoD2v6TQD4djiCE?usp=sharing

[6] https://dev.heuristiclab.com

2.2.3 *ARIMA model*

In addition to the GP and GE models, we have also fitted two auto-regressive integrated moving average, ARIMA(p, d, q) models to estimate glucose values. Equation (3) presents the expression of an ARIMA(p, d, q) model where g_s is the actual value of the glucose and ϵ_s is the random error at sample s, respectively, while p, q, and d integers called the orders of the model. All our models only include glucose values and do not use exogenous variables such as insulin doses or carbohydrates.

$$\hat{g}_s = \sum_{i=1}^{p} \alpha_i g_{s-i} + \left(\epsilon_s + \sum_{i=1}^{q} \theta_i \epsilon_{s-i} \right) + \sum_{i=0}^{d} \phi_i s^i \qquad (3)$$

We evaluate both off-line and on-line models. The off-line models are created using the training data for each patient. We define 192 models by sweeping the three ARIMA parameters as follows. The auto-regressive order ranges in $p \in [2, 10]$, the moving average $q \in [2, 10]$, and the integrative part uses the values $d \in [0, 1]$, so that $9 \times 9 \times 2 = 192$. The basics behind the election of these ranges is that the model takes into account glucose values up to 10 samples (50 minutes) previous to the current time. The model's coefficients –up to $p + q + d$ coefficients per model– are estimated using maximum likelihood given the univariate glucose time series, g_s, on the complete training dataset for each patient. Once the 192 models have been estimated, we select two models per patient: the model with the lowest RMSE at 30-minutes horizon and the one with the lowest RMSE at 60 minutes.

Regarding on-line models, with each new glucose value in the testing dataset, the procedure defines a 4-hour time window using the last 48 samples –including the last one–, and it estimates the 192 ARIMA models over the time window using maximum likelihood. Again, the 192 models are created by sweeping the three ARIMA parameters, as stated above. Next, we select the best model. Unlike off-line models, now we cannot use future glucose values to select the model that will provide the lowest RMSE in the future. Hence, we select the current best model based on the history of the best models up to the current sample. We have evaluated four different criteria to choose the best model for 30-minutes predictions and six criteria for 60-minute predictions.

- We select the values of (p, q, d) of the model with the lowest absolute error 30 minutes ago to create the current model for 30-minutes and 60-minutes predictions. Note that given the current glucose value, we know the model with the lowest error 30 minutes ago.
- We select the values of (p, q, d) of the model with the lowest absolute error 60 minutes ago in the prediction of the current glucose to create the current model for a 60-minutes prediction.
- We select the values of (p, q, d) of the off-line model for 30-minutes and 60-minutes predictions.
- We define an "ensemble" ARIMA averaging the value of p and q for the six best models 30, 35, 40, ..., and 55 minutes ago. We use the rounded averaged values of p and q to create the current model for 30-minutes and 60-minutes predictions.
- Similar approach than the previous item, but we average the parameters of the six best models 60, 65, ..., and 85 minutes ago. We use the rounded averaged values of p and q to create the current model for a 60-minutes prediction.
- We select the model with the lowest Akaike Information Criterion (AIC) value to estimate both, 30-minutes and 60-minutes predictions. AIC is a criteria to compare models with different number of parameters and select the models with better trade-off between goodness-of-fit and the number of parameters of the model, a.k.a parsimony.

In some cases, the procedure cannot bring the best model because the parameters that provided the best estimation either 30 minutes or 60 minutes ago cannot produce a stable ARIMA model in the current time. Due to this fact, the overall best-performing criteria is to choose the current ARIMA model using the Akaike Information Criterion.

3 EXPERIMENTAL RESULTS

Table 1 presents the experimental results in terms of RMSE and MAE for all the algorithms and for both 30 and 60 minutes prediction horizons. For GP, GP-OS, and GE, predictions are obtained with the models that obtained the lowest RMSE value in the training phase after 10 runs. The remaining 9 models were not evaluated nor reported. For the MOGE, also 10 run were made in the training phase, and from all the solutions of the 10 Pareto fronts, we also selected the model with RMSE value, independently of the value of the F_{CLARKE}. Results represent the values for the predictions of this selection. GE and MOGE were run just with the configuration explained on section 2.2.2 and no parameters optimization was performed.

Regarding GP and GP-OS results on table 1 may differ from those reported in the submitted files. After analyzing the results we noticed that at the beginning and the end of the data our results are fluctuating. A few high and low predicted glucose values influence the quality of the results a lot. We decided to remove those unnatural values by more likely results (lower boundary: 40, upper boundary: 400). This procedure is only included in the results of this paper, not in the submitted files.

