Forecasting Glucose Levels in Patients with Diabetes Mellitus using Semantic Grammatical Evolution and Symbolic Aggregate Approximation

Jose Manuel Velasco
Universidad Complutense de Madrid
mvelascc@ucm.es

Oscar Garnica
Universidad Complutense de Madrid
ogarnica@ucm.es

Sergio Contador
Universidad Complutense de Madrid
scontador@ucm.es

Marta Botella
Hospital Hospital U. Principe Asturias, Alcala de Henares
marta.botella@saludmadrid.org

Juan Lanchares
Universidad Complutense de Madrid
julandan@ucm.es

J. Ignacio Hidalgo
Universidad Complutense de Madrid
hidalgo@ucm.es

ABSTRACT

Type 1 Diabetes Mellitus can only be treated injecting insulin and glucagon into the blood stream. This research is motivated by the challenge to accurately predict future blood glucose levels of a diabetic patient so that an automatic system could decide when is necessary the injection of a bolus of insulin to keep blood sugar in the healthy range.

In this paper, we have studied different evolutionary strategies based on geometric semantic genetic programming and grammatical evolution. The main contribution of this paper is the use of the symbolic aggregate approximation representation of the glucose time series that allow us to define easily semantic operators. We have developed a new strategy that combines grammatical evolution with the geometric semantic approach and that, thanks to the use of the symbolic representation, improves the previous models of glucose time series. We also present a variation of this technique that employs a univariate marginal distribution algorithm to tune the parameters of the symbolic representation. The experimental results are compared against classical grammatical evolution and geometric semantic hill climbing genetic programming. The baseline is provided by the conventional ARIMA model.

Our experimental results show that the symbolic representation improves the performance of the geometric semantic strategy and reduces the number of mistakes that, if in an automatic system, would put patient's health at risk.

CCS CONCEPTS

•**Computing methodologies** → **Genetic programming;** *Optimization algorithms;* •**Applied computing** → *Health informatics;*

ACM Reference format:
Jose Manuel Velasco, Oscar Garnica, Sergio Contador, Marta Botella, Juan Lanchares, and J. Ignacio Hidalgo. 2017. Forecasting Glucose Levels in Patients with Diabetes Mellitus using Semantic Grammatical Evolution and Symbolic Aggregate Approximation. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082493

GECCO '17 Companion, Berlin, Germany

978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3082493

1 INTRODUCTION

The pancreas is a gland of the endocrine system that produces insulin and glucagon. These two hormones keep blood glucose levels within optimal ranges working together to regulate the carbohydrate metabolism. There is a group of metabolic diseases associated with an inability of the body to regulate this metabolism, and that can produce high blood glucose levels over a prolonged period. These diseases are called Diabetes, generally speaking, and in 2015, it was estimated that more than 415 million people worldwide suffer from it. Type 1 Diabetes mellitus (T1DM) represents 10% of this number. T1DM is a chronic autoimmune disorder in which the immune system attacks the insulin-secreting cells of the pancreas. The result is that cells do not assimilate sugar and, as a consequence, there is a rise in blood glucose levels (or hyperglycemia). If this situation extends for a long period of time, the patient can develop serious long-term complications including heart diseases, blindness, kidney failure or foot ulcers. T1DM can only be treated with synthetic insulin injected into the blood stream. However, this is not an easy task. An excessive dose of insulin can produce hypoglycemia (very low blood sugar). If hypoglycemia is very severe, it can lead to unconsciousness or even a comma. In Figure 1 we can see the evolution of blood glucose levels in a diabetic patient. As we can see, the episodes of hyperglycemia and, even, hypoglycemia are very frequent. The ideal solution for T1DM would be an artificial pancreas (AP) capable of maintaining good control of the levels of sugar in the blood and allowing the patient to have a normal life while at the same time avoiding (or at least delaying) the appearance of complications. AP is the main area of research in the field. One of the main problems for the development of the AP is the lack of accurate models for predicting the future of the glucose. Although there are some classical approximations, there is still too much to do for predictions within a horizon of more than 90 minutes.

This research is motivated by the challenge to predict accurately future glucose levels so that an automatic system can decide when and how much insulin to inject in order to keep blood sugar in the healthy range. At the same time, it's very important, for a correct

blood glucose control system, to avoid predictions that can trigger unnecessary treatments or, even worse, treatments that go against patient's needs. For achieving this task, we have studied different evolutionary strategies based on geometric semantic genetic programming and grammatical evolution. The main contribution of this paper is the use of the symbolic aggregate approximation representation of the glucose time series. As we will see later, this representation allow us to define geometric semantic operators that facilitate the search of suitable functions for forecasting. Our experimental results show that the main problem of geometric semantic genetic programming, when applied to time series modeling, seems to be overfitting. Another advantage of the use of symbolic aggregate approximation is that we can tune its parameters to avoid overfitting. For doing so, we employ another evolutionary technique: the univariate marginal estimation distribution algorithm.

