This article can be cited as D. Azar and M. Bitar, AI-Based Methods for Predicting Required Insulin Doses for Diabetic Patients, International Journal of Artificial Intelligence, vol. 13, no. 1, pp. 8-24, 2015.

AI-Based Methods for Predicting Required Insulin Doses for Diabetic Patients

Danielle Azar[1] **and Mandy Bitar**[2]

[1]Department of Computer Science and Mathematics
Lebanese American University
Byblos, Lebanon 14010
danielle.azar@lau.edu.lb

[2]Department of Computer Science and Mathematics
Lebanese American University
Byblos, Lebanon 14010
mandy.bitar@lau.edu

ABSTRACT

Treating diabetes mellitus requires patients to retrieve multiple measurements daily over multiple years. This results in an enormous amount of data. Endocrinologists need to find a certain pattern in this data that would help them determine the optimal dosage of insulin to administer to each patient. However, keeping track of the data for this purpose is overwhelming. As a result, they often follow a trial and error approach until they find the individualized insulin dosage, required for each patient, to reach their optimal glucose level. Hence, there is a great need to automate this process. In this paper, we propose and compare three techniques two of which are Artificial Intelligence techniques, namely C4.5 and Case-Based Reasoning, and the third one is a meta-heuristic namely genetic algorithms. The performance of the three algorithms is evaluated on a data set found in the public UCMI repository.

Keywords: C4.5, Case-Based Reasoning, Decision Trees, Diabetes Mellitus, Genetic Algorithms, Glucose, Insulin.

ACM Computing Classification System: I.2.1

1 Introduction and Problem Statement

Diabetes Mellitus is a disease in which the human body has very low levels of insulin and/or is unable to produce or respond to insulin in order to keep normal levels of sugar in the blood. This results in poor maintenance of normoglycemia which is defined as blood glucose $70 - 100mg/dl$ and elevated blood glucose concentration ($300mg/dl$ or above)(Parker, Doyle and Peppas, 1999). If the accumulation of sugar in the blood goes untreated, it may lead to many complications such as heart diseases, kidney failure, retinopathy, risk of amputation, etc.(Watkins, 2003). To remedy to this, insulin doses are administered over a defined period of time in order to compute the optimal glucose level. For this, medical doctors need to collect

the following information for each patient: fasting glucose levels (3 times a day), after-meal glucose (after each meal), and bed-time glucose over a relatively long period of time (years). They, then, need to combine this information with both the ultra-lente and intermediate insulin dosages. The huge amount of collected data makes it difficult, if not impossible, to analyze and draw conclusions that help administer the correct dose of insulin for future(unseen) patients.
In this paper, we present three different techniques to tackle the problem: C4.5, Case-Based Reasoning (CBR) and Genetic Algorithms(GA). While C4.5 is a fixed technique that is used as is, we show how we instantiate CBR and GA to the specific problem in hand. Results show that all three techniques are suitable for this problem. In Section 2, we give an overview of the related work. In Section 3, we describe the techniques and the data we used in our work. In Section 4, we explain in detail the experiments that we performed. We show the results and we derive a comparative study of all three techniques. In Section 5, we conclude with a summary of the work and we open the floor for further improvement.

2 Related work

Artificial intelligence techniques have been used on a wide variety of optimization problems ranging from hardware to disease-related problems. For example, in (Preitl, Precup, Fodor and Bede, 2006), the authors propose a new design method for the PI-fuzzy controllers. The work focuses on 2-Degree of Freedom fuzzy control system structures. The approach is validated on a class of plants applied in servo systems as part of mechatronics systems and embedded systems. The work shows the potential of Iterative Feedback Tuning(IFT) algorithms when combined with fuzzy control in complex plants. It also guarantees the convergence of IFT algorithms. In (Spall, 1992), the authors tackle stochastic approximation (SA) which is a recursive procedure for finding the roots of an equation in the presence of noisy measurements. The authors present an SA procedure that significantly outperforms the usual p-dimensional algorithms (of Kiefer-Wolfowitz/Blum type) by reaching similar estimation accuracy with much less data. The procedure is based on a "simultaneous perturbation" gradient approximation. In (Zavoianu, Bramerdorfer, Lughofer, Silber, Amrhein and Klement, 2013), the authors apply Non-dominated Sorting Genetic Algorithm II (NSGA-II) in order to obtain high-quality pareto-optimal solutions for three optimization scenarios. The nature of these scenarios requires the usage of objective functions that are computationally expensive since they require intensive finite element simulations. For this, the authors introduce the novel aspect of creating on-the-fly highly accurate and stable surrogate fitness functions based on artificial neural networks. Results show that the procedure significantly reduces the computation time while achieving pareto-optimal solution sets with better quality. In (El-Hefnawi, 2014), the author proposes a modified particle swarm optimization technique to solve fuzzy bi-level single and multi-objective problems. The approach shows promising results.
Artificial intelligence techniques have also been widely used in the medical field in general, and diabetes mellitus in particular. For example, in (Hamou, Simmons, Bauer, Lewden, Zhang, Wahlund, Westman, Pritchard, Kloszewska, Mecossi, Soininen, Toslaki, Vellas, Muehlboeck, Evans, Julin, Sjogren, Spenger, Lovestone and Gwardy-Sridhar, 2011), the authors use clus-