#P	RF	GP	GP-OS	LR	GE	MOGE	A-0	A-1
60 minutes - RMSE								
540	44,06	37,13	39,97	38,87	41.16	40.94	47,26	57,40
544	28,08	28,45	28,77	28,40	33.46	29.64	35,61	45,63
552	27,24	26,08	25,91	28,90	31.04	29.85	27,18	34,39
567	37,76	35,99	35,82	36,19	39.68	37.82	47,53	51,16
584	38,11	37,84	34,63	37,12	38.17	37.84	41,05	48,03
596	29,58	27,56	27,12	27,77	30.31	28.65	33,33	42,36
Avg.	34,14	32,18	32,04	32,88	35.64	34.12	38,66	46,49
60 minutes - MAE								
#P	**RF**	**GP**	**GP-OS**	**LR**	**GE**	**MOGE**	**A-0**	**A-1**
540	31,62	27,83	30,33	29,65	32.01	31.76	31,71	39,35
544	20,37	20,13	20,35	21,17	26.77	22.50	23,38	29,96
552	20,47	19,78	19,51	22,42	23.56	23.08	15,28	19,56
567	27,65	26,06	25,87	27,14	30.26	28.50	30,60	35,11
584	29,18	27,45	26,09	27,74	29.08	28.82	26,24	32,83
596	21,70	20,26	20,07	20,89	22.82	21.27	21,44	21,44
Avg.	25,17	23,58	23,70	24,83	27.42	25.99	24,77	29,71
30 minutes - RMSE								
#P	**RF**	**GP**	**GP-OS**	**LR**	**GE**	**MOGE**	**A-0**	**A-1**
540	27,00	21,67	22,26	22,00	23.10	22.04	31,09	41,39
544	17,96	17,83	17,46	17,54	19.20	17.62	21,49	31,82
552	17,45	17,50	20,84	19,42	17.29	16.61	16,59	22,66
567	25,61	22,16	23,03	23,56	23.31	22.17	29,66	35,59
584	25,69	24,83	25,81	27,08	22.87	22.21	27,01	36,96
596	19,90	16,76	16,85	17,68	18.58	16.96	21,23	21,23
Avg.	22,27	20,13	21,04	21,22	20.73	19.60	24,51	31,61
30 minutes - MAE								
#P	**RF**	**GP**	**GP-OS**	**LR**	**GE**	**MOGE**	**A-0**	**A-1**
540	19,19	15,82	15,89	16,22	16.26	16.36	20,17	26,88
544	12,60	12,10	12,06	12,44	13.80	12.97	13,92	19,51
552	13,30	13,13	15,38	14,55	12.33	12.44	9,41	12,30
567	17,12	14,69	15,80	16,18	16.41	14.97	18,87	23,17
584	17,71	16,63	17,72	17,54	17.00	16.64	17,06	23,28
596	14,23	11,91	12,03	12,84	13.36	12.10	13,57	13,57
Avg.	15,69	14,05	14,81	14,96	14.86	14.25	15,50	19,79

Table 1. Quality of the models created for 30 / 60 minutes predictions. For each modeling method we give error metrics (RSME, MAE) for 30 / 60 minutes predictions.

Table 2 shows the percentage of predictions on zones of the Clarke Error Grid [6] for both time horizons. Results are ordered by higher %A, then higher %B, lower %E, lower %D and lower %C. The first thing that can be said is that, in terms of CEG, 30 minutes is not very hard to predict. Most of the algorithms achieved excellent results with less than 3% of the predictions in the dangerous zones. For a prediction horizon of 60 minutes, all the machine learning techniques obtained less than 5% of dangerous predictions, and GP approaches seems to be the best option. However, a deeper analysis for statistical significance in required. First, we depict in figure 1 a

Algorithm	%A	%B	%C	%D	%E
30 minutes					
GP	88.03	10.43	0.25	1.45	0.02
GP-OS	86.40	11.97	0.17	1.54	0.02
LR	86.03	12.23	0.25	1.65	0.02
RF	85.25	12.57	0.45	2.10	0.00
MOGE	87.52	11.29	0.00	1.19	0.00
GE	86.46	12.69	0.03	0.82	0.00
A-0	84,93	13,90	0,33	0,83	0.07
A-1	77.91	19.66	1.60	0.64	0.20
60 minutes					
GP	69.90	26.56	0.26	3.32	0.03
GP-OS	69.90	26.73	0.21	3.13	0.05
LR	67.35	28.76	0.35	3.60	0.03
RF	67.57	28.73	0.30	3.61	0.00
MOGE	64.27	31.15	0.29	4.31	0.00
GE	60.82	34.66	0.29	4.24	0.00
A-0	62.25	36.08	1.66	1.66	0.36
A-1	54.37	39.53	4.50	0.97	0.63