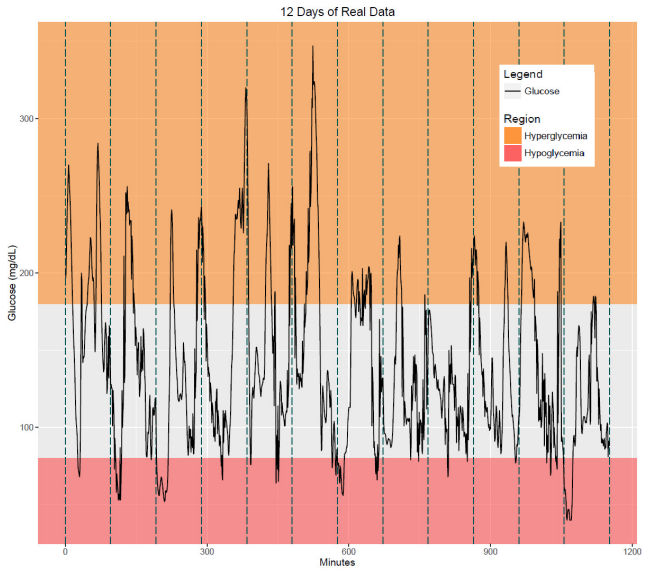

Figure 1: Real Glucose Data - 12 Days.

The rest of the paper is organized as follows. Section 2 describes the related work whereas Section 3 introduces the theoretical background of the techniques. Section 4 explains our approach. The experimental setup and results are showed in Section 5. Conclusions and future work are exposed in Section 6.

2 RELATED WORK

To our best knowledge, Semantic Genetic Programming has not been tested yet in the field of medical time series data. In [17], McDermott et al. applied Geometric Semantic Genetic Programming to financial time series to get an automatic trading strategy. Their paper explores the benefits of two different semantic operators. Our work is complementary to theirs, as it is based on a different way of representing time series data: the Symbolic Aggregate Approximation (Section 3.4).

The problem of predicting and modeling glucose levels has been an intensive area of research during the last ten years. Two are the main targets of these studies. Some of them tried to predict the glucose levels with a time horizon of up to two hours since this is usually the time step needed by the patient to be comfortable after a meal. There are also some researchers that tried to identify 24 hours models. The utility of the last is different and is usually more effective when programming an insulin pump or when establishing an insulin profile for longer periods. We can find in literature some approximations providing models for the average case [16]. However, there are hardly few approaches adapted to the particularities of each patient. Most of the articles in the literature apply classical modeling techniques, resulting in models or profiles defined by linear equations with a limited set of inputs [11, 19]. Recently Hidalgo et al. proposed the application of Grammatical Evolution to obtain customized models of patients, unlike most of the previous approaches which get averaged models. The proposal has been tested with in-silico patient data and results are clearly positive. Authors also present a study of four different grammars and five objective functions [8]. Our study extends this research as it employs real data from a diabetic patient and, in addition, we introduce several improvements to the Grammatical Evolution approach.

Other personalized control approaches were presented by the main research groups on AP [4, 9, 12, 15]. Those are proposals following the clinical practice. Treatment for subjects with T1DM uses rates of basal insulin delivery, insulin to carbohydrate ratios and individual correction factors, typically from observations of the specialist. However, those models are often inaccurate, since clinical data in T1DM are not extensive enough to identify the exact models [29]. There are also some models used in artificial pancreas systems or closed loop control models trying to emulate the action of a pancreas [3, 27]. They are based on the assumption that it is possible to reach a good control with approximate models, provided that the model is related to the control objective [6]. Our experimental results suggest that in this approach and due to the lack of accurate individualized models, there is a significant risk of an excessive insulin administration and therefore, the possibility that blood glucose levels fall down to hypoglycemia zone. Our evolutionary models try to avoid this situation.

3 BACKGROUND

3.1 Problem Description

Patients can inject themselves two types of insulin, IS (short-term insulin) and IL (long-term insulin), and their doses depend on an estimation of the amount of carbohydrates they ingest, and their actual values of blood glucose. The problem of modeling the glucose blood level of a patient can be formulated as follows: to find an expression of estimated glucose values based on previous values of glucose, carbohydrates, and insulin. In its most simplified form, it is described in Equation 1, where predicted glucose is denoted as $\widehat{GL}$, GL corresponds to previous glucose values, CH corresponds to previously ingested carbohydrates and IS and IL represent the previously injected insulin for both types, short and long effect. It should be noted that the model will provide estimated glucose values in a forecasting horizon H obtained by using the N previous samples of glucose values, carbohydrates and insulin units. In this paper, $N = 8$ which corresponds to two hours of recorded data. Table 1 shows a reduced version of our data set.

k	GL	CH	IS	IL
...	...	...	...	...
30	170	0	0	0
31	171	0	0	0
32	172	0	3	12
33	173	30	0	0
34	174	0	0	0
...	...	...	...	...
40	237	0	0	0
41	247	20	0	0
42	250	0	0	0
43	251	0	0	0
...	...	...	...	...

Table 1: Portion of a 24-hours dataset for a patient.

```
# Model expression
<func> ::= <exprgluc> + <exprch> - <exprins>

# Glucose
<exprgluc> ::= (<exprgluc> <op> <exprgluc>) |
  <preop> (<exprgluc>) | (<cte> <op> <exprgluc>) |
  realData(t-<idx>)

# CH
<exprch> ::= (<exprch> <op> <exprch>)
         |<preop> (<exprch>)
         |(<cte> <op> <exprch>)
         |(getPrevData(1,t,1) * <cte> * <curvedCH>)

# Insulin:
<exprins> ::= (<exprins> <op> <exprins>)
         |<preop> (<exprins>)
         |(<cte> <op> <exprins>)
         |getVariable(2,t-<idx>)

<op> ::= +|-|*|/
<preop> ::= exp|sin|cos|log
<cte> ::= <dgtNoZero><dgtNoZero>.<dgt><dgt>
<idx> ::= <dgtNoZero>|<dgtNoZero><dgt>|<dgtNoZero><dgt><dgt>
<dgtNoZero> ::= 1|2|3|4|5|6|7|8|9
<dgt> ::= 0|1|2|3|4|5|6|7|8|9
```

