

# Temporal Case-based Reasoning for Type 1 Diabetes Mellitus Bolus Insulin Decision Support

Daniel Brown [corresponding author], Arantza Aldea, Rachel Harrison, Clare Martin, Ian Bayley

---

## Abstract

Individuals with type 1 diabetes have to monitor their blood glucose levels, determine the quantity of insulin required to achieve optimal glycaemic control and administer it themselves subcutaneously, multiple times per day. To help with this process bolus calculators have been developed that suggest the appropriate dose. However these calculators do not automatically adapt to the specific circumstances of an individual and require fine-tuning of parameters, a process that often requires the input of an expert.

To overcome the limitations of the traditional methods this paper proposes the use of an artificial intelligence technique, case-based reasoning, to personalise the bolus calculation. A novel aspect of our approach is the use of temporal sequences to take into account preceding events when recommending the bolus insulin doses rather than looking at events in isolation.

The in silico results described in this paper show that given the initial conditions of the patient, the temporal retrieval algorithm identifies the most suitable case for reuse. Additionally through insulin-on-board adaptation and postprandial revision, the approach is able to learn and improve bolus predictions, reducing the blood glucose risk index by up to 27% after three revisions of a bolus solution.

*Keywords:* case-based reasoning, temporal, diabetes, feature selection, knowledge based systems, similarity measures

---

## 1. Introduction

Type 1 diabetes mellitus (also known as T1DM) is a medical condition in which insulin-producing cells in the pancreas are destroyed and the body can no longer produce the hormone, insulin, which in turn means that blood glucose is no longer absorbed by other cells including fat cells and muscle cells. This leads to high blood glucose levels that can have serious health consequences. A complementary hormone, glucagon, is secreted when blood glucose levels fall too low. Patients with T1DM typically have to manage blood glucose levels by introduction of insulin themselves through periodic

monitoring of blood glucose levels and an insulin-injection regime, a complex task even for the most motivated individual [1].

Bolus insulin calculators have been shown to be effective in assisting the management of the condition [2]. However, these bolus calculators will always produce the same result from the user's inputs unless certain settings such as the carbohydrate-to-insulin ratio (CIR) and insulin sensitivity factor (ISF) are altered, a process often guided through clinicians. Our research aims to address this issue by replacing the static formula with the ability to learn and improve bolus recommendations automatically through case-based reasoning (CBR).

The contributions of this research are a novel temporal approach to enhance case retrieval by identifying the most similar sequences of events; the incorporation of an adaptation rule for active insulin in a temporal context; and a postprandial (post-meal) revision algorithm to allow the system to learn. Additionally, a detailed comparison of single-attribute evaluation algorithms included in

---

<sup>★</sup>We thank Oxford Brookes University for funding this research.

the Weka data mining tool for the purpose of feature weighting in the similarity measure is also carried out as part of this research [3].

This paper is organised as follows. Section 2 explains the fundamentals of CBR, highlighting the limitation of using cases in isolation in temporal domains where events in the past has consequences in the current actions such as T1DM. In Section 3 we describe our method for solving this problem using temporal CBR. In Section 4 we discuss and analyse Weka’s different single-attribute feature selection algorithms, adopted for the purpose of feature weighting in case retrieval. This is followed by details of the data and statistical measures used for evaluating the proposed methodology in Section 5. Section 6 outlines the *in silico* results of this approach, showing the system’s ability to improve results over time. Limitations and improvements in the proposed system are discussed in Section 7. Related work in the T1DM domain and temporal CBR is discussed in Section 8. Finally, conclusions reached are discussed in Section 9.

## 2. Case-based Reasoning

Case-based reasoning attempts to mimic the human ability to recall similar situations that occurred in the past and adapt them to address new problems. The foundations of CBR can be found in the work conducted by Kolodner based on the idea of dynamic memory modelling proposed by Schank [4, 5]. Several applications were developed to demonstrate the capabilities of CBR for solving real world problems, notable seminal examples include CHEF, MEDIATOR, and CASEY [6, 7, 8].

The primary knowledge store in CBR is the case-base, which is a collection of situations, scenarios, or events. Each case contains values for a set of features and a corresponding solution. In the case of T1DM, the features could be carbohydrates ingested, physical activity, time of the day, and other patient data, with the solution being the bolus insulin recommendation. The goal of CBR is to identify the case that best reflects the current situation and to adapt it to solve a new problem. One widely adopted CBR model is the  $R^4$  model (Figure 1) proposed by Aamodt and Plaza [9]. The  $R^4$  model consists of four stage cycle: *retrieve*, *reuse*, *revise*, and *retain*. First, a new problem is presented to the system consisting of features and feature-values, then a similar case is retrieved. The retrieved case is then reused to solve the new problem; this may

involve some form of adaptation to resolve any discrepancies between the proposed problem and the retrieved case. A solution is then presented and revised by the user or system. If the proposed solution is accepted it is then retained in the case-base as a new case. This cycle then repeats to enable the system to continuously improve suggestions as its knowledge (case-base) grows.

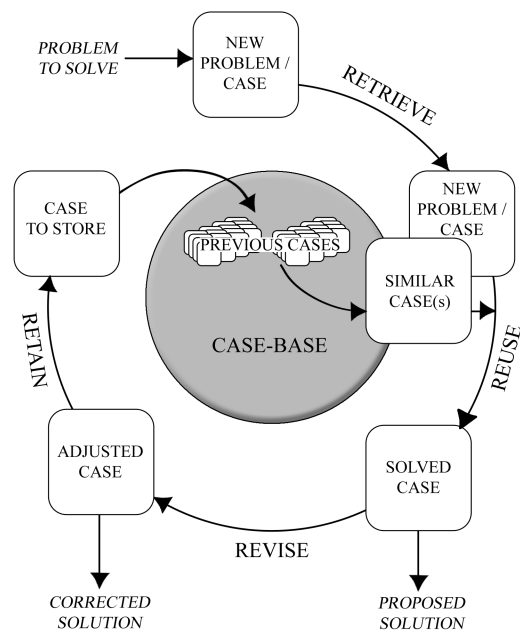


Figure 1:  $R^4$  CBR cycle [9]

The majority of research and development using CBR considers each case as an isolated event. In the context of T1DM we believe that temporal effects should be factored into the retrieval step so that an individual’s recent events can be taken into account. Research into temporal CBR has been relatively limited, with the majority of methods requiring specialist case representation, e.g. [10, 11]. To overcome this, sequences of continuous temporal cases that are linked to each other can be merged into a singular case [12]. This method allows the temporal sequences to be compared using standard distance metrics (e.g. Euclidean distance) without the need for additional rules. Plausible episodes are generated from a new problem, which are then compared to similar retrieved episodes in order to solve the new problem. We use this formation of episodes as the foundation for our temporal approach.

### 3. Methodology

This section describes our proposed temporal CBR system (Figure 2) for insulin bolus advice [13, 14] based on the R<sup>4</sup> model.

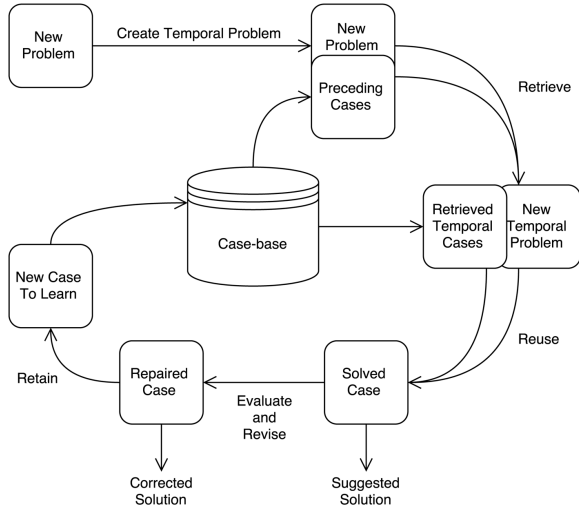


Figure 2: Case-based reasoning model for T1DM bolus advice

First we identify the features that are required to represent a case and determine how the cases for T1DM bolus are modelled. This is followed by a description of how each step of the R<sup>4</sup> model was developed and adapted to deal with temporal information.

#### 3.1. Case Structure

Unlike other CBR systems where case features may vary from case to case, in this context the features representing a case are well-defined. The initial step taken by this research was to determine which parameters are required by bolus calculators. Through assessment of commercial and freely available smart phone bolus calculators (Accu-Chek<sup>®</sup> Aviva Expert, RapidCalc, Diabetes Personal Calculator, Diabetic Dosage, and InsulinCalc), the parameters described in Table 1 were identified. The selection method of the bolus calculator applications is described by Martin et al. [15].