tering and decision trees in order to show the influence of particular bio-markers and neuro-psychological variables on a patient's health. In (Rizvi and Bhattacharya, 2012), correlation and regression analysis are used in order to estimate the preponderance of GC, GC3, AT3 bias at nucleotide positions of genome. In (Ibrahim, Abdul-Khalid, Manaf and Ngah, 2011), the authors propose a Particle Swarm Optimization (PSO) based technique and a Seed-Based Region Growing (SBRG) approach for the segmentation of human brain tissue abnormalities. They compare both techniques which show promising results in detecting abnormalities in the human brain tissues. In (Torres, Quintana and Pinzn, 2013), the authors present a neural network-based approach to the differential diagnosis of Dengue Fever, Leptospirosis and Malaria, using the Adaptive Resonance Theory Map (ARTMAP) family. A subset of symptoms were identified which enhanced the performance of the classifiers considered. In particular regarding Diabetes Mellitus, Artificial Intelligence techniques have been used on many instances of problems related to the disease. In (Barriga, Harman, Hoag, Marshall and Shetterly, 1996), the authors use CART (Brieman, Friedman, Stone and Olshen, 1984) -a decision tree learning algorithm- to screen and analyze a population of patients in order to reduce the number of times patients are tested for oral glucose tolerance. In (Bellazzi, Larizza, Magni, Montani and Stefanelli, 2000), Intelligent Data Analysis is used to analyze and interpret clinical time series in diabetes to monitor a patient's overall situation. In (Maimone, 2006), (Miller, 2009) and (Walker, 2007), the authors propose the use of Case-Based Reasoning (CBR) to manage and control diabetes. In (Wahab, Kong and Quek, 2006), the authors use the model reference adaptive control on a controller to adaptively maintain normal levels of glucose in the patient's blood via an infusion pump. They also apply the proposed adaptive control algorithm to a fuzzy logic control of the glucose-insulin system which surpasses the conventional application. In (Nguyen, Pham and Triantaphyllou, 2009), the authors use meta-heuristics, in particular, Homogeneity-Based Algorithm (HBR), to predict diabetes. In (Pham and Triantaphyllou, 2008), the authors try data mining approaches which balance between fitting and generalization. In (Purnami, Embong, Zain and Rahayu, 2009), Smooth Support Vector Machine(SSVM) is used to predict the presence of diabetes in patient. In (Cho, Yu, Hwanjo, Kim, Kim and Kim, 2008), Support Vector Machines(SVM) are used along with feature selection methods to predict diabetic neuropathy. In (Vosolipour, Aliyari and Teshnehlab, 2008), fuzzy inference systems are used to predict diabetes. In (Vosoulipour, Teshnehlab and Moghadam, 2008), artificial neural networks are used to classify subjects into two classes namely non-diabetics and diabetics. In (Qiu, Rajagopalan, Connor, Damian, Zhu, Handzel, Hu, Amanullah, Bao, Woody, MacLean, Lee, Vanderwall and Ryan, 2008), the authors use multivariate classification analysis of metabolomic data for candidate biomarker discovery in Type 2-diabetes. In (Barakat, Bradley and Barakat, 2009), the authors add an explanation module to the Support Vector Machines technique in order to diagnose diabetes. In (Zhang, Song and Wu., 2009), a fuzzy integral technique is used to create a module that diagnoses gestational diabetes mellitus and uses history as a main source to predict how much a person is pre-disposed to the disease. In our work, we propose three different techniques namely Case-Based Reasoning (CBR), Genetic Algorithms (GA) and C4.5 (Quinlan, 1993). We demonstrate the effectiveness of all three techniques using a data set from the UCI Machine Learning Repository.

Table 1: Example of a data file describing a single patient

Date (mm-dd-yy)	Time) (hh:mm)	Measurement Type	Glucose level
04-21-1991	9:09	58	100
04-21-1991	9:09	33	007
04-21-1991	9:09	34	016
04-21-1991	17:08	62	119
04-21-1991	17:08	33	009
04-21-1991	22:51	48	123
04-22-1991	7:35	58	216
04-22-1991	7:35	33	006
04-22-1991	7:35	34	013
04-22-1991	13:40	33	006
04-22-1991	16:56	62	211
04-22-1991	16:56	33	006
04-23-1991	7:25	58	257
04-23-1991	7:25	33	011
04-23-1991	7:25	34	013
04-23-1991	14:10	61	238
04-23-1991	22:16	48	340

3 Materials and Methods

We start by describing the data that we use in our experiments. Then, we describe in detail the three techniques that we propose.