Table 2. Average percentage of predictions on zones of the Clarke Error Grid [6] for both time horizons. Results are ordered by higher %A, then higher %B, lower %E, lower %D and lower %C.

graphical ranking (in terms of RMSE) of all the algorithms for each patient and for 30 a 60 minutes prediction horizons. Each algorithm is represented by its acronym and a different color, the closer the position to the name of id of the patient, the better, i.e the lower RMSE on test files. GP is the best for all the patients in 30 minutes and for 4 out of 6 in 60 minutes. Looking for statistical significance, the first

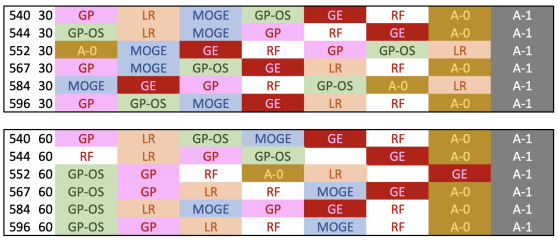

Figure 1. A graphical view of the ranking of each algorithm for each patient dataset. Clearly GP approach is the best as a general rule in terms of RMSE.

plots we created are density plots, using a kernel density estimation (KDE) of the distribution of the samples to visualize it. The objective is to visualize if the data meets the conditions for a parametric test, which is not the case. Figure 2 shows that the data is not distributed according to a Gaussian distribution and, nor the variance is the same for all the algorithms. Data distribution is multi-modal and a non-parametric test is necessary. All the plots were obtained with [4]. We use the graphical representation of the Nemenyi test [10], that compares all the algorithms pairwise. This non parametric test is based on the absolute difference of the average rankings of the predictors. For a significance level $\alpha = 0.05$ the test determines the critical difference (CD) and if the difference between the average ranking of two algorithms is grater than CD, then the null hypothe-

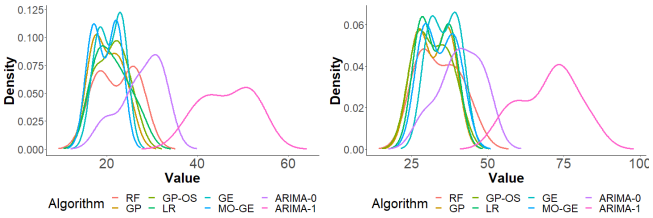

Figure 2. Density plots of the distribution of the RMSE results for all the algorithms for 30 minutes (left). The distribution are clearly multi-modal and a non parametric test is recommended. Similar plots were obtained for 60 minutes (right) and for MAE.

sis that the algorithms have the same performance is rejected. Figure 3 shows the graphical comparison where statistical differences are demonstrated to be significant. Finally we follow the Bayesian

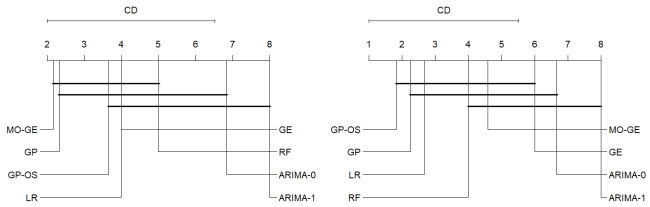

Figure 3. Nemenyi test for all the algorithms and RMSE (30 min, left, 60 min right) using the graphical representation of [10].

model of [3, 5] based on the Plackett-Luce distribution over rankings to analyse multiple algorithms in multiple problems. Figure 4 shows that GP and MOGE have the highest probability of being the best for 30 minutes, however there is not clear evidence for 60 minutes.

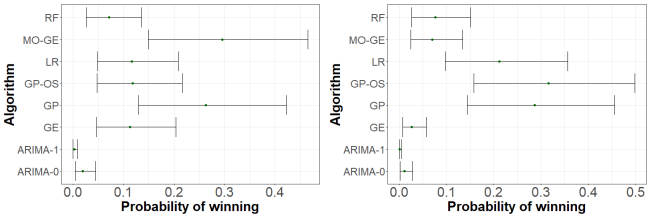

Figure 4. Bayesian model of [5] to analyse the algorithms in the set of patients and RMSE. Figure represents the probability of being the best and its standard deviation.(30 min, left, 60 min right)

4 CONCLUSION

The competition proved to be a very good test-bed for the modelling approaches as it is concerned with real-world data. The large amount of data for training proved to be challenging. For instance the ARIMA training process took several days to complete.

The decision made by the organizers of the competition to disallow usage of all future data is in our point of view not ideal. If we want to use prediction models for optimal blood glucose control it is necessary to assume that we can control the bolus and basal for

the forecasting horizon. Of course, a large amount of uncertainty remains because of unknown events in the forecasting horizon such as meals and higher activity or stress levels.