The parameters shown in Table 1 allow us to describe the features of a case. It is clear that the carbohydrate intake, preprandial (pre-meal) blood glucose level, and target blood glucose level are essential case parameters as they are taken into account by all the calculators assessed.

The *Insulin Sensitivity Factor* (ISF) and *Carbohydrate-to-Insulin Ratio* (CIR) are the primary parameters used to tune the bolus calculator. However, these factors will be omitted from the cases. This is due to the fact that the ISF and CIR values are usually defined as personal settings on the device, and largely remain static for different time periods. As a result, the majority of cases will retain the same ISF and CIR for a given time period, making them redundant to the CBR retrieval step. Our CBR approach will instead use the date and time of the event to infer the ISF and CIR, which also allows preceding cases to be identified and used in our temporal case retrieval process.

Insulin-on-board (IOB) is a crucial parameter which helps to avoid the negative effects of insulin stacking, caused by administering insulin when some already remains active in the body. To cater for IOB, the *retrieve* step (Section 3.2) uses a temporal approach that factors in preceding bolus doses. This is coupled with an adaptation rule in the *reuse* step (Section 3.4), which resolves differences between the IOB in the problem and the retrieved case(s).

Exercise is a parameter that we believe should be included. However, the UVa/Padova T1DM simulator [16] used in this research did not allow this to be modelled, so it was excluded in the results shown in this paper.

Finally, the *solution* needs to be retained by the case for reuse in solving new problems. The solution in this approach is the bolus dose. This will also serve as a feature in temporal aspect described in the *retrieve* step.

In summary, following the assessment of parameters used by bolus calculators and the properties of the reasoning mechanism, the cases will then be represented by the date and time, carbohydrate intake, preprandial blood glucose level, and the solution of the case will be the recommended bolus.

#### 3.2. Retrieve

The retrieval step is where the temporal aspect is introduced to the system. As opposed to looking at the new problem and previous cases in isolation, we believe the bigger picture should be considered, most notably preceding events. Events that have occurred in the recent past may have an effect on the amount of insulin required now. Whilst the temporal aspect of CBR has been considered previously, none of the previous methods appear to be completely suitable for the task of bolus decision

Parameter	Diabetes				
	Accu-Chek® Aviva Expert	RapidCalc	Personal Calculator	Diabetic Dosage	InsulinCalc
Carbohydrate intake	✓	✓	✓	✓	✓
Preprandial blood glucose	✓	✓	✓	✓	✓
Target blood glucose	✓	✓	✓	✓	✓
Insulin sensitivity factor	✓	✓	✓	✓	✓
Carbohydrate-to-insulin ratio	✓	✓	✓		✓
Insulin-on-board	✓	✓	✓		
Exercise	✓	✓			

Table 1: Parameters used by existing bolus calculators

support. To address this, we propose the use of a *temporal sequence* to describe both new problems and previous cases [12].

Definitions 3.1 through to 3.4 describe the method more formally. In Def. 3.1 and Def. 3.2 a case and the case-base are defined. Where  $c$  represents the new problem or a retained case, and  $CB$  is the case-base. Definitions 3.3 and 3.4 define the temporal problem sequence  $TP$  and temporal case sequence  $TC_n$  respectively.

**Definition 3.1 (Case).** A case  $c$  is a tuple comprised of a number  $n$  of features  $f$ , and a solution  $s$ .

$$c = (f_1, f_2, \dots, f_n, s)$$

**Definition 3.2 (Case-base).** A case-base  $CB$  is a sequence of cases  $c_i$ , where  $i$  ranges from 1 to the size of the case-base  $|CB|$ .

$$CB = \langle c_1, c_2, \dots, c_{|CB|} \rangle$$

**Definition 3.3 (Temporal problem sequence).** A temporal problem  $TP$  is a sequence comprised of the individual new problem proposed to the system  $c'$  and the preceding cases  $c$  in the case-base  $CB$ . The size of  $TP$  is determined by the defined temporal sequence length  $t$ , where  $1 \leq t \leq |CB|$ . A  $TP$  with  $t = 1$  will be a sequence containing the new problem  $c'$ , resulting in traditional CBR retrieval where no previous events are included. For a  $TP$  with  $t > 1$ , the sequence must start from  $t - 2$  less than the size of the case-base, because at the very least the sequence must contain the new problem  $c'$  and the last case in the case-base  $c_{|CB|}$ .

$$TP = \langle c_{|CB|-(t-2)}, c_{|CB|-(t-3)}, \dots, c_{|CB|}, c' \rangle$$

**Definition 3.4 (Temporal case sequence).** A temporal case sequence  $TC_n$  is sequence of  $t$  number of cases  $c$  from the case-base  $CB$  order by date and time, with the sequence ending with case  $n$ . The number of cases in the sequence  $t$  must be equal to  $t$  in the temporal problem sequence  $TP$  (Def. 3.3).

$$TC_n = \langle c_{n-(t-1)}, c_{n-(t-2)}, \dots, c_n \rangle$$

In the retrieval step, the temporal problem sequence  $TP$  (Def. 3.3) is compared to sequences in the case-base  $TC_n$  (Def. 3.4) of the same temporal sequence length  $t$ . To deal with broken sequences - those with assumed missing events (gaps) - the outer fence defined by Tukey is used [17]. Where such gaps exist, the features are replaced by the maximum integer distance of 1 on the scale  $[0, 1]$ .

A weighted distance function is used to compare the similarity of  $TP$  and  $TC_n$ , this helps to ensure that the importance of each feature in the overall similarity is representative of the problem. Weighting each feature in the sequence is a domain dependant task, performed through expert guidance or automated approaches. One such method for automated weighting is the use of *single-attribute evaluators* such as the algorithms *Information Gain*, *Gain Ratio*, *Symmetrical Uncertainty*, *Chi-Squared*, *One Rule* and *RELIEF-F* [18]. These algorithms are described in Section 4.

In this research we have adopted the weighted Euclidean distance function in order to determine similarity

$$d(TP, TC) = \sqrt{\sum_{i=1}^I w_i (TP_i - TC_i)^2} \quad (1)$$

where  $TP$  and  $TC$  is the temporal problem and temporal case sequences respectively,  $d$  is the re-

sulting distance,  $I$  is the total number of features, and  $w$  is the weight of the respective feature.

Prior to computing the distance, all features should be normalised to avoid unwanted bias of features. The closer the distance result is to 0, the closer the similarity.

At the end of the retrieval step the case(s) that are more closely related to the current problem are then reused and further adapted to the current scenario (Section 3.4).

### 3.3. Worked Retrieval Example

To demonstrate the use of temporal sequences, a new problem  $c'$  (Table 2) and case-base  $CB$  (Table 3) are presented for a temporal sequence of length 2.

The temporal sequence is ordered by the date and time of the cases such that the most recent case is the last element of the sequence. The cases are ordered sequentially, for example  $c_2$  occurs after  $c_1$ .  $TC_1$  is omitted as there are no previous cases and so a temporal sequence cannot be formed. Additionally, the new problem solution  $c's$  is not applicable until it has been solved, for this reason the corresponding feature weighting  $ws_2$  is 0.00.

The process begins by first normalising the features  $f$  of the case-base  $CB$  and new problem  $c'$  in Step 3.1. The process of creating the temporal sequences and transforming the sequences in to a new compound case is then illustrated in Step 3.2, with the example feature weights shown in Table 6. Step 3.3 demonstrates the use of the Euclidean distance function with this method.

$c'c$	$c'bg$	$c't$	$c's$
carbo- hydrates (grams)	blood glucose (mmol/L)	time (minutes)	solution (insulin)
60	3.50	160	

Table 2: Example new problem  $c'$

$c_n$	$c_nc$	$c_nbg$	$c_nt$	$c_ns$
case $_n$	carbo- hydrates (grams)	blood glucose (mmol/L)	time (mins)	solution (insulin)
$c_1$	50	4.00	120	4.00
$c_2$	30	6.00	240	2.00
$c_3$	60	5.00	180	4.50

Table 3: Example case-base  $CB$

Normalisation of the case features factors in the feature-values of both the case-base and the new problem. Step 3.1 demonstrates the normalisation of the preprandial blood glucose level. The normalisation process results in the values shown in Tables 5 and 4.

$$f = \langle c'bg, c_1 \cdot cbg, c_2 \cdot cbg, c_3 \cdot cbg \rangle$$

$$= \langle 3.50, 4.00, 6.00, 5.00 \rangle$$

$$f_{min} = 3.50 \quad f_{max} = 6.00$$

$$n(f_1) = \frac{3.50 - 3.50}{6.00 - 3.50} = 0.00$$

$$n(f_3) = \frac{6.00 - 3.50}{6.00 - 3.50} = 1.00$$

$$n(f_2) = \frac{4.00 - 3.50}{6.00 - 3.50} = 0.20$$

$$n(f_4) = \frac{5.00 - 3.50}{6.00 - 3.50} = 0.60$$

### Step 3.1 (Normalisation).

$c_n$	$n(c_nc)$	$n(c_nbg)$	$n(c_nt)$	$n(c_ns)$
case $_n$	carbo- hydrates	blood glucose	time	solution (insulin)
$c_1$	0.67	0.20	0.00	0.80
$c_2$	0.00	1.00	1.00	0.00
$c_3$	1.00	0.60	0.50	1.00

Table 4: Example normalised case-base  $CB$

$n(c'c)$	$n(c'bg)$	$n(c't)$	$n(c's)$
carbohydrates	blood glucose	time	solution
1.00	0.00	0.33	0.00 (n/a)

Table 5: Example normalised new problem  $c'$

In order to improve the retrieval process weights are defined for the new compound cases. Table 6 shows the weights to be used in this example. The weights are normalised using the same method as the features; however, to avoid weights of 0.0, the minimum and maximum ranges of the feature selection algorithm are used instead of the minimum and

maximum weights returned by the feature selection algorithm.