3.1 Data preparation

Treating diabetic patients involves measuring their glucose levels on a daily basis several times a day (typically three) over a period of years. Our data set is taken from the UCMI repository (*UCMI repository for machine learning*, 2014). It consists of $29,330$ cases extracted from 70 diabetic patients. A patient is described in terms of the following attributes: the date on which a sample of glucose level is taken, the time, the type of glucose evaluated and the level of glucose obtained. Table 1 shows an example of a single patient and Table 2 shows the different types of measurements that are made.

We first start by cleaning the data from erroneous and missing values. Medical experts help is used to determine erroneous entries[1]. We, then augment our data set with some additional attributes namely: *fasting glucose*, *after-meal glucose*, *bed-time glucose*, *intermediate insulin* and *ultralente insulin*. These are computed by combining the data pertaining to one patient on

[1]One example of an erroneous entry is a recorded measurement level greater than 40 when measurement type is 33.

Table 2: Description of Measurement Types

Measurement Type	Description
33	Regular insulin dose
34	NPH insulin dose
35	Ultra-lente insulin dose
48	Unspecified blood glucose measurement
57	Unspecified blood glucose measurement
58	Pre-breakfast blood glucose measurement
59	Post-breakfast blood glucose measurement
60	Pre-lunch blood glucose measurement
61	Post-lunch blood glucose measurement
62	Pre-supper blood glucose measurement
63	Post-supper blood glucose measurement
64	Pre-snack blood glucose measurement
65	Hyperglycemic symptoms
66	Typical meal ingestion
67	More-than-usual meal ingestion
68	Less-than-usual meal ingestion
69	Typical exercise activity
70	More-than-usual exercise activity
71	Less-than-usual exercise activity
72	Unspecified special (very emotional event, etc.)

Table 3: Values extracted from Table 1 for the following dates: 04-21-1991, 04-22-1991 and 04-23-1991

Date	Fasting Glucose	Aftermeal Glucose	Bedtime Glucose	Intermediate Insulin	Ultralente Insulin	Regular Insulin
04-21-1991	109.5	0	123	16	0	8
04-22-1992	213.5	0	0	13	0	6
04-23-1991	257	238	340	13	0	11

one particular day and computing the averages of the respective attributes. An example of the resulting data set is shown in Table 3.

3.2 Proposed Techniques

We propose two artificial intelligence techniques, namely C4.5 and Case-Based Reasoning, and one meta-heuristic- genetic algorithm- to tackle the problem. We used C4.5 on Linux[2] and implemented Genetic Algorithm and Case-Based Reasoning in JAVA[3]. We will, next, give an overview of all algorithms and, as we proceed, we explain in detail how we instantiated the last two to our problem.

3.2.1 C4.5

C4.5 is a machine learning algorithm used to build decision trees from a set of data (Quinlan, 1993). A decision tree is a tree where internal nodes encode attributes, edges conditions on attributes and leaves classification labels (Figure 1). A path from the root to a leaf encodes a conjunction of conditions and the result of the test is recorded in the leaf. Given an instance, it is classified by starting at the root and following the path formed of those edges where conditions evaluate to true. The instance is then assigned the classification label stored in the leaf. In the tree in Figure 1, an instance where the fasting glucose is less than or equal to 70 and the after-meal glucose is less than or equal to 100 and the bed-time glucose is less than or equal to 120 is assigned a regular insulin dose of 0.
Trees can become very large which makes them harder to interpret by human experts. For this, we transform them into rule sets. A rule set is a collection of rules and a default classification label. A rule is a conjunction of conditions or attributes and a classification label. One example of a rule extracted from the tree in Figure 1 is: *IF Fasting Glucose* ≤ 70 *AND Aftermeal Glucose* ≤ 100 *AND Bedtime Glucose* ≤ 120 *THEN Regular Insulin* $= 0$. Usually, rule sets are extracted from the trees after a pruning process during which branches are eliminated from the tree without deteriorating the classification accuracy. The resulting rule set may not cover all instances in the data set. For this, it is augmented with a default classification label which

[2]C4.5 can be downloaded for free from http://www2.cs.uregina.ca/ dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html and http://www.rulequest.com/Personal/

[3]Code for Genetic Algorithm and Case-Based Reasoning is available upon request to the primary author.