Figure 2: Grammar for Glycemic Modeling.

$$\widehat{GL}(t+H) = f(GL(t-k), CH(t-k), IS(t-k), IL(t-k)), 1 \leq k \leq N \quad (1)$$

3.2 Grammatical Evolution

To get a prediction of a patient's glucose level in the future, we deal with a kind of Symbolic Regression (SR) problem. SR tries to obtain a mathematical expression to reproduce a set of discrete data. Genetic Programming (GP)[13] has proven effective in some SR problems but it also has limitations. During last years, variants to GP like Grammatical Evolution (GE)[24] appeared to propose different evaluation approaches. GE is an evolutionary computation technique pioneered by C. Ryan, J.J. Collins and M. O'Neill in 1998. In contrast to genetic algorithms, which work with a representation of solutions, GE works (evolves) with a genetic code that determines the production process of the solutions. The code translation process is defined by grammars represented in Backus-Naur Form (BNF) which is a notation for expressing context-free grammars. This way, GE allows generation of computer programs, that is, symbolic expressions in an arbitrary language using grammars to specify the rules for obtaining the programs. In the definition of the grammars and due to its flexibility, we can insert up to a point our knowledge of the glucose-insulin interaction.

Figure 2 represents an extract of a grammar, in BNF, designed for finding a forecasting model of future glucose levels. The code that represents an expression will consist of elements of the set of terminals. These are combined according to the rules of the grammar. Besides, grammar can be adapted to bias the search of the evolutionary process because there is a finite number of options for each production rule, which limits the search space. So, in this case, we have restricted to two hours the previous data that can be employed in the model and the forecasting horizon is one hour. Next, we detail the most important rules of our grammar. We search for an expression based on glucose (<exprgluc>), plus some expression regarding carbohydrates (<exprch>), minus an expression of insulin (<exprins>). The expression of glucose denoted by exprgluc is a recursive rule that may produce a complex formula using arithmetic operators (<op>), functions (<preop>) and constant values (<cte>) which, in our case, are generated through a base and an exponent built with integer values.

3.3 Geometric Semantic Genetic Programming

Both Grammatical Evolution and Genetic Programming are techniques that transform programs at a syntactic level without being aware of the behavior of the programs. This produces that, generally speaking, the parents and offspring behaviors are quite unrelated, because the distance between the fitness values of parents and offspring can be great; that is, the fitness landscape is very bumpy. Because of this, many researchers consider GP and GE blind optimization techniques. This situation was pointed out in [25] and lead to a new research field that is currently exploring the addition of semantic knowledge (program behavior) to strategies based on GP. In the classical Koza's GP [13], programs are represented by trees, and new individuals are created swapping and mutating subtrees amongst previous individuals. The idea behind Geometric Semantic Genetic Programming (GSGP) [21] and the techniques that have been derived from it [28] is to find semantic crossover and mutation operators that guarantee that the distance between the fitness values of parents and offspring is bounded. This way, the fitness landscape (as it is seen by the new operators) transforms itself into a cone and the optimization process becomes a hill-climbing algorithm (GSHCGP). Nevertheless, nothing comes without a cost and the semantic operators usually produce extraordinary large trees and the evolutionary process needs a great amount of memory resources. This limitation prohibits the application of GSGP into embedded devices with limited memory.

3.4 Symbolic Aggregate Approximation

Symbolic Aggregate Approximation (SAX) [14], translates the time series data into a symbolic representation. It has two main characteristics that are useful for GSGP:

- It uses Piecewise Aggregate Approximation (PAA)[14] to produce a reduction in the dimensionality of the time series.

- The distance function defined into the symbolic representation is lower bounded regarding a distance function in the corresponding original series. This feature allows us to develop semantic aware operators easily.

The PAA technique reduces the time series from N dimensions to M dimensions dividing the time span into windows of N/M size. Within each window, we calculate the mean value of the different samples that lay into it. These mean values give us a new time series which is the PAA representation.

After obtaining the PPA approximation, we get the distribution of the time series. Providing it is a Gaussian distribution, the SAX technique translates the PAA values to SAX symbols using a predefined table of breakpoints obtained from Gaussian lookup tables to translate the PAA values into symbols. We have also experimented with the approach presented in [5]. These authors found the breakpoints using a genetic algorithm, and they called their technique GASAX. In this paper, we have employed an UMDA strategy (see Section 3.6) to find the best breakpoints dynamically and we call it USAX. In Figure 3, we present the application of USAX to glucose time series. If we reduce the first 32 glucose samples (marked with a vertical dashed line) to eight dimensions with four symbols, the glucose time series is translated into the word dddccbaa.