$wc_1$	$wbg_1$	$wt_1$	$ws_1$
carbohydrates	blood glucose	time	solution (insulin)
0.40	0.20	0.10	0.50

$wc_2$	$wbg_2$	$wt_2$	$ws_2$
carbohydrates	blood glucose	time	solution (insulin)
1.00	0.30	0.20	0.00

Table 6: Example normalised weights for the temporal sequence

### Step 3.2 (Create the temporal sequence).

$$\begin{aligned}
TP &= \langle c_3, c' \rangle \\
&= \langle (1.00, 0.60, 0.50, 1.00), (1.00, 0.00, 0.33, 0.00) \rangle \\
TC_2 &= \langle c_1, c_2 \rangle \\
&= \langle (0.67, 0.20, 0.00, 0.80), (0.00, 1.00, 1.00, 0.00) \rangle \\
TC_3 &= \langle c_2, c_3 \rangle \\
&= \langle (0.00, 1.00, 1.00, 0.00), (1.00, 0.60, 0.50, 1.00) \rangle
\end{aligned}$$

### Step 3.3 (Calculate Euclidean distance).

$$\begin{aligned}
d(TP, TC_n) &= \sqrt{\sum_{i=1}^{|TP|} \left( wc_i(TP_i c - TC_i c)^2 + wbg_i(TP_i bg - TC_i bg)^2 + wt_i(TP_i t - TC_i t)^2 + ws_i(TP_i s - TC_i s)^2 \right)} \\
&= \sqrt{wc_1(TP_1 c - TC_1 c)^2 + wbg_1(TP_1 bg - TC_1 bg)^2 + wt_1(TP_1 t - TC_1 t)^2 + ws_1(TP_1 s - TC_1 s)^2 + wc_2(TP_2 c - TC_2 c)^2 + wbg_2(TP_2 bg - TC_2 bg)^2 + wt_2(TP_2 t - TC_2 t)^2 + ws_2(TP_2 s - TC_2 s)^2}
\end{aligned}$$

$$\begin{aligned}
d(TP, TC_2) &= \sqrt{0.40 \times (1.00 - 0.67)^2 + 0.20 \times (0.60 - 0.20)^2 + 0.10 \times (0.50 - 0.00)^2 + 0.50 \times (1.00 - 0.80)^2 + 1.00 \times (1.00 - 0.00)^2 + 0.30 \times (0.00 - 1.00)^2 + 0.20 \times (0.33 - 1.00)^2 + 0.00 \times (0.00 - 0.00)^2} \\
&= 1.23
\end{aligned}$$

$$\begin{aligned}
d(TP, TC_3) &= \sqrt{0.40 \times (1.00 - 0.00)^2 + 0.20 \times (0.60 - 1.00)^2 + 0.10 \times (0.50 - 1.00)^2 + 0.50 \times (1.00 - 0.00)^2 + 1.00 \times (1.00 - 1.00)^2 + 0.30 \times (0.00 - 0.60)^2 + 0.20 \times (0.33 - 0.50)^2 + 0.00 \times (0.00 - 1.00)^2} \\
&= 1.04
\end{aligned}$$

In this example, the Euclidean distance function finds that  $TC_3$  is the nearest neighbour when considering a temporal sequence length of 2 and the weightings from Table 6. The result of the worked example is as expected given the weights applied. However, the distance calculated in both cases is closer than if just the single case was used; since the first cases in the temporal sequences  $TP$  and  $TC_2$  ( $c_3$  and  $c_2$  respectively) share greater similarity than those of the temporal sequences  $TP$  and  $TC_3$  ( $c_3$  and  $c_2$  respectively). However, the method still considers the importance of the primary case of the temporal sequences over preceding cases, with the new problem  $c'$  and last case  $c_3$  of the temporal sequence  $TC_3$  have identical carbohydrates ingested values and carbohydrates ingested has the maximum normalised weight of 1.0. As a result  $TC_3$  is considered to be of greater similarity to  $TP$  than  $TC_2$ .

### 3.4. Reuse

For the reuse step we adopted a simple  $k$ -NN regression strategy to average the bolus prediction

of  $k$  retrieved cases. Equation 2 defines the reuse strategy, where  $k$  is the number of retrieved cases, and  $i_n$  defines the bolus solution provided by a retrieved case.

$$\text{suggested bolus dose} = \frac{1}{k} \sum_{n=1}^k i_n \quad (2)$$

The result is then adapted to resolve differences in the IOB from the new problem to the retrieved cases to further tune the bolus recommendation. Whilst the use of temporal sequences somewhat resolves this issue, it is important to prevent the negative effects of insulin stacking. In this research a linear IOB algorithm (Eq. 3) is adopted [19].

The adapted bolus suggestion is calculated by deducting the average of the sum of the IOBs ( $iob_s$ , Eq. 4) for  $k$  number of retrieved cases from the original bolus suggestion  $i$  to determine the difference  $d'$ , as described in Eq. 6. For Equations 3 - 6 the variables are defined as follows: the case-base  $CB$  is a sequence of cases  $c$ , with each case  $c$  a tuple of case time  $ct$  in minutes and the bolus dose  $ci$ .  $t$  denotes time in minutes,  $pt$  is the time of a new problem in minutes.  $RC$  denotes a sequence of case times in minutes. The active insulin time  $a$  is a constant to reflect the duration of a bolus dose in minutes. The suggested bolus dose  $i$  is the original bolus dose to be adapted.

$$iob(c, t, a) = \begin{cases} ci \times \left(1 - \frac{t - ct}{a}\right), & a > t - ct > 0 \\ 0.0, & \text{otherwise.} \end{cases} \quad (3)$$

$$iob_s(CB, t, a) = \sum_{n=1}^{|CB|} iob(c_n, t, a) \quad (4)$$

$$d(pt, RC, CB, a) = iob_s(CB, pt, a) - \frac{\sum_{n=1}^k iob_s(CB, RC_n, a)}{k} \quad (5)$$

$$d' = \begin{cases} i - d(pt, RC, CB, a), & i - d(pt, RC, CB, a) \geq 0 \\ 0.0, & \text{otherwise.} \end{cases} \quad (6)$$

### 3.5. Revise

The revise step is crucial to allow the system to improve sub-optimal recommendations. The degree of success can be inferred from the difference

between postprandial (post-meal) blood glucose of the subject and their target blood glucose level. If the postprandial reading is equal or close to the target blood glucose level then the recommendation can be considered optimal and no revision is required. However, if the postprandial reading is higher than the target level, the recommended bolus dose should be increased; otherwise if the postprandial reading is lower than the target level, the bolus dose should be decreased.

To determine this, a method for correcting bolus doses described by Eq. 8 is used based on the subject's *Total Daily Dose* (TDD) to estimate the ISF (Eq. 7) [20, 21].

To calculate the ISF, University College London Hospitals suggest the use of the 100 rule for blood glucose measured in mmol/L, the equivalent for mg/dL is the 1800 rule [22]. However, Davidson et al. undertook extensive statistical studies of various constants and suggest the 1,700 rule, which was applied to Active Insulin Management (AIM) system [21, 23]. The 1,700 rule is an established recommendation, and has been used in this research (Eq. 7) [24]. To convert the ISF from mg/dL to mmol/L, the ISF is multiplied by the constant 0.0555.

$$ISF = \frac{1700}{TDD} \times 0.0555 \text{ mmol/L} \quad (7)$$

$$\text{revised bolus} = \frac{pbg - tbg}{ISF} \quad (8)$$

To calculate TDD based on guidelines by University College London Hospitals, the sum of 4 days of bolus and basal insulin doses is divided by four [22]. Equation 9 describes this calculation, where  $I$  represents the sequence of bolus and basal doses over a period of  $d$  days, and  $I_i$  is an individual bolus or basal dose from the sequence of insulin doses  $I$ .

$$TDD = \frac{\sum_{i=1}^{|I|} I_i}{d} \quad (9)$$

In situations where there is not sufficient data to calculate the subject's TDD, then an estimate can be calculated (Eq. 10) using the subject's body weight  $w$  in pounds [23].

$$TDD = 0.24 \times w \quad (10)$$

One difficulty to overcome is when to perform the postprandial blood glucose reading. If it occurs too soon after the dose was administered or too late then the revision is likely to be sub-optimal. To determine this, *in silico* evaluation for 2, 3 and 4-hour

offsets are conducted, with the results presented in Section 6.3.