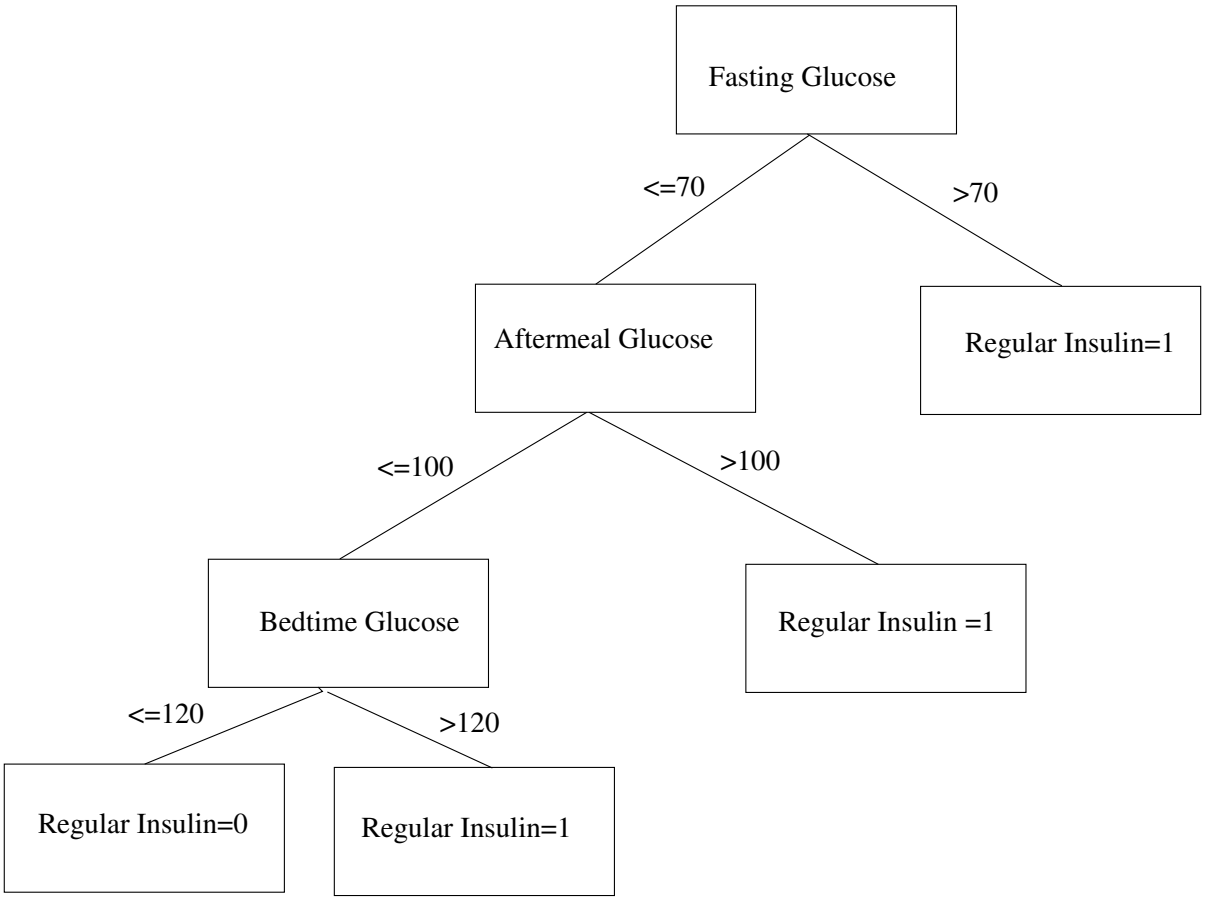

Figure 1: Decision tree. Nodes encode attributes, edges test on attributes and leaf nodes classification labels.

applies to all cases that are not covered by any of the rules[4]. C4.5 has a time complexity of $O(m.n^2)$ where m is the size of the training set and n is the number of attributes.

3.2.2 Case-Based Reasoning

Case-Based Reasoning (CBR) was first introduced by Roger Schank and his students at Yale University in the early 1980's. This approach is widely used in biomedical reasoning nowadays (Bichindaritz and Marling, 1992) and (Bichindaritz and Marling, 2005). CBR is based on the idea of learning from past cases and reusing knowledge on current similar cases. This is very similar to the diagnostic mechanism used by medical experts. Given a case x (a target case), CBR goes through four basic steps in order to classify it. These are: *Retrieve*, *Reuse*, *Retain* and *Revise*.

- *Retrieve:*The algorithm maintains a list of cases similar to x. A case is said to be similar to x if it has at least one of the following attributes equal to the respective one in x: *fasting glucose*, *after-meal glucose* or *bed-time glucose*. If one or more cases match, they are added to the *similar cases list* and the algorithm proceeds to the next step (*Reuse*). Otherwise, it assigns to x the default class label which depends on the value of *Fasting glucose* (Table 4) as defined after consultation with the endocrinologist.

Table 4: Default classification label (regular insulin) dependence on fasting glucose levels.

Fasting Glucose	**Regular Insulin**
< 150	6
$150 \leq$ Fasting glucose < 200	12
$200 \leq$ Fasting glucose < 250	18
≥ 250	24

  Figure 2 shows an example of target case x and three similar cases. The first case has the same aftermeal glucose value. The second and the third have the same fasting glucose and intermediate insulin values.

- *Reuse:* Once the similar cases have been retrieved, CBR computes the average values of *ultra-lente* and *intermediate insulin* and assigns to x the average of regular insulin computed from these similar cases. This gives a general idea of the average value of regular insulin given the values of the other attributes.

- *Revise:* At this step, CBR tests the target case to see if the assigned value of regular insulin matches. The accuracy of the test is measured in terms of the fasting glucose value $\pm\theta$ a safety buffer value provided by the endocrinologist. In case of a mismatch, it assigns a default label to the case.