It would be interesting to try to improve the models by using all the available data for each data contributor. We only used the intersection of features available in all data sets which however limits the potential for specialization of models to individuals.

ACKNOWLEDGMENTS

This work has been also partially funded with the support of the Christian Doppler Research Association within the Josef Ressel Centre for Symbolic Regression. This work has been also partially supported by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU/AEI/FEDER, UE) under grant ref. PGC2018-095322-B-C22; and Comunidad de Madrid y Fondos Estructurales de la Unión Europea with grant ref. P2018/TCS-4566. UCM group is supported by Spanish Ministerio de Economía y Competitividad grant RTI2018-095180-B-I00, Fundación Eugenio Rodríguez Pascual, Comunidad de Madrid grants B2017/BMD3773 (GenObIA-CM) and Y2018/NMT-4668 (Micro-Stress - MAP-CM), and structural Funds of European Union.

REFERENCES

[1] Michael Affenzeller and Stefan Wagner, 'Offspring selection: A new self-adaptive selection scheme for genetic algorithms', in *Adaptive and Natural Computing Algorithms*, 218–221, Springer, (2005).

[2] Leo Breiman, 'Random forests', *Machine Learning*, **45**, 5–32, (2001).

[3] Borja Calvo, Josu Ceberio, and Jose A Lozano, 'Bayesian inference for algorithm ranking analysis', in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 324–325, (2018).

[4] Borja Calvo and Guzmán Santafé Rodrigo, 'scmamp: Statistical comparison of multiple algorithms in multiple problems', *The R Journal, Vol. 8/1, Aug. 2016*, (2016).

[5] Borja Calvo, Ofer M Shir, Josu Ceberio, Carola Doerr, Hao Wang, Thomas Bäck, and Jose A Lozano, 'Bayesian performance analysis for black-box optimization benchmarking', in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1789–1797, (2019).

[6] W.L. Clarke, D. Cox, L.A. Gonder Frederick, W. Carter, and S.L. Pohl, 'Evaluating clinical accuracy of systems for self-monitoring of blood glucose', *Diabetes Care*, **10**(5), 622–628, (September 1987).

[7] Sergio Contador, J Manuel Colmenar, Oscar Garnica, and J Ignacio Hidalgo, 'Short and medium term blood glucose prediction using multi-objective grammatical evolution', in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 494–509. Springer, (2020).

[8] Sergio Contador, J Ignacio Hidalgo, Oscar Garnica, J Manuel Velasco, and Juan Lanchares, 'Can clustering improve glucose forecasting with genetic programming models?', in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1829–1836, (2019).

[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan, 'A fast and elitist multiobjective genetic algorithm: Nsga-ii', *IEEE transactions on evolutionary computation*, **6**(2), 182–197, (2002).

[10] Janez Demšar, 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine learning research*, **7**(Jan), 1–30, (2006).

[11] J Ignacio Hidalgo, Marta Botella, J Manuel Velasco, Oscar Garnica, Carlos Cervigón, Remedios Martínez, Aranzazu Aramendi, Esther Maqueda, and Juan Lanchares, 'Glucose forecasting combining markov chain based enrichment of data, random grammatical evolution and bagging', *Applied Soft Computing*, **88**, 105923, (2020).

[12] J Ignacio Hidalgo, J Manuel Colmenar, Gabriel Kronberger, Stephan M Winkler, Oscar Garnica, and Juan Lanchares, 'Data based prediction of blood glucose concentrations using evolutionary methods', *Journal of medical systems*, **41**(9), 142, (2017).

[13] J Ignacio Hidalgo, J Manuel Colmenar, J Manuel Velasco, Gabriel Kronberger, Stephan M Winkler, Oscar Garnica, and Juan Lanchares, 'Identification of models for glucose blood values in diabetics by grammatical evolution', in *Handbook of Grammatical Evolution*, 367–393, Springer, (2018).

[14] John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.

[15] Cindy Marling and Razvan Bunescu, 'The ohiot1dm dataset for blood glucose level prediction: Update 2020', (2020).

[16] Michael O'Neill and Conor Ryan, 'Grammatical evolution', *IEEE Trans. Evolutionary Computation*, **5**(4), 349–358, (2001).

[17] Jose Manuel Velasco, Oscar Garnica, Juan Lanchares, Marta Botella, and J Ignacio Hidalgo, 'Combining data augmentation, edas and grammatical evolution for blood glucose forecasting', *Memetic Computing*, **10**(3), 267–277, (2018).

[18] G.Peter Zhang, 'Time series forecasting using a hybrid arima and neural network model', *Neurocomputing*, **50**, 159 – 175, (2003).