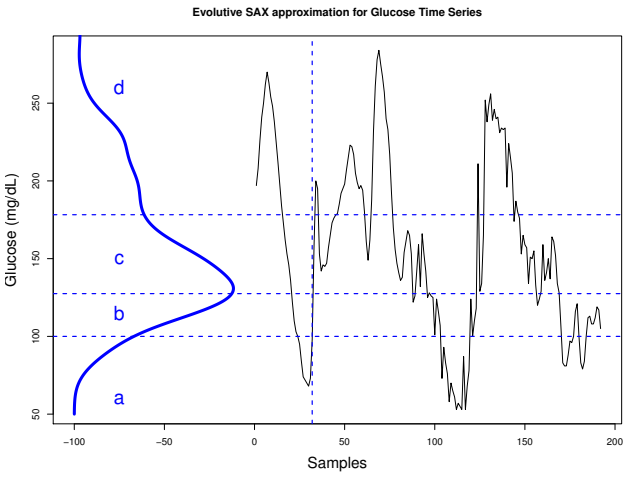

Figure 3: Glucose Time Series into SAX Representation with four symbols using UMDA. The blue line is the full time series distribution.

3.5 ARIMA

Following the ideas presented in [18], we have measured the performance of the evolutionary strategies against non-seasonal Auto Regressive Integrated Moving Average (ARIMA) model [1]. ARIMA is a well established standard technique for modeling and forecasting time series. In Equation 2 we can see that the model can be segregated into two parts. The first part is a linear regression of y_t based on the observations during period p, that is to say, is an autoregression. The other part is a linear combination of the current error term and the q most recent past error terms during period p. Integrated means that the drift has been removed (if present) by differencing the time series.

$$y_t = \overbrace{\underbrace{c + \phi_1 \dot{y}_{t-1} + \cdots + \phi_p \dot{y}_{t-p}}_{\text{Autoregresive Part}} + \underbrace{\theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t}_{\text{Moving Average}}}^{\text{ARIMA}} \quad (2)$$

3.6 Univariate Marginal Distribution Algorithm

The Univariate Marginal Distribution Algorithm (UMDA) [22, 26] is a stochastic optimization method that belongs to a subclass of evolutionary strategies called Estimation of Distribution Algorithms (EDAs). In the EDAs, the usual genetic operators for creating new individuals (crossover and mutation) are not used. Instead, they create a new generation of candidates, sampling a probabilistic distribution which is estimated from the best individuals of the previous generation. The UMDA algorithm is one of the simplest forms of an EDA, and the estimation of the distribution is achieved using the univariate marginal probability, that is, the frequency of each component in the population.

3.7 Error Grid Analysis

For Diabetic patients, the forecasting mistakes can have an entirely different impact depending on whether the actual blood glucose level is in the hypoglycemic, hyperglycemic or in-between zone. For instance, let's suppose that our prediction points up erroneously that a patient's future glucose levels are going to be in the hyperglycemic zone, but the actual blood glucose falls under the hypoglycemic threshold. In this case, the treatment for the hyperglycemic zone will get the patient deeper into the hypoglycemic zone provoking a very dangerous situation which must be avoided at all cost. To take into account this kind of situations we use the method that was presented in 1987 by Clarke et al. [2], known as the Error Grid Analysis (EGA). The EGA method was proposed to quantify the patient's estimates versus the values given by a blood glucose monitoring device, but it has been used since then as a way to standardize the behavior of glucose meters. In this study, we use it as a way of measuring the accuracy of the predictions of our methods.

Following the EGA method, we draw a scatterplot of the experimental results. In one axis, we have the real observations and on the other, the values obtained through a forecasting method. The main diagonal represents the perfect prediction, and depending on the severity of the misprediction, the rest of the points can fall into five regions:

- Region A is those values within 20% of the actual values,
- Region B contains points that are outside of 20% but would not lead to inappropriate treatment,
- Region C is those points leading to unnecessary treatment,
- Region D is those points indicating a potentially dangerous failure to detect hypoglycemia or hyperglycemia, and
- Region E is those points that would confuse treatment of hypoglycemia for hyperglycemia and vice versa.

Therefore, the lesser points that appear in the C zone, the better, being very critical to avoid the E and D zones.

4 METHODOLOGY

In Figure 4 we can see a flow-chart picture summarizing the way we generate and optimize our model using as a background the original structure of GSHCGP.

(1) As a previous step, we get the SAX representation (a string of symbols) of the real data training set (top-right in the figure) which is our fitness goal.
(2) Then, we begin with the evolutive process. Our grammar produces a function (top-left in the figure). From this function, we get a time series that is translated to the SAX approximation.
(3) We, then, calculate the fitness of this solution by comparing the two SAX strings. If the fitness is better than our current solution, the new function becomes the new model. It is important to remark that for GSHCGP the fitness is calculated based on the Euclidean distance.
(4) We get the offspring of the current model through semantic mutation:
    - Our GE produces a new function and a scale factor between $(-1, 1)$. This is another difference respect to GSHCGP.
    - The result of the new function and the scale is added up to the current model.
(5) From the new function, we get a new time series that is translated into a string of symbols.
(6) A new fitness is calculated, and the offspring is either discarded or become the current tentative model.
(7) If the fitness is at its highest value or if we have consumed the maximum amount of generations, we end up validating the final model with the real data validation set.

If we run this process with a set of fixed SAX breakpoints, we have our SAX-SGE model. We have also experimented with a dynamic tuning of the SAX breakpoints. During this training phase, we get these parameters using the UMDA algorithm (see Section 3.6) implemented in the library [7]. In this case, we evolve a population of solutions, and the UMDA algorithm employs the best solutions to estimate the breakpoints in the distribution that, afterwards, the SAX algorithm uses to translate the time series data into symbols. We call this option the USAX-SGE model.