- *Retain:* At this step, x is retained in memory.

[4]This is the majority classification label in the part of the data that is not classified by any of the rules in the rule set.

Target case x:

Aftermeal Glucose	Fasting Glucose	Bedtime Glucose	Intermediate Insulin	Ultralente	Regular
135	110	123	0	8.50	?

Similar cases:

Aftermeal Glucose	Fasting Glucose	Bedtime Glucose	Intermediate Insulin	Ultralente	Regular
135	140	115	5	11	14
85	110	111	0	5	6
185.16	110	135	0	3	13

Figure 2: Target case x and three similar cases.

Algorithm 1 shows the pseudo-code of CBR and how we instantiated it to our problem. The time complexity of this algorithm is $O(n.t.a)$ where n is the size of the training set, t is the size of the testing set and a is the number of attributes.

3.2.3 Overview of genetic algorithms

Genetic algorithms (GA) were initially introduced by John Holland in the late 1960's (Holland, 1992). In this algorithm, a population of individuals representing solutions to the underlying problem is created. These individuals compete to survive and produce progeny (other solutions) through an inspiration of the theory of natural selection and survival of the fittest. The pseudo-code of a generic GA is shown in Algorithm 2. To tackle a problem using GA, we need to instantiate the basic elements in this algorithm namely the encoding scheme (how a solution is represented as an individual), the fitness function, and the genetic operators (crossover, mutation and elitism).

The encoding scheme In our case, a chromosome is formed of genes each of which defines a finite range of values for a particular attribute (Figure 3). The fitness function should reflect the quality of the underlying solution. In our case, we define the fitness of a chromosome to be a linear combination of the accuracy of the underlying rule and its coverage (Equation 3.1). *Accuracy* of a rule is the percentage of cases in the data set which are correctly classified by the rule (Equation 3.2). *Coverage* is the percentage of cases in the data set to which the rule applies (Equation 3.3), and α and β are two parameters that we tune according to the weights we wish to give to *Accuracy* and *Coverage*.

$$Fitness(X) = \alpha Accuracy(X) + \beta Coverage(X) \tag{3.1}$$

$$Accuracy(X) = \frac{T}{T+F} \tag{3.2}$$

Algorithm 1 CBR Pseudocode

Data: n is the size of testing set, p is the size of training set

1: **for** $i = 0 \rightarrow n$ **do**
2: Pick target size X_i
3: **for** $j = 0 \rightarrow p$ **do**
4: **if** (fasting glucose of C_j = fasting glucose of X) OR (aftermeal glucose of C_j = aftermeal glucose of X) OR (bedtime glucose of C_j = bedtime glucose of X) **then**
5: Save the matched C_j case to the matching cases array.
6: **end if**
7: **if** matching cases array is empty **then**
8: Give X the default class
9: **end if**
10: Compute the average of classification labels c of:
▷ 2 cases from the array of matching cases that have the closest values of ultralente insulin values to those of X.
▷ 2 cases from the array of matching cases that have the closest values of intermediate insulin values to those of X.
11: Classify X with c.
12: Compute frequency, classification label and standard deviation of X.
13: **end for**
14: **if** the classification of X is correct **then**
15: Retain
16: **else**
17: Give X the default class
18: **end if**
19: **end for**

Algorithm 2 Pseudocode of GA

Input: A population of individuals

Output: The fittest individuals in the last population.

1: **for** a defined number of iterations **do**
2: **repeat**
3: Select from population $Parent_1$ and $Parent_2$
4: Crossover $Parent_1$ and $Parent_2$ with a certain probabilityto form offspring
5: Mutate offspring with a certain probability
6: Add both offspring to the next generation.
7: **until** the next population is complete
8: **end for**

G1	G2	G3	G4	G5
[104-108]	[103- 105]	[275- 288]	[12- 14]	[0-2]

Figure 3: Chromosome formed of 5 genes. G_1 indicates the range of values for attribute *fasting glucose*, G_2 *after-meal glucose*, G_4 *intermediate insulin* and G_5 *ultra-lente insulin*.

In Equation 3.2, T is the number of cases correctly classified by X, and F is the number of cases incorrectly classified by X.

$$Coverage(X) = \frac{T+F}{S}, \text{where } S \text{ is the size of the whole data} \tag{3.3}$$

Preparing the initial population Preparation of the data set consists of 5 steps.

Step1 For each attribute, we calculate its minimum and maximum values in the data set. For example, in Table 5, *fasting glucose* has the following values: min $= 103$ and max $= 213.50$. We then divide the values in K ranges (K is determined empirically). If we let $K = 40$ in our example, we get the following range size: $(213.5 - 103.0)/40 = 2.7625$ and the following ranges for fasting glucose: $[103.0 - 105.7625[$, $[105.7625 - 108.525[$, $[108.525 - 111.2875[$, etc.