To determine this, *in silico* results for 2, 3 and 4-hour offsets are presented.

### 3.6. Retain

The retain step of the cycle stores the evaluated recommendation into the case-base for future reuse. The complexity of retaining cases largely depends on how the cases are stored. In this work the case structure remains consistent. However, we are aware of the importance of case-base maintenance to ensure the search space does not grow to cause time-complexity issues, and to prevent bad solutions being retained.

## 4. Discussion and Analysis of Weka’s Feature Selection Algorithms

To determine the similarity between a problem and an existing case, it is necessary to consider the importance of each feature. If the similarity comparison is conducted with each feature having equal importance, the result may not be desirable. Instead, each feature should be weighted in accordance to its ability to correctly predict the outcome. These weights can then be used in the similarity comparison to ensure accurate case retrieval.

Determining a weighting for each feature can be achieved by identifying the relevance between a feature-value and the outcome. An understanding of the domain may tell us that certain features are more important than others in determining the outcome. However, an estimate is not satisfactory, and may vary from subject to subject. Instead, weightings can be identified from information known to the system, which in this instance is available through the case-base. In data mining, the process of feature selection is used to determine which features are required to reliably predict the outcome [25]. This allows features of little or no importance to be ignored, which is useful in big data environments. There are two main approaches to performing feature selection: *subset evaluation* and *single-attribute evaluation* [18].

Subset evaluators aim to identify the smallest subset of features which can successfully predict the outcome [18]. This is usually driven by a random cycle, which depending on the number of features present, may or may not include every combination. In order to prevent excessive computation time, the

process is often limited to a certain number of iterations. Each subset selected is tested to see how reliably the solution can be predicted, and the smallest possible subset which achieves this is returned.

Single-attribute evaluators do not attempt to identify the smallest subset of features, and instead evaluate each feature independently [18]. Each attribute is assigned a numerical result based on its ability to predict the outcome, which allows for the features to be ranked.

Both approaches are suitable for feature selection. However, for the purpose of weighting features only the single-attribute evaluators will be considered. Through performing single-attribute evaluation on each of the features, the results obtained can be used within the distance function for feature weighting. The use of feature selection using single-attribute evaluators is analysed during the experimental phase of this research. Several well-established attribute evaluators within the Weka data mining application will be used to obtain these weightings [3]. Weka provides a number of single-attribute evaluation algorithms such as *Chi-Squared*, *Information Gain*, *Gain Ratio*, *Symmetrical Uncertainty*, *One Rule*, and *RELIEF-F* [18]. Each of these attribute evaluators are described in the proceeding subsections. The performance of the feature selection algorithms for T1DM bolus advice case retrieval is discussed in Section 6.1.2.

### 4.1. Chi-Squared

In Weka, *Chi-Squared* is a single-attribute feature selection algorithm based on the  $\chi^2$  statistic [18]. The algorithm seeks to evaluate the  $\chi^2$  value for a feature in respect to predicting a class by comparing the number of observations to the expected frequency. The  $\chi^2$  value for any feature is defined by Eq. 11, where  $E_{ij}$  is the expected frequency, and  $O_{ij}$  is the observed frequency for class  $i$  with the feature-value  $j$  [26].

$$\chi^2 = \sum_i \sum_j \frac{(E_{ij} - O_{ij})^2}{E_{ij}} \quad (11)$$

The expected frequency  $E_{ij}$  is defined by Eq. 12, where  $n_j$  is the number of instances of the feature with value  $j$ ,  $n_i$  is the number of instances of the class  $i$ , and  $n$  is the total number of instances.

$$E_{ij} = \frac{n_j n_i}{n} \quad (12)$$

Any continuous feature-values should be transformed through discretisation into nominal values



prior to performing the  $\chi^2$  test. Liu and Setiono presented *Chi2*, an automated discretisation and feature selection algorithm using the  $\chi^2$  statistic [27]. *Chi2* is based upon *ChiMerge*, an algorithm designed purely to discretise data [28]. Equation 13 defines the *ChiMerge* algorithm for discretisation, where  $C$  is the total number of classes,  $A_{ij}$  is the number of examples in interval  $i$  of class  $j$ ,  $E_{ij}$  is the expected frequency of  $A_{ij} = R_i \times C_j / N$ ,  $R_i$  is the number of examples in interval  $i = \sum_{j=1}^C A_{ij}$ ,  $C_j$  is the number of examples in class  $j = \sum_{i=1}^2 A_{ij}$ , and  $N$  is the total number of examples  $= \sum_{j=1}^C A_{ij}$  [28, 26].

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^C \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad (13)$$

Weka does not adopt the *Chi2* or *ChiMerge* algorithms prior to applying the  $\chi^2$  statistic for feature selection [18]. Instead, Weka applies a discretisation method proposed by [29]; a minimum entropy heuristic which uses the *Minimum Description Length Principle (MDLP)* to determine useful cut points for discretising the features [30]. Liu et al. found using entropy with *MDLP* to be the best choice for data discretisation when compared to other methods including *Chi2* and *ChiMerge* [31]. The same discretisation filter is also applied to the entropy based feature selection algorithms *Information Gain*, *Gain Ratio*, and *Symmetrical Uncertainty* [18]; algorithms which also favour nominal values [32].

Equation 14 defines the entropy function for discretisation, which is applied recursively until a stop criterion is met. Where  $S_1$  and  $S_2$  are two intervals of set  $S$  bound by cut point  $T$ ,  $A$  is the feature, and  $\text{Ent}(S)$  (Eq. 15) is the entropy for a subset of  $S$ , with  $P(C_i, S)$  the proportion of examples in  $S$  of class  $C_i$  [29, 26].

$$E(A, T; S) = \frac{|S_1|}{|S|} \text{Ent}(S_1) + \frac{|S_2|}{|S|} \text{Ent}(S_2) \quad (14)$$

$$\text{Ent}(S) = - \sum_{i=1}^C P(C_i, S) \log_2(P(C_i, S)) \quad (15)$$

The criterion for stopping discretisation uses *MDLP*, where a partition induced by the cut point  $T$  is only accepted if encoding the partition costs less than the encoding prior to the split. This stop

criterion is defined by Eq. 16, where  $N$  is the number of instances in the set  $S$ , and the distinct classes present in  $S$ ,  $S_1$ , and  $S_2$  is  $c$ ,  $c_1$ , and  $c_2$  respectively [29, 26].

$$\text{Ent}(S) - E(A, T; S) > \frac{\log_2(N-1)}{N} + \frac{\log_2(3^c - 2) - [c\text{Ent}(S) - c_1\text{Ent}(S_1) - c_2\text{Ent}(S_2)]}{N} \quad (16)$$

#### 4.2. Information Gain

*Information Gain* is a feature evaluation approach which uses entropy to evaluate the uncertainty of a feature [33]. This is achieved through evaluating how the inclusion of additional information provided by a feature reduces the entropy. The entropy of a feature  $X$  is determined by Eq. 17, with  $P(x_i)$  being the frequency of value in  $X$  [34, 32].

$$H(X) = - \sum_i P(x_i) \log_2(P(x_i)) \quad (17)$$

The entropy of feature  $X$  with the additional information provided by  $Y$  is calculated by Eq. 18, where  $P(y_j)$  is the frequency of a value in feature  $Y$ , and  $P(x_i|y_j)$  is the frequency of a value of  $X$  given the evidence of a value of  $Y$ . For classification,  $X$  is the class or outcome and  $Y$  is a feature of the problem.

$$H(X|Y) = - \sum_j P(y_j) \sum_i P(x_i|y_j) \log_2(P(x_i|y_j)) \quad (18)$$

*Information Gain* (Eq. 19) is calculated through the reduction in entropy of  $X$  following the information of  $X$  provided by  $Y$ .

$$IG(X|Y) = H(X) - H(X|Y) \quad (19)$$

A limitation of the *Information Gain* algorithm is the ability to only cater for nominal feature-values. To resolve this, continuous data is first partitioned into nominal values. In Weka, this is achieved using the *MDLP* discretisation method.