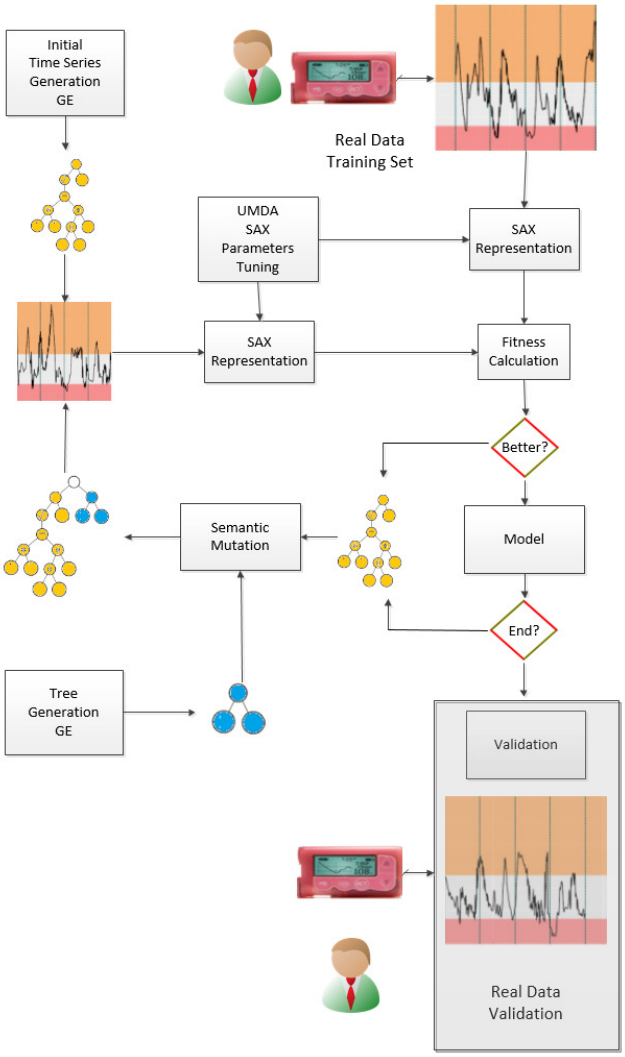

Figure 4: SAX-SGE Model generation and optimization.

5 EXPERIMENTS

5.1 Experimental Setup

Thanks to the staff at the Principe de Asturias Hospital at Alcala de Henares, Spain, we have been able to collect data from a real patient with a continuous glucose monitoring system (CGMS) during twelve days [1]. We have observations every fifteen minutes up to a total of 1152 measures. We also have recorded carbohydrate units ingested and insulin injected, distinguished by insulin type, for every day.

Our twelve days of real data (Section 3.1) have been segregate into two sections of six days. We have used the first section for training the models, and the last section has been targeted for validating the three new techniques and the two used as baseline in this study.

[1] On 6 June 2012, the Clinical Research Ethics Committee of the Hospital of Alcalá de Henares (Spain) authorized the use of the data collected, provided that the privacy of the data is ensured and the informed consent of patients is made.

The models were trained rolling a window of two hours (eight samples) through the data to make a prediction with the corresponding horizon (30, 60, 90 and 120 minutes). We run the experiment ten times and we show here the average of the ten runs.

The fitness function for the GE and GSCHGP models were the mean squared error (MSE), that is to say, the average of the squares of the Euclidean distance between the time series of the candidate and the actual data. For the SAX-SGE models, the fitness function is the number of identical symbols between the SAX approximation of the actual and candidate time series.

The value of the parameters of the evolutionary algorithms used in the training phase was selected after a set of preliminary experiments. In these preliminary experiments we did a systematic sweep of values ranging from 50% to 200% of the values shown on Table 2. We must remember that:

- The Semantic (with Hill Climbing) models do not use CrossOver nor Selection and the number of generations has the meaning of number of steps. The Depth parameter is the maximum number of levels of the new tree that is used in the mutation phase.
- The USAX-SGE strategy employs two evolutionary algorithms: the SAX Semantic Grammatical Evolution and the UMDA.

Table 2: Experimental parameters.

Grammatical Evolution		**UMDA**	
Parameter	Value	Parameter	Value
Population	200	Population	200
Generations	2000	Generations	1000
Selection	25%	Selection	25%
CrossOver	70%		
Mutation	20%		
GSHCGP		**SAX-SGE**	
Parameter	Value	Parameter	Value
Generations	2000	Generations	2000
Depth	4	Depth	4
		Compression rate	8
		Symbols	16

5.2 Experimental Results

Figure 5 shows the results of all models within a Clarke Error Grid, and Table 3 summarizes the results of the five models. The legend for both is:

- ARIMA: The classical ARIMA model that we use as a reference to compare against the evolutionary strategies (see Section 3.5). We have employed the library from [10].
- GE: Grammatical Evolution Strategy. We have used the library from [23] and the grammar presented in Section 3.2.
- GSHCGP: The Geometric Semantic Hill Climbing Genetic Programming model. We have modified the code from [20] to implement the forecasting function based on the lagged glucose time series as in the GE model. This version of GSGP only uses mutation to generate the offspring. The fitness is calculated using the Euclidean distance between the time series.
- SAX-SGE and USAX-SGE: These are our proposal, and they are explained in Section 4.
- MAPE is the Mean Absolute Percentage Error for both Training (T) and Validation (V) phases.