Table 5: Sample of the Training Set

Fasting Glucose	Aftermeal Glucose	Bedtime Glucose	Intermediate Insulin	Ultralente Insulin	Regular Insulin
109.50	0.00	123.00	13.00	0.00	8.00
213.50	0.00	0.00	13.00	0.00	6.00
111.00	0.00	240.00	14.00	0.00	6.00
136.50	288.00	0.00	14.00	0.00	5.00
194.33	0.00	81.00	81.00	14.00	7.00
150.66	0.00	104.00	14.00	0.00	7.00
103.00	0.00	0.00	14.00	0.00	7.00

Step2 We calculate the frequency, classification label and standard deviation of the ranges. For example, using the attribute *fasting glucose*, the following range $[103.0 - 105.7625[$ covers one case in the data set. The classification label is the average of all classification labels that this range covers. In this case it is only one value (7). Therefore the standard deviation is 0. Range $[108.525 - 111.2875[$ covers 2 cases in the data set. The classification label in this case is $(8 + 6)/2 = 7$ and the standard deviation is 1.

Step3 For each attribute, we create a set of ranges (Figure 4).

Step 4 Creating the initial population: We form chromosomes by picking random ranges from all sets of ranges such that each attribute gets exactly one range in the same chromosome (Figure 3).

A1	A1	...	A5	A5
[103-105.76]	[210-213]		[275-279]	[280-288]

Figure 4: Ranges for each attribute.

To complete a chromosome, the classification label of the underlying rule is computed as the average of classification labels of all cases that the rule classifies. For example, the chromosome in Figure 3 gets classification label $c = 8$ and standard deviation $std = 0$ on the training set shown in Table 5. This is because the chromosome covers a single case.

Step 5 The Evolution Process: We use the Roulette Wheel Selection technique to choose the two parent chromosomes that will undergo crossover. This technique ensures a chance of selection proportionate to the fitness of the chromosomes. We then perform single-point crossover with probability δ on the selected chromosomes. This is illustrated in Figure 5.

X : ↓

A_1	A_2	A_3	A_4	A_5
$[103.0 - 105.76]$	$[280.8 - 288]$	$[114 - 120]$	$[79.3 - 81]$	$[0 - 0.35]$

Y : ↓

A_1	A_2	A_3	A_4	A_5
$[108.52 - 111.28]$	$[0 - 7.20]$	$[120 - 126]$	$[13 - 14.7]$	$[0 - 0.35]$

O_1 :

A_1	A_2	A_3	A_4	A_5
$[103.0 - 105.76]$	$[280.80 - 288]$	$[114 - 120]$	$[13 - 14.7]$	$[0 - 0.35]$

O_2 :

A_1	A_2	A_3	A_4	A_5
$[108.525 - 111.28]$	$[0 - 7.2]$	$[120 - 126]$	$[79.3 - 81]$	$[0 - 0.35]$

Figure 5: Single-Point Crossover on X and Y. Arrows indicate cut points. This results in two offspring chromosomes O_1 and O_2.

The offspring are then mutated with probability μ. Mutation consists of changing the range of one attribute into another one selected from the set of ranges for this attribute. Figure 6 illustrates this operator on Offspring O_1 where the fourth gene is changed from $[13 - 14.7]$ to $[11.3 - 13]$. The mutated offspring are then added to the new population.

Step 6 Elitism: The best chromosome is preserved to the next population. Then, Step 5 is repeated p times replacing at each iteration the old population with the new one.

Figure 6: Mutation. The fourth gene in the chromosome is mutated : The value of A_4 is changed.

O_1 before mutation: $\downarrow$

A_1	A_2	A_3	A_4	A_5
$[103.0 - 105.76]$	$[280.80 - 288]$	$[114 - 120]$	$\mathbf{[13 - 14.7]}$	$[0 - 0.35]$

O_1 after mutation: $\downarrow$

A_1	A_2	A_3	A_4	A_5
$[103.0 - 105.76]$	$[280.8 - 288]$	$[114 - 120]$	$\mathbf{[11.30 - 13]}$	$[0 - 0.35]$

The genetic algorithm runs in $O(i.p.n.a)$ where i is the number of iterations over which the evolution process is repeated, p is the size of the population, n is the size of the training data[5] and a is the number of attributes.

4 Empirical evaluation

In this section, we describe the experiments used to validate each of the three algorithms. C4.5 was used as is. We implemented GA and CBR as described in the previous section and we validated all three algorithms on the same data set.

4.1 Experimental setup

In order to validate our approach, we use $10-$fold cross validation. By this technique, the data set is divided into 10 folds of roughly equal size. Nine of the folds are combined and their union forms the training set. The remaining fold is used as a testing set. This is repeated 10 times leaving each time a different fold as a testing set. Furthermore, since the GA incorporates an element of randomness, we repeat each run 30 times and we report the average over the 30 runs. Movreover, since the computation of the classification labels in both GA and CBR might result in decimal points, we round up or down the class label to evaluate whether a classification of a case is correct. For example, if we have a class of 2.7 we consider this classification label to be 3. We performed the experiments on a PC with 1 GB of RAM, 1.7 Ghz CPU running on Windows XP Operating System. The experiments were completed in less than 1 minute for C4.5 and CBR and around 5 minutes per run for the GA. In the case of the GA, we performed several experiments with different sets of parameters. In Table 6, we report the parameters that showed to be the most suitable for our problem. The number of iterations seemed to have the greatest impact on the results. However, beyond the value indicated in the table (200), the results varied very slightly at the expense of a significant increase in the running time.