#### 4.3. Gain Ratio

A limitation of *Information Gain* is the algorithm's preference to select features containing a

large quantity of values [33]. This is due to increased partitioning leading to an increased number of subsets which may only point to a single class. An example of this would be a unique identification feature for which each subset would only contain one case, and subsequently relate to only one class. This results in the maximal information gain  $IG(X|Y) = 1$  despite the information being irrelevant for prediction. To overcome this, an extension of the *Information Gain* algorithm was developed called *Gain Ratio*. *Gain Ratio* attempts to normalise the information gain by factoring in the useful proportion of information.

*Gain Ratio* (Eq. 20) is calculated through the division of the *Information Gain* (Eq. 19) for  $X$  given the evidence of  $Y$  by the entropy of feature  $Y$  (Eq. 17) [26].

$$GR(X, Y) = \frac{IG(X|Y)}{H(Y)} \quad (20)$$

*Gain Ratio* has been shown to be robust and provide consistently better results than *Information Gain*. However, evidence has shown that *Gain Ratio* can favour unbalanced splits when one particular subset is smaller than the others [35].

#### 4.4. Symmetrical Uncertainty

*Symmetrical Uncertainty* was developed to compensate for *Information Gain*'s preference for features with more values, similar to *Gain Ratio* [36, 32]. *Symmetrical Uncertainty* normalises the value to the range  $[0, 1]$ , where 1 implies the knowledge provided by a value of  $Y$  always predicts the value of  $X$ , and 0 implies the values are completely independent of each other. *Symmetrical Uncertainty* is defined by Eq. 21.

$$SU(X, Y) = 2 \times \left[ \frac{IG(X|Y)}{H(X) + H(Y)} \right] \quad (21)$$

#### 4.5. One Rule

*One Rule* was designed with simplicity in mind [37]. The algorithm uses the error rate of a feature as opposed to the entropy approached used by *Information Gain*. Nevill-Manning et al. describe the pseudocode for the *One Rule* algorithm as displayed in Fig. 3 [38].

*One Rule* was tested against C4.5 (used to generate decision trees) on 16 datasets that are regularly used for evaluation [33]. The results showed that despite the simple approach taken, *One Rule* was

```

for each attribute a, form a rule as follows:
  for each value v from the domain of a,
    select the set of instances where a
    has value v.
    let c be the most frequent class in
    that set.
    add the following clause to the rule
    for a:
      if a has value v then the class
      is c
    calculate the classification accuracy of
    this rule.
use the rule with the highest classification
accuracy

```

Figure 3: One Rule pseudocode [38]

only marginally less accurate (by 3.1%) [37]. *One Rule* demonstrates that with most real-world data problems, rules can perform as well as more complex algorithms. In Weka, the *One Rule* feature selection algorithm performs discretisation using the same error rate principle [18].

#### 4.6. RELIEF-F

*RELIEF-F* is an extension of the *RELIEF* instance based feature ranker [39, 40]. *RELIEF* was designed as an efficient method for estimating how the values of features are able to distinguish between instances which are near to each other; these values can either be discrete or continuous. This is achieved through finding a near-hit and near-miss instance. A near-hit is an instance which belongs to the same class and neighbourhood as the instance being evaluated. A near-miss is an instance in the same neighbourhood as the instance being evaluated but is not of the same class. A limitation of the original *RELIEF* algorithm is its ability to only deal with a two class problem, which would not be suitable for this research as there will be more than two classes of bolus insulin dose available. The *RELIEF-F* extension was devised to eliminate this constraint.

*RELIEF-F* is the result of incremental development of the *RELIEF* algorithm, starting with *RELIEF-A* [39]. *RELIEF-A* extended *RELIEF* to include more than one near-hit or near-miss. *RELIEF-B*, *C* and *D* implemented improvements upon *RELIEF-A* for dealing with incomplete datasets. Multi-class problems were introduced with *RELIEF-E* by including near-misses from each class present in the dataset. *RELIEF-F* improved

this through averaging the contribution of the near-misses for each class. This was introduced to allow the algorithm to estimate the ability a feature has to separate two classes without considering if they are closest to each other.

Equation 22 defines *RELIEF-F*, where  $m$  is in the sample size,  $diff(A, R, H)$  is the difference between the values of the features in instance  $R$  and  $H$ ,  $diff(A, R, M(C))$  is the difference between the values of the features in instance  $R$  and the near miss instance  $M(C)$ , and  $P(C)$  is the prior probability of the feature for the class.

$$W(A) := W(A) - \frac{diff(A, R, H)}{m} + \sum_{C \neq class(R)} \frac{P(C) \times diff(A, R, M(C))}{m} \quad (22)$$

#### 4.7. Summary of the Feature Selection Algorithms

This section introduced six feature selection algorithms provided by the Weka data-mining tool. Three of the algorithms - *Information Gain*, *Gain Ratio* and *Symmetrical Uncertainty* - use entropy to determine a features ability to classify. In contrast *Chi-Squared* applies the  $\chi^2$  statistic to predicting a class, *One Rule* applies error rate, and *RELIEF-F* applies a near-hit and near-miss method.

All the algorithms in this section with the exception of *RELIEF-F* require features and classes to be nominal values. As a result, all features with continuous values should be transformed into nominal values prior to applying the discussed feature selection algorithms, with the exception of *RELIEF-F*. Weka applies the entropy function described by Fayyad and Irani with *MDLP* to determine optimal cut points to perform discretisation [29]. This method is used for all feature selection algorithms requiring nominal values with the exception of *One Rule*, which instead applies the same error rate principle used in the feature selection process.

It is expected that all the entropy based algorithms - *Information Gain*, *Gain Ratio* and *Symmetrical Uncertainty* - will result in similar feature weightings. However, it is possible that *Gain Ratio* and *Symmetrical Uncertainty* could produce more accurate weightings than *Information Gain*, as these algorithms improve upon *Information Gain*. The other algorithms - *Chi-Squared*, *One Rule* and *RELIEF-F* - all apply different methods to feature

selection, and the results are likely to differ for each algorithm. In terms of execution speed, the *One Rule* algorithm is likely to outperform the other algorithms due to its simplistic approach. The optimal algorithm for this system is difficult to predict and is evaluated during analysis and testing of the proposed CBR retrieval algorithm in Section 6.1.1.

## 5. Data Generation and Statistical Measures

This section describes the data and statistical measures used for evaluating the methodology discussed in Section 3. We begin by briefly describing the T1DM simulator used for both the generation of data and *in silico* evaluation of the proposed methodology. This is followed by a description of how we generated sample case-base and problem set data for the purpose of evaluation. Finally, we discuss the statistical measures identified for evaluating the effect of bolus recommendations on simulated blood glucose levels.

### 5.1. UVa/Padova T1DM Simulator

The FDA-approved UVa/Padova T1DM simulator is used in this research to create test data and to evaluate the proposed CBR model [16]. The simulator allows blood glucose management to be assessed using *closed-loop* and *open-loop* control algorithms. The closed-loop algorithm acts as an artificial pancreas, where bolus insulin is automatically calculated and dosed to the subject as required [41]. In contrast, the open-loop algorithm is reliant on the simulated subject administering the bolus insulin themselves.

Data can be simulated in two ways. One method is through the user interface, which allows five meals at pre-defined times to be input over the course of multiple days. Alternatively, simulation can be conducted using a scenario file, which is a text file that allows various parameters of the simulation to be set with greater flexibility. Parameters include meal times and carbohydrates, simulation length, bolus times and dose, and closed-loop or open-loop control. The only inputs required for a meal to be simulated using closed-loop control is the quantity of carbohydrates in grams, and the time of the meal. When using open-loop control, the bolus insulin dose and time must also be specified. The results of the simulation can be exported as minute by minute information on signals such as bolus doses, basal rate, and blood glucose level.

This allows for a continuous blood glucose reading to be obtained for the purpose of analysis and evaluation.

The simulator serves two important purposes for the research and development of our CBR system. First, the simulator allows for the production of datasets in order to create sample case-bases and problem sets through closed-loop simulation. Second, the simulator enables bolus suggestions obtained through CBR to be analysed and evaluated using open-loop simulation. The simulator does have some limitations for this research, including the inability to model physical activity, stress, time period based insulin sensitivity factors, and time period based carbohydrate-to-insulin ratios.