Figure 5 is a scatterplot of real values (horizontal axis) and predicted values (vertical axis) during the second section of our real data. For clarity, we show only the worst predictions for each strategy (that is, the predictions with a higher mean square error). The

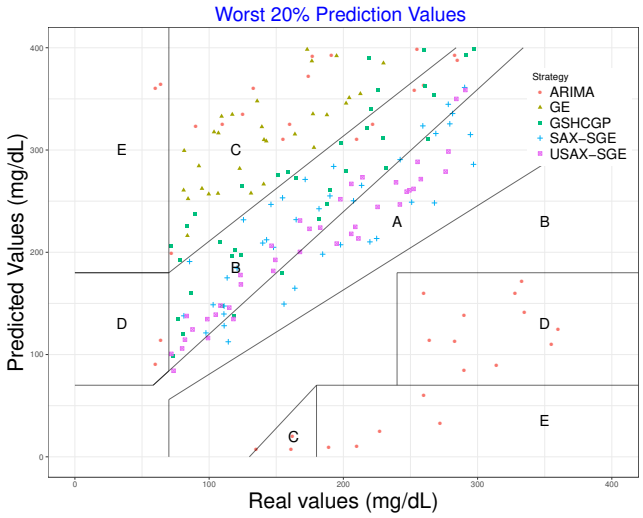

Figure 5: Clarke Error Grid results for the worst 20% of predictions. Forecasting Horizon = 60 min.

forecasting horizon is one hour. For bigger horizons, the predictions get worse, but here, we can see that all the evolutionary strategies stay within the limits of the A, B and C zones. Whereas the ARIMA model has its points scattered across the C, D and E zones. So, several conclusions arise from this figure and Table 3:

- Prediction Horizon equals to 30 minutes:
  - All the strategies have similar performance with no points in the D and E zones.
  - The GE model has the biggest percentage of points in the C zone and, therefore, qualifies as the worst option in this time frame.
  - The results for both SAX-based models are slightly better than for the rest of models, being the best strategy the SAX-SGE although we cannot state that this is noteworthy.
- Prediction Horizon equals to 60 minutes:
  - Here, the ARIMA model has near a ten percent of prediction in the D and E zones whereas the rest of models get their forecastings out of those zones.
  - Although the GE model has no points in the D and E zones, it has a lot of predictions into the C zone. We must remember that predictions into the C zone can lead to unnecessary treatments and therefore this situation is not admissible. Enhancing the GE behavior is one of the motivations of this research work.
  - The best strategy in this time frame is the geometric semantic genetic programming. The GSHCGP model has clearly the minimum percentage error.
- Prediction Horizon equals to 90 and 120 minutes:
  - With these larger horizons, all the techniques get bigger MAPEs, but the semantic strategies keep a low percentage of predictions into the most dangerous zones.
  - The GSHCGP model achieves great results but it seems to suffer from overfitting. It always gets the

best results during the training phase but the SAX models overtake it during the validation phase. Therefore, the advantage of using the SAX representation is that it can avoid the overfitting. As we can see, the SAX models are capable of better results during the validation phase.

– In these horizons the strategy that gets more robust results (lesser points in the C, D and E zones and lower MAPE-V values) is the USAX-SGE.

Summarizing, for the short-term forecasting horizons, the performance of the semantic algorithms seem to suggest that they could be good candidates for an automatic system. Nevertheless, for the further forecasting horizons, all these techniques incur in a considerable amount of hazardous predictions. The SAX and USAX results for the distant horizons during the validation phase, although promising, are still not good enough.

Table 3: Clarke Zones for Predicted Values.

Strategy	Horizon = 30 min					MAPE-T	MAPE-V
	Zones Percentage						
	A	B	C	D	E		
ARIMA	47.59	50.64	1.74	0.00	0.00	3.3	6.23
GE	45.88	47.18	6.92	0.00	0.00	4.1	7.1
GSHCGP	57.44	41.12	1.43	0.00	0.00	2.1	5.45
SAX-SGE	62.49	37.2	0.31	0.00	0.00	4.3	5.3
USAX-SGE	59.25	39.92	0.83	0.00	0.00	5.7	6.33
Strategy	Horizon = 60 min					**MAPE-T**	**MAPE-V**
	Zones Percentage						
	A	B	C	D	E		
ARIMA	15.74	27.81	47.23	5.45	3.77	8.7	14.4
GE	16.56	44.46	38.98	0.00	0.00	11.72	18.52
GSHCGP	61.19	35.78	3.03	0.00	0.00	5.08	7.77
SAX-SGE	47.11	38.89	14.01	0.00	0.00	7.79	8.98
USAX-SGE	46.43	41.20	12.37	0.00	0.00	8.39	10.72
Strategy	Horizon = 90 min					**MAPE-T**	**MAPE-V**
	Zones Percentage						
	A	B	C	D	E		
ARIMA	15.7	45.23	21.47	9.2	8.4	18.4	21.4
GE	19.26	40.28	33.03	6.88	0.56	12.20	19.07
GSHCGP	49.94	31.96	14.83	1.49	1.78	10.63	20.33
SAX-SGE	52.14	33.17	8.84	4.41	1.45	11.43	17.70
USAX-SGE	449.51	33.89	12.30	3.82	0.48	11.96	15.74
Strategy	Horizon = 120 min					**MAPE-T**	**MAPE-V**
	Zones Percentage						
	A	B	C	D	E		
ARIMA	14.37	37.23	26.40	10.70	11.30	23.14	31.10
GE	15.45	35.81	40.30	3.78	4.66	17.31	23.89
GSHCGP	39.53	36.07	15.89	4.19	3.32	12.52	21.85
SAX-SGE	58.98	33.05	5.05	3.85	3.15	18.71	19.13
USAX-SGE	42.58	33.54	18.97	2.89	2.01	16.33	18.13

6 CONCLUSIONS AND FUTURE WORK

This research is motivated by the challenge to predict accurately future blood glucose levels of a diabetic patient so that an automatic system can decide when to inject a bolus of insulin and its doses to keep blood glucose in the healthy range. It is also crucial, for a correct blood glucose control system, to avoid predictions that can trigger unnecessary treatments or, even worse, treatments that go against patient's needs.