[5]Every time a new chromosome is created, the fitness is computed by going over the whole training set.

Table 6: GA parameters. K is the number of ranges for each attribute. i is the number of iterations. Q is the number of chromosomes in a population. δ is the crossover probability and μ is the mutation rate.

K	i	Q	δ	μ
40	200	150	1	0.01

4.2 Results

Table 7 summarizes the results obtained with all three algorithms. Results show a low standard deviation for all three algorithms proving their stability on this problem. In the case of the GA, this also shows that the random parameters do not affect significantly the results . GA scored the highest accuracy on the testing set (65.52%) outperforming CBR by about 8% and C4.5 by about 16%. This shows that GA is the best algorithm to use in order to predict doses to administer to new patients. On the other hand, CBR scored the highest accuracy on the training sets (85.55%) outperforming GA by 19% and C4.5 by 21%. We believe that this is due to the fact that CBR mimics the behavior of human experts who tend to classify cases by comparing them to previously seen ones.

Table 7: Accuracy and standard deviation of all three techniques on both the training and the testing data.

Technique	**Accuracy on training (stdv)**	**Accuracy on testing (stdv)**
C4.5	64.40(1.16)	49.80(1.52)
CBR	85.55(0.27)	57.62(1.85)
GA	66.13(1.51)	65.52(2.20)

In order to test our results for statistical significance, we use the non-parametric Wilcoxon signed rank test. We perform a pair-wise comparison of the three algorithms. All tests show a z value of -2.75 which is less than the critical $z-$ value (-2.29) indicating a significance level $\alpha = 0.05$. This shows that it is very unlikely that results are due to chance.

5 Conclusion and Future Work

Treating diabetes mellitus results in an enormous volume of data. Endictronoglists need to find a pattern in this data that helps them determine the optimal dosage of insulin to administer to new patients. In this work, we propose three different artificial intelligence-based algorithms to tackle the problem. They are C4.5, Case-Based Reasoning (CBR) and Genetic Algorithms(GA). We have tested all three algorithms on a data set extracted from the UCMI repository. The data set consists of $29,330$ cases and describes 70 patients. Among all three algorithms, CBR showed the best performance on the training set and GA on the testing set. We believe that the behavior of CBR is due to the fact that it mimics to some extent the diagnosis process performed by human experts. The latter read the patient's medical history, and

based on similar past cases, predict the most convenient treatment. But if the medical doctor is faced with a new patient, there is no past similar history to match with, then the patient is normally administered a default treatment and through trial and error, the treatment will be tailored and personalized to that patient. This is the limitation of CBR which, nonetheless, remains better than C4.5. On the other hand, GA is a heuristic that detects a pattern in the whole data set (based on recombination of attribute value ranges) and extracts rules or guidelines to apply on new unseen patients. Thus, GA learns from different patients and combines the knowledge it acquires from all. We believe that GA is better than CBR for solving this problem given that every diabetic patient is unique to some extent and there is a need to have general rules that can apply to the majority of the patients. Finally, since CBR performed well on the training data and GA performed well on the testing data, an interesting approach would be to combine both algorithms and benefit from the strengths of each. We believe that this can lead to an improved solution to the problem. Although the results of the three techniques are not excellent but we hope that this work will pave the way to future approaches.

Acknowledgment

This work has been funded by a grant from the School of Arts And Sciences Research and Development Council at the Lebanese American University.

References

Barakat, N. H., Bradley, A. P. and Barakat, M. N. 2009. Intelligible support vector machines for diagnosis of diabetes mellitus, *Information Technology in Biomedicine* **14**: 1114–1120.

Barriga, K. J., Harman, R. F., Hoag, S., Marshall, J. A. and Shetterly, S. 1996. Population screening for glucose intolerant subjects using decision tree analyses, *Diabetes Research and Clinical Practice* **34**: 17–29.

Bellazzi, R., Larizza, C., Magni, P., Montani, S. and Stefanelli, M. 2000. Intelligent analysis of clinical time series: an application in the diabetes mellitus domain, *Artificial Intelligence in Medicine* **20**(1): 37–57.

Bichindaritz, I. and Marling, C. 1992. An Introduction to Case-Based Reasoning, *Artificial Intelligence Review* **6**: 3–34.

Bichindaritz, I. and Marling, C. 2005. Case-based reasoning in the health sciences: What's next?, *Artificial Intelligence in Medicine* **36**: 127–135.

Brieman, L., Friedman, J., Stone, C. J. and Olshen, R. 1984. *Classification and Regression Trees*, Chapman & Hall.