### 5.2. Generation of Data Sets for Evaluation

To conduct testing of our method, we require sample case-bases and sets of problems and the computation of the feature weights using the feature selection algorithms. The sample case-bases and problem sets used for evaluation are as follows:

- 5 case-bases, each containing cases over a period of 6 months (Section 5.2.1)
- 5 problem sets, each containing problems over the period of 1 month (Section 5.2.2)

#### 5.2.1. Generation of the Sample Case-bases

Test runs of the simulator determined that 6 months of meals can be simulated reliably at any one time. As a result we chose to limit the size of the sample case-bases to 6 months of cases. To generate the case-bases, 6 months of meal information are created. These meals are generated randomly with varying time intervals between 120 and 420 minutes, and carbohydrate intakes of 0 to 240 grams [42]. A rule is imposed on these generated meals to make the earliest meal of the day occur on or after 7 a.m., this is in order to prevent meals occurring when the subject is likely to be asleep. The blood glucose level and bolus insulin dose for each case are obtained from the simulation of the meal information. These generated meals provide a wide variety of realistic meal patterns over a 6 month period, with each case-base containing an average of 821 cases.

Following simulation of the meal information, the simulated blood glucose and bolus insulin signals require merging with the corresponding meal inputs of carbohydrates and time. These blood glucose and bolus insulin values are extracted from the

minute by minute signals exported following simulation using the time the meal occurred during simulation.

#### 5.2.2. Generating Problem Sets

The same process for generating the sample case-bases is used to generate sample problem sets. The problem sets contain 1 month of randomly generated meals using the same method applied to generating the sample case-bases discussed in Section 5.2.1. Each problem set is simulated to obtain realistic blood glucose levels for the problem, and the blood glucose signal exported and merged to complete the problems.

#### 5.2.3. Acquiring Feature Weights Through Feature Selection Algorithms

Each generated sample case-base is run through the six feature selection algorithms (*Chi-Squared*, *Information Gain*, *Gain Ratio*, *One Rule*, *RELIEF-F*, and *Symmetrical Uncertainty*) in Weka to acquire feature weights for use in the retrieval similarity measure. Prior to importing the case-bases into Weka, the case-bases are formatted to represent temporal sequences of up to 5 in length. Each of the six feature selection algorithms are then performed on the case-bases and the resulting weights extracted.

### 5.3. Statistical Measures for Evaluating Bolus Insulin Suggestions

To evaluate the CBR system, statistical measures needed to be identified for analysing and evaluating the bolus advice obtained through case retrieval. The data available for testing is limited to information about the meals themselves and continuous blood glucose data retrieved from the simulator. For testing purposes, the blood glucose data provides the best representation of the subject's well-being, as T1DM management fundamentally revolves around maintaining safe blood glucose levels. The continuous blood glucose data is generated through open-loop simulation of the meal information input into the CBR system alongside the bolus insulin suggested by the reuse of retrieved cases. This continuous blood glucose data provides a simulated blood glucose reading for every minute of the simulation.

Several statistical measures have been identified for use on continuous blood glucose data [43]: blood glucose risk index (BGRI), low blood glucose

risk index (LBGI), high blood glucose risk index (HBGI), time within target blood glucose range, mean, and standard deviation.

### 5.3.1. Blood Glucose Risk Index

The BGRI can be applied to continuous blood glucose data to determine overall variance of LBGI and HBGI [44, 45, 43]. LBGI is used as an early indicator for detecting potential hypoglycaemic events, whilst HBGI is used as an indicator of hyperglycaemic events. These risk indexes are obtained by splitting the data into low and high glucose values to assess the variance independently of each other. Defined boundaries (Table 7) have been outlined for LBGI and HBGI values to determine the risk level [44, 45, 46].

Risk level	LBGI	HBGI
Minimal	$x \leq 1.1$	$x < 5.0$
Low	$1.1 < x \leq 2.5$	$5.0 \leq x \leq 10.0$
Medium	$2.5 < x \leq 5.0$	$10.0 < x \leq 15.0$
High	$x > 5.0$	$x > 15.0$

Table 7: Low and high blood glucose risk index severity levels [46]

The LBGI and HBGI are calculated by firstly applying Eq. 23 for blood glucose readings  $bg$  in mmol/L and Eq. 24 for blood glucose readings  $bg$  in mg/dL.

$$f(bg) = 1.509 \times (\ln(bg)^{1.084} - 5.381) \quad (23)$$

$$f(bg) = 1.509 \times (\ln(18 \times bg)^{1.084} - 5.381) \quad (24)$$

The risk function  $r(bg)$  for each blood glucose is calculated by Eq. 25 of the readings.

$$r(bg) = 10 \times f(bg) \quad (25)$$

The results of the risk function are split into two branches  $rl$  for low blood glucose readings (Eq. 26), and  $rh$  for high blood glucose readings (Eq. 27).

$$rl(bg) = \begin{cases} r(bg), & \text{if } f(bg) < 0 \\ 0, & \text{otherwise.} \end{cases} \quad (26)$$

$$rh(bg) = \begin{cases} r(bg), & \text{if } f(bg) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

To calculate LBGI and HBGI, two sequences of low blood glucose readings  $LBG$  and high blood glucose readings  $HBG$  are firstly determined by Eq. 26 and Eq. 27. LBGI is then calculated for the sequence of low blood glucose readings  $LBG$  using Eq. 28, and HBGI is calculated for the sequence of high blood glucose readings  $HBG$  using Eq. 29.

$$LBGI(LBG) = \frac{1}{|LBG|} \sum_{i=1}^{|LBG|} LBG_i \quad (28)$$

$$HBGI(HBG) = \frac{1}{|HBG|} \sum_{i=1}^{|HBG|} HBG_i \quad (29)$$

Finally, the overall BGRI is calculated through the sum of LBGI and HBGI as shown in Eq. 30.

$$BGRI = LBGI + HBGI \quad (30)$$

Lower values of LBGI and HBGI indicate a lower risk of hypoglycaemic and hyperglycaemic events respectively as illustrated by Table. 7. BGRI determines the overall risk of both hypoglycaemic and hyperglycaemic events, with lower values indicating a reduced chance of either hypoglycaemic or hyperglycaemic events occurring.

### 5.3.2. Time Within Target Blood Glucose Level Range

The percentage of time the subject spends within a pre-defined target range is a good indicator that the solutions provided by the CBR system are safe for the subject. The standard target range for T1DM subjects is between 4 mmol/L and 10.0 mmol/L [47, 48]. The higher the percentage of time spent within the target range, the better the blood glucose control. However, the implications of extreme highs and lows are not represented by this measure.

### 5.3.3. Mean Blood Glucose Level

The mean ( $\mu$ ) of the blood glucose data provides a simple approach to determining the subject's overall well-being, but does not aid in representing the variance, extreme lows, and extreme highs. The optimal target mean blood glucose is dependent upon the subject's target blood glucose level, which discounting the 2 hours following a meal is between 4 mmol/L and 7 mmol/L [48].

### 5.3.4. Standard Deviation

Calculating the standard deviation ( $\sigma$ ) of the continuous blood glucose data allows the overall stability of the subject’s blood glucose to be assessed [47]. A high standard deviation indicates that the subject’s blood glucose levels are varying greatly over time. In contrast, a low standard deviation represents improved stability, and if the subject is also within the target range, good blood glucose control. Low deviation combined with low or high blood glucose readings would indicate that the subject is consistently outside the safe zone.

## 6. Results

In this section we describe the *in silico* results of the approach outlined in Section 3 using the data sets discussed in Section 5. The results obtained have been broken down into three sections for the *retrieve*, *reuse* and *revise* steps of the CBR cycle in order to highlight how our approach help to progressively improve the decisions made by the system.

### 6.1. Retrieve

This section describes the *in silico* results for the retrieval method described in Section 3.2. First, a comparison of how different temporal sequence lengths affected the statistical measures. Second, how the feature selection algorithms effected the decisions made. Finally, a comparison of our method against two other methods: closed-loop simulation and a state-of-the-art bolus calculator.

#### 6.1.1. Temporal Sequences

We present the percentage reduction in BGRI of the different temporal sequence lengths (TS2 - TS5) in comparison to a single temporal case sequence (TS1 - where no preceding cases are included) for all feature selection algorithms in Table 8 and Figure 4; where the highest percentage reduction in BGRI result is best. Overall, retrievals with a temporal sequence length of 1 (TS1) resulted in the poorest predictions, shown by TS2 - TS5 resulting in a BGRI reduction for all feature selection algorithms used. This provides some evidence that including preceding events helps to improve predictions. The results indicate a decline in BGRI reduction as the length of the temporal sequences increases. However, the largest temporal sequence evaluated (TS5) demonstrated varied retrieval results depending on

the feature selection algorithm used, this is discussed in more detail in Section 6.1.2.