In this paper, we have studied different evolutionary strategies based on Geometric Semantic Genetic Programming and Grammatical Evolution.

The main contribution of this paper is the use of the Symbolic Aggregate Approximation representation of the glucose time series. As we have seen in the experimental results, the use of this representation has reduced the mean absolute percentage error of the predictions and the number of mistakes that, if in an automatic system, would put patient's health at risk. Besides, in this paper, we have studied an evolutionary technique for tuning the parameters of the symbolic approximation which has improved the general performance of the semantic strategy.

As future work, a lot of problems remain open:

- The evolutive process of the semantic strategies needs a lot of memory resources due to the increment in the size of the tentative solution in every step. In its current state, the Geometric Semantic Genetic Programming technique and its variants (as the ones studied in this paper) are not suitable for their future implementation into an embedded device with constraint resources. So, we need to develop techniques that mitigate this problem.
- For the forecasting horizons of 90 and 120 minutes, the results of the semantic strategies, although promising, are still not good enough for their inclusion into an automatic Artificial Pancreas.

7 ACKNOWLEDGMENT

This research is supported by the Spanish Minister of Science and Innovation (TIN2014-54806-R). The authors would like to thank the staff in the Principe de Asturias Hospital at Alcala de Henares for their support and assistance with this project. Special thanks also go to Maria Aranzazu Aramendi Zurimendi and Remedios Martinez Rodriguez.

REFERENCES

[1] George Edward Pelham Box and Gwilym M. Jenkins. 1994. *Time Series Analysis: Forecasting and Control* (3rd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[2] W.L. Clarke, D. Cox, L.A. Gonder-Frederick, W. Carter, and S.L. Pohl. 1987. Evaluating clinical accuracy of systems for self-monitoring of blood glucose. *Diabetes Care* 10, 5 (Sept. 1987), 622–628. DOI: http://dx.doi.org/https://doi.org/10.2337/diacare.10.5.622

[3] E. Dassau, H. Zisser, B. Grosman, W. Bevier, M.W. Percival, L. Jovanovic, and F.J. Doyle. 2009. Artificial pancreatic betta-cell protocol for enhanced model identification. *Diabetes* (2009), A105–A106.

[4] F.H. El-Khatib, S.J. Russell, D.M. Nathan, R.G. Sutherlin, and E.R. Damiano. 2010. A bihormonal closed-loop artificial pancreas for type 1 diabetes. *Sci Transl Med.* 2 (27) (2010).

[5] Muhammad Marwan Muhammad Fuad. 2012. Genetic Algorithms-based Symbolic Aggregate Approximation. In *Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'12)*. Springer-Verlag, Berlin, Heidelberg, 105–116. DOI: http://dx.doi.org/10.1007/978-3-642-32584-7_9

[6] Michel Gevers. 2005. Identification for Control: From the Early Achievements to the Revival of Experiment Design*. *European Journal of Control* 11, 4 (2005), 335 – 352. DOI: http://dx.doi.org/10.3166/ejc.11.335-352

[7] Yasser Gonzalez-Fernandez and Marta Soto. 2014. copulaedas: An R Package for Estimation of Distribution Algorithms Based on Copulas. *Journal of Statistical Software* 58, 9 (2014), 1–34. http://www.jstatsoft.org/v58/i09/

[8] J. Ignacio Hidalgo, Esther Maqueda, José L. Risco-Martín, Alfredo Cuesta-Infante, J. Manuel Colmenar, and Javier Nobel. 2014. glUCModel: A monitoring and modeling system for chronic diseases applied to diabetes. *Journal of Biomedical Informatics* 48 (2014), 183 – 192. DOI: http://dx.doi.org/10.1016/j.jbi.2013.12.015

[9] Roman Hovorka, Janet M Allen, Daniela Elleri, Ludovic J Chassin, Julie Harris, Dongyuan Xing, Craig Kollman, Tomas Hovorka, Anne Mette F Larsen, Marianna Nodale, Alessandra De Palma, Malgorzata E Wilinska, Carlo L Acerini, and David B Dunger. 2010. Manual closed-loop insulin delivery in children and adolescents with type 1 diabetes: a phase 2 randomised crossover trial. *The Lancet* 375 (2010), 743 – 751.