Cho, B. H., Yu, H., Hwanjo, Kim, K., Kim, T. H. and Kim, S. I. 2008. Application of irregular and unbalanced data to predict diabetic nephropathy using visualization and feature selection methods, *Artifificial Intelligence in Medicine* **42**(1): 37–53.

El-Hefnawi, N. A. 2014. Solving bi-level problems using modified particle swarm optimization algorithm, *International Journal of Artificial Intelligence* **12**: 88–101.

Hamou, A., Simmons, A., Bauer, M., Lewden, B., Zhang, Y., Wahlund, L.-O., Westman, E., Pritchard, M., Kloszewska, I., Mecossi, P., Soininen, H., Toslaki, M., Vellas, B., Muehlboeck, S., Evans, A., Julin, P., Sjogren, N., Spenger, C., Lovestone, S. and Gwardy-Sridhar, F. 2011. Cluster analysis of mr imagining in alzeheimer's disease using decision tree refinement, *International Journal of Artificial Intelligence* **6**(S11): 90–99.

Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press.

Ibrahim, S., Abdul-Khalid, N. E., Manaf, M. and Ngah, U. K. 2011. Particle swarm optimization vs seed-based region growing: Brain abnormalities segmentation, *International Journal of Artificial Intelligence* **7**(A11): 174–188.

Maimone, A. 2006. *Data and knowledge acquisition in case-based reasoning for diabetes management*, Master's thesis, University of Ohio.

Miller, W. A. 2009. *Problem detection for situation assessment in case-based reasoning for diabetes management*, Master's thesis, University of Ohio.

Nguyen, H., Pham, A. and Triantaphyllou, D. E. 2009. An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets, *Expert Systems with Application* **36**: 9240–9249.

Parker, R. S., Doyle, F. J. and Peppas, N. A. 1999. A model-based algorithm for blood glucose control in type i diabetic patients, *IEEE Transactions on Biomedical Engineering* **46**(2).

Pham, H. N. A. and Triantaphyllou, E. 2008. Prediction of diabetes by employing a new data mining approach which balances fitting and generalization, *Computer and Information Science* (1): 11–26.

Preitl, S., Precup, R.-E., Fodor, J. and Bede, B. 2006. Iterative feedback tuning in fuzzy control systems. theory and applications, *Acta Polytecnica Hungarica* **3**(3): 81–96.

Purnami, S. W., Embong, A., Zain, J. M. and Rahayu, S. 2009. A new smooth support vector machine and its applications in diabetes disease diagnosis, *Journal of Computer Science* **5**(12): 1003–1008.

Qiu, Y., Rajagopalan, D., Connor, S. C., Damian, D., Zhu, L., Handzel, A., Hu, G., Amanullah, A., Bao, S., Woody, N., MacLean, D., Lee, K., Vanderwall, D. and Ryan, T. 2008. Multivariate classification analysis of metabolomic data for candidate biomarker discovery in type 2 diabetes mellitus, *Metabolomics* **4**: 337–346.

Quinlan, J. 1993. *C4.5: Programs for machine learning*, Morgan Kaufmann.

Rizvi, A. Z. and Bhattacharya, C. 2012. An efficient algorithm and schematic for computation of gc, gc3, and at3 bias spectra, *International Journal of Artificial Intelligence* **9**(A12): 19–25.

Spall, J. C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Transactions on Automatic Control* **37**(3): 332–341.

Torres, W. C., Quintana, M. and Pinzn, H. 2013. Differential diagnosis of hemorrhagic fevers using artmap and an artificial immune system, *International Journal of Artificial Intelligence* **11**(A13): 150–169.

UCMI repository for machine learning 2014. "`http://archive.ics.uci.edu/ml/datasets/Diabetes`"

Vosolipour, A., Aliyari, M. and Teshnehlab, M. 2008. Hierarchical takagi-sugeno type fuzzy system for diabetes mellitus forcasting, *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, San Diego, California, USA, pp. 1265–1270.

Vosoulipour, A., Teshnehlab, M. and Moghadam, H. A. 2008. Classification on diabetes mellitus dataset based on artificial neural networks and anfis, *Biomed* **21**: 11–26.

Wahab, A., Kong, Y. K. and Quek, C. 2006. Model reference adaptive control on glucose for the treatment of diabetes mellitus, *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pp. 315–320.

Walker, D. 2007. *Similarity Determination and Case Retrieval in an Intelligent Decision Support System for Diabetes Management*.

Watkins, P. J. 2003. *ABC of Diabetes*, MJ Publishing Group Ltd.

Zavoianu, A.-C., Bramerdorfer, G., Lughofer, E., Silber, S., Amrhein, W. and Klement, E. P. 2013. Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives, *Engineering Applications of Artificial Intelligence* **26**: 1781–1794.

Zhang, C., Song, J. and Wu., Z. 2009. Fuzzy Integral Applied to the Diagnosis of Gestational Diabetes Mellitus, *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, Tianjin, China, pp. 296–300.