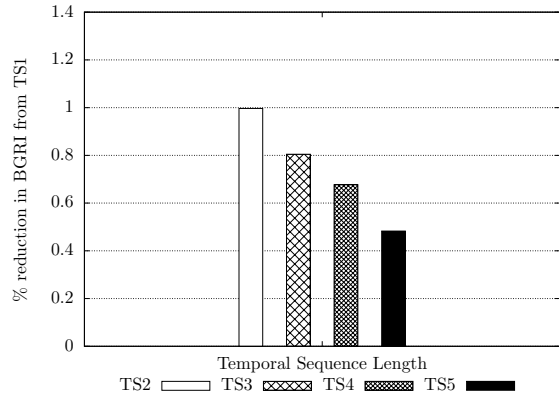


Figure 4: Comparison of temporal sequence lengths to no temporal sequence (TS1), optimising for minimal BGRI, averaged over all feature selection algorithms

#### 6.1.2. Feature Selection Algorithms

The results for the six feature selection algorithms used in isolation are presented in Figure 5. A notable observation is the improvement in prediction for larger temporal sequences (up to 5) when using the *One Rule* algorithm. This algorithm considers features from previous cases to have more significant weight than the other algorithms which explains difference in these results. For the T1DM domain *One Rule* provided favourable results for greater temporal sequence lengths up to length 5, however this may not be generalised to other domains.

The similar results obtained using the entropy based algorithms *Information Gain*, *Gain Ratio* and *Symmetrical Uncertainty* are expected since *Gain Ratio* and *Symmetrical Uncertainty* are evolutions of the *Information Gain* algorithm designed to reduce potential bias for some features [33, 32]. On average, *Gain Ratio* resulted in the highest BGRI reduction where  $t = 2$ . Additionally, the results also show *Chi-Squared* provides similar but slightly poorer outcomes to the entropy based algorithms.

Overall *RELIEF-F* resulted in lowest BGRI reduction of all the feature selection algorithms used. One exception is where a single preceding case is involved (TS2), with *RELIEF-F* outperforming *One Rule*, but still falling behind the other four algorithms. In instances where  $t > 2$ , *RELIEF-F* results in the lowest percentage reduction in BGRI

Feature Selection Algorithm	TS2	TS3	TS4	TS5
Chi-Squared	1.07%	1.02%	0.58%	0.29%
Information Gain	1.21%	0.95%	0.74%	0.23%
Gain Ratio	1.26%	1.11%	0.83%	0.49%
One Rule	0.52%	0.60%	0.81%	1.23%
RELIEF-F	0.72%	0.12%	0.26%	0.26%
Symmetrical Uncertainty	1.20%	1.03%	0.84%	0.39%

Table 8: Percentage reduction in BGRI for different temporal sequence lengths to no temporal sequence (TS1)

from TS1.

In summary, we see positive outcomes for all feature selection algorithms used. *Information Gain*, *Gain Ratio*, *Symmetrical Uncertainty* and *Chi-Squared* demonstrate a similar pattern of BGRI reduction in correlation to the temporal sequence length, exhibiting the highest reduction when a single preceding case is factored into the retrieval step (TS2). In contrast, *One Rule* demonstrates the opposite behaviour, improving BGRI reduction as the temporal sequence length increases up to 5. The results indicate that *Gain Ratio* with temporal sequence length of 2 results in the highest percentage reduction in BGRI (1.26%) in comparison to TS1; however, other algorithms with the exceptions of *RELIEF-F* and *Chi-Squared* are capable of producing similar BGRI reductions depending on the length of the temporal sequence.

### 6.1.3. Comparison to Other Methods

In this section the continuous blood glucose results from CBR retrieval are compared to bolus suggestions through closed-loop simulation, and the formula used by the Accu-Chek<sup>®</sup> Aviva Expert bolus calculator. Both the closed-loop simulation and bolus calculator were applied to the same five problem sets used to evaluate CBR retrieval. The bolus calculator results were computed and then simulated using open-loop simulation to obtain continuous blood glucose levels. The results are compared to the optimal retrieval configuration, and are visualised in Fig. 6.

The mean statistical results across all five problems sets using closed-loop simulation can be seen in Table 9. As the sample case-bases were created using the simulator it is expected that the CBR results would be similar if the retrieval algorithm can correctly identify similar cases. The BGRI, LBGi and HBGI continuous blood glucose results for the CBR suggestions and closed-loop simulation (Fig. 6) yield, as expected, similar results. This provides

evidence that the retrieval algorithm is identifying appropriate cases within the case-base.

Measure	Value
BGRI	4.34
LBGI	2.11
HBGI	2.20
<TR %	0.12
>TR %	0.00
$\sigma^2$	0.79
$\sigma$	0.89
$\mu$	6.37

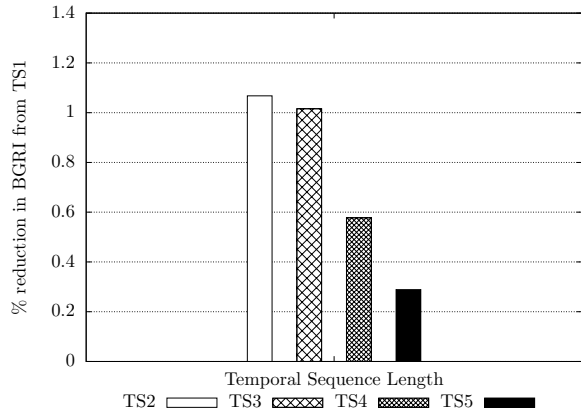
Table 9: Closed-loop simulation statistics

Table 10 displays mean continuous blood glucose results for the bolus calculator advice. The results show that the bolus calculator outperforms both the simulator and the CBR retrievals. This provides evidence that the formulas used by the state-of-the-art bolus calculators can produce reliable and accurate results. These results provide reassurance, as one potential method of building the initial case-base for the subject is through utilising an existing formula.

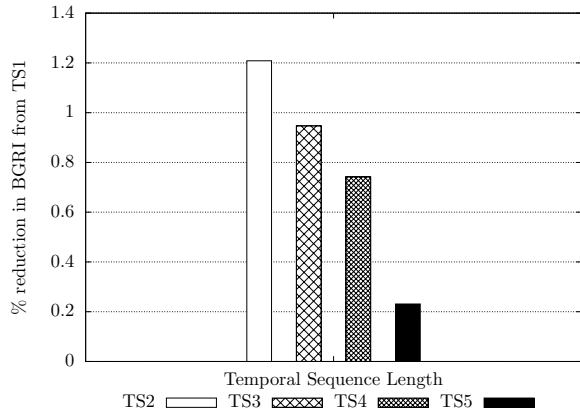
Measure	Value
BGRI	4.21
LBGI	1.92
HBGI	2.29
<TR %	0.00
>TR %	0.00
$\sigma^2$	0.76
$\sigma$	0.87
$\mu$	6.49

Table 10: Bolus calculator statistics

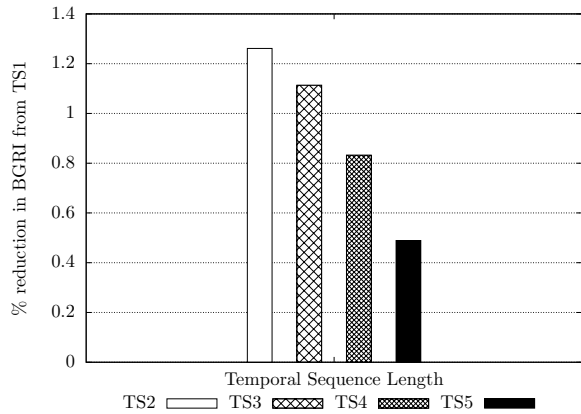
The state-of-the-art bolus calculator results question whether performing CBR retrieval on a case-base produced by the calculation formula will result in similar performance. To test this, the op-