[10] Rob J Hyndman and Yeasmin Khandakar. 2008. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software* 26, 3 (2008), 1–22. http://www.jstatsoft.org/article/view/v027i03

[11] Christina-Maria Kastorini, George Papadakis, Haralampos J. Milionis, Kallirroi Kalantzi, Paolo-Emilio Puddu, Vassilios Nikolaou, Konstantinos N. Vemmos, John A. Goudevenos, and Demosthenes B. Panagiotakos. 2013. Comparative analysis of a-priori and a-posteriori dietary patterns using state-of-the-art classification algorithms: A case/case-control study. *Artificial Intelligence in Medicine* 59, 3 (2013), 175 – 183. DOI: http://dx.doi.org/10.1016/j.artmed.2013.08.005

[12] B. Kovatchev, C. Cobelli, E. Renard, S. Anderson, M. Breton, S. Patek, W. Clarke, D. Bruttomesso, A. Maran, S. Costa, A. Avogaro, C. Dalla Man, A. Facchinetti, L. Magni, G. De Nicolao, J. Place, and A. Farret. 2010. Multinational study of subcutaneous model-predictive closed loop control in type 1 diabetes mellitus: Summary of the results. *Diabetes Sci Technol* 4 (2010), 1374–1381.

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

[14] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery* 15, 2 (2007), 107–144. DOI: http://dx.doi.org/10.1007/s10618-007-0064-z

[15] L. Magni, Marco Forgione, M. Forgione, Chiara Toffanin, Chiara Toffanin C FAU Dalla Man, Boris Dalla Man C FAU Kovatchev, Giuseppe Kovatchev B FAU De Nicolao, Claudio De Nicolao G FAU Cobelli, and Cobelli C. 2009. Run-to-run tuning of model predictive control for type 1 diabetes subjects: in silico trial. 1932-2968 (Linking) (2009), –.

[16] Lucy Mays. 2015. Diabetes Mellitus Standards of Care. *Nursing Clinics of North America* 50, 4 (2015), 703 – 711. DOI: http://dx.doi.org/10.1016/j.cnur.2015.08.001 Pathophysiology and Care Protocols for Nursing Management.

[17] James McDermott, Alexandros Agapitos, Anthony Brabazon, and Michael O'Neill. 2014. *Geometric Semantic Genetic Programming for Financial Data*. Springer Berlin Heidelberg, Berlin, Heidelberg, 215–226. DOI: http://dx.doi.org/10.1007/978-3-662-45523-4_18

[18] James McDermott, David R. White, Sean Luke, Luca Manzoni, Mauro Castelli, Leonardo Vanneschi, Wojciech Jaskowski, Krzysztof Krawiec, Robin Harper, Kenneth De Jong, and Una-May O'Reilly. 2012. Genetic Programming Needs Better Benchmarks. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO '12)*. ACM, New York, NY, USA, 791–798. DOI: http://dx.doi.org/10.1145/2330163.2330273

[19] Mirko Messori, Chiara Toffanin, Simone Del Favero, Giuseppe De Nicolao, Claudio Cobelli, and Lalo Magni. 2016. Model individualization for artificial pancreas. *Computer Methods and Programs in Biomedicine* (2016), –. DOI: http://dx.doi.org/10.1016/j.cmpb.2016.06.006

[20] Alberto Moraglio. 2014. Project Title. https://github.com/amoraglio/GSGP. (2014).

[21] Alberto Moraglio, Krzysztof Krawiec, and Colin G. Johnson. 2012. *Geometric Semantic Genetic Programming*. Springer Berlin Heidelberg, Berlin, Heidelberg, 21–31. DOI: http://dx.doi.org/10.1007/978-3-642-32937-1_3

[22] Heinz Mühlenbein. 1997. The Equation for Response to Selection and Its Use for Prediction. *Evol. Comput.* 5, 3 (Sept. 1997), 303–346. DOI: http://dx.doi.org/10.1162/evco.1997.5.3.303

[23] Farzad Noorian, Anthony M. de Silva, and Philip H. W. Leong. 2016. gramEvol: Grammatical Evolution in R. *Journal of Statistical Software* 71, 1 (2016), 1–26. DOI: http://dx.doi.org/10.18637/jss.v071.i01

[24] Michael O'Neill and Conor Ryan. 2003. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, Norwell, MA, USA.

[25] Michael O'Neill, Leonardo Vanneschi, Steven Gustafson, and Wolfgang Banzhaf. 2010. Open Issues in Genetic Programming. *Genetic Programming and Evolvable Machines* 11, 3-4 (Sept. 2010), 339–363. DOI: http://dx.doi.org/10.1007/s10710-010-9113-2

[26] Martin Pelikan and Heinz Mühlenbein. 1998. Marginal distributions in evolutionary algorithms. In *Proceedings of the International Conference on Genetic Algorithms Mendel*, Vol. 98. Citeseer, 90–95.

[27] G. M. Steil, C. C. Palerm, N. Kurtz, G. Voskanyan, A. Roy, S. Paz, and F. R. Kandeel. 2011. The Effect of Insulin Feedback on Closed Loop Glucose Control. *J Clin Endocrinol Metab* 96 (2011), 1402–1408.

[28] Leonardo Vanneschi, Mauro Castelli, and Sara Silva. 2014. A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines* 15, 2 (2014), 195–214. DOI: http://dx.doi.org/10.1007/s10710-013-9210-0

[29] Chengxia Yu and Chunhui Zhao. 2014. Rapid Model Identification for Online Glucose Prediction of New Subjects With Type 1 Diabetes Using Model Migration Method. *{IFAC} Proceedings Volumes* 47, 3 (2014), 2094 – 2099. DOI: http://dx.doi.org/10.3182/20140824-6-ZA-1003.00534 19th {IFAC} World Congress.