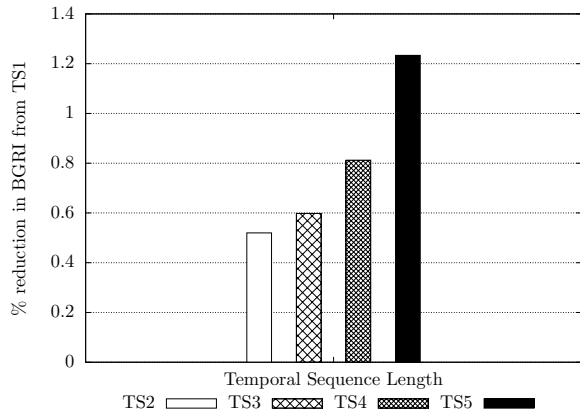
(a) Chi-Squared



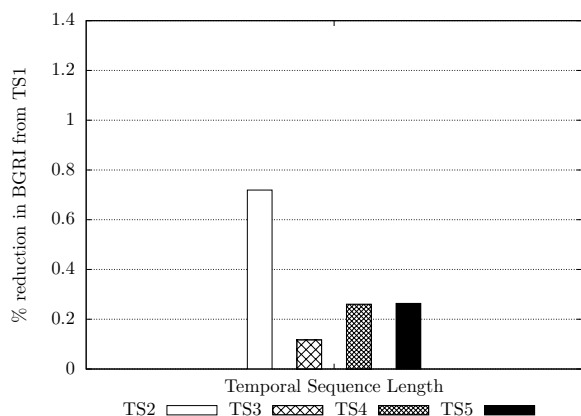
(b) Information Gain



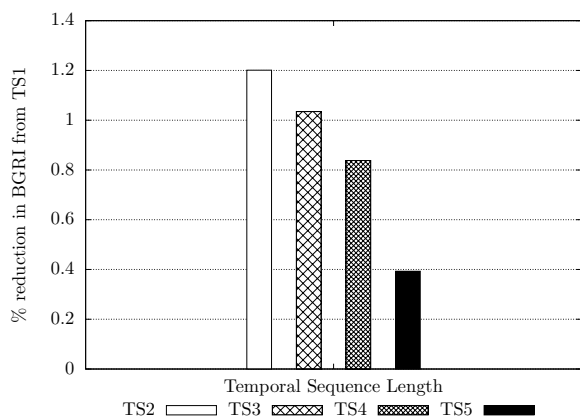
(c) Gain Ratio



(d) One Rule



(e) RELIEF-F



(f) Symmetrical Uncertainty

Figure 5: Comparison of various temporal sequence lengths to no temporal sequence (TS1) for the different feature selection algorithms



timal retrieval configuration is tested against case-bases produced by the bolus calculator as opposed to closed-loop simulation.

The case-bases were identical to those used previously, but the bolus solutions are replaced by those obtained from the bolus calculator. The results shown in Table 11 demonstrate similar CBR retrieval results to those produced by the bolus calculator, with a marginal improvement in some statistical measures. This is visualised in Fig. 7 where the original results are shown in addition to CBR retrieval on the bolus calculator case-base.

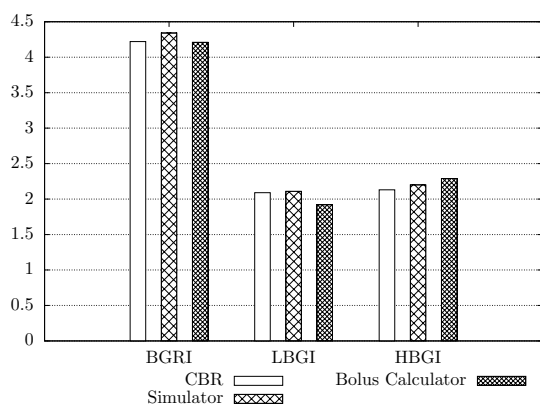


Figure 6: Comparison of CBR, simulated and bolus calculator results

Measure	Value
BGRI	4.14
LBGI	1.87
HBGI	2.26
<TR %	0.00
>TR %	0.00
$\sigma^2$	0.74
$\sigma$	0.86
$\mu$	6.48

Table 11: CBR retrieval result for case-bases produced by a bolus calculator

The comparison of the CBR retrieval results to the other methods show evidence that the retrieval method discussed is capable of selecting reliable bolus advice when presented with a new problem. The results also show that the quality of the CBR suggestions is related to the quality of the cases retained in the case-base.

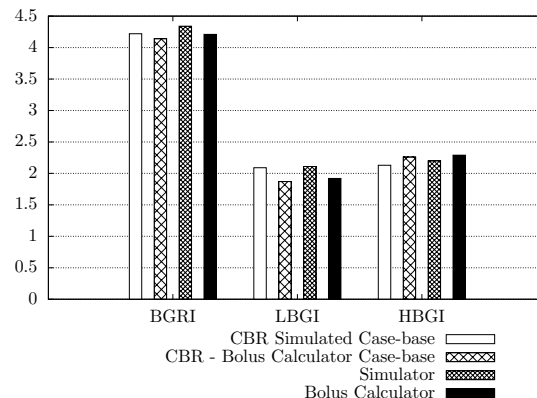


Figure 7: Comparison of CBR to other methods including CBR with bolus calculator case-base

Measure	Without IOB	With IOB
BGRI	$4.22 \pm 0.31$	$3.94 \pm 0.27$
LBGI	$2.09 \pm 0.23$	$1.96 \pm 0.23$
HBGI	$2.13 \pm 0.16$	$1.98 \pm 0.26$
< TR %	$0.03 \pm 0.19$	$0.01 \pm 0.12$
> TR %	$0.00 \pm 0.00$	$0.00 \pm 0.00$
$\sigma$	$0.87 \pm 0.05$	$0.81 \pm 0.04$
$\mu$ mmol/L	$6.34 \pm 0.13$	$6.30 \pm 0.21$

Table 12: Comparison without and with insulin-on-board adaptation

## 6.2. Reuse

Insulin-on-board adaptation was tested against a combination of five case-base sets using the optimal retrieval configuration. The purpose of the IOB adaptation is to resolve the differences in active insulin between the new problem and retrieved case(s). Table 12 and Figure 8 illustrate the improvement IOB adaptation provides across all statistical measures.

The results show some improvement in overall blood glucose control with IOB adaptation, with a 6.6% reduction in BGRI, in addition to a reduction in deviation. The decrease in both LBGI and HBGI suggests that the adaptation rule is able to correctly decrease or increase the bolus insulin suggestion to account for differences between the IOB of the new problem and retrieved case(s). This provides evidence that including this adaptation rule in the reuse step of the CBR system does help to improve the bolus advice. As the results only show a slight improvement from the pre-adaptation solution, it provides an indication that temporal se-

quences help to reduce the effect of insulin stacking. Since the goal is to seek the optimal solution, any adjustment which can improve the suggestion is beneficial to the system.

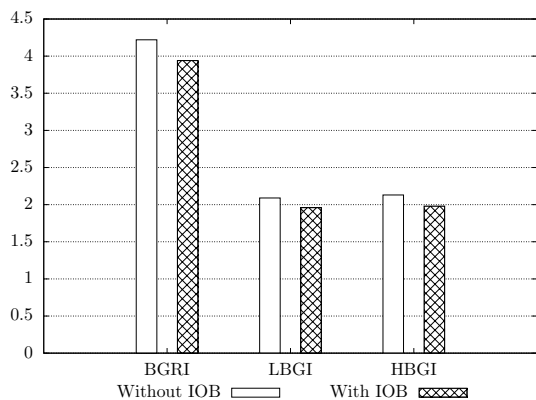


Figure 8: BGRI, LBG1 and HBGI before and after IOB adaptation

### 6.3. Revise

Successful revision is crucial for CBR to learn from mistakes. To test the effectiveness of the revision rule described in Section 3.5, the bolus insulin solutions following IOB adaptation were subject to five cycles of postprandial evaluation. Three sets of results are recorded for postprandial blood glucose readings taken at 2, 3 and 4 hours after the bolus dose. The test cycles aim to determine the effect on the simulated continuous blood glucose level using same the statistical measures to evaluate the retrieval process. The target blood glucose for the subject is set as 6.66 mmol/L, and TDD used to estimate ISF is calculated using Eq. 9 over 4 preceding days, where each day has at least three meals recorded. A daily basal dose of 20 insulin units is added to each of the 4 days, as defined for the subject by the simulator. For instances where the proceeding bolus dose occurs before the postprandial evaluation offset time, the offset is adjusted to 15 minutes prior to the time of the proceeding bolus dose.

Results for five cycles of postprandial blood glucose evaluation and revision using readings 2 hours post meal bolus are displayed in Table 13. The results display gradual improvements in all statistical measures for each evaluation cycle with the exception of HBGI, which shows a slight rise following the fourth cycle, but is still well below the minimal

risk boundary of less than 5.0. From observing the results of increased offset times (discussed below), the most likely cause for this slight rise in HBGI is due to the short offset time. The mean blood glucose level statistic demonstrates how the revised bolus insulin doses on average result in a continuous blood glucose level closer to the target blood glucose level of 6.66 mmol/L.

Table 14 displays postprandial evaluation results for blood glucose readings taken 3 hours post meal bolus. Increasing the postprandial evaluation offset time to 3 hours shows improvements to all statistical measures where an improvement can occur in comparison to the 2-hour offset. Most notably, LBG1 is reduced to the minimum risk category ( $\leq 1.1$ ) after three cycles of evaluation. The increased offset time also removed the rise in HBGI observed in the 2-hour offset.

The final postprandial evaluation uses a 4-hour offset and is presented in Table 15. The results show a slight degradation in blood glucose control in comparison to the 3-hour offset, this is likely due to duration of active insulin in simulation.

Visualisation of the postprandial evaluation revision results for BGRI, LBG1, HBGI and standard deviation are displayed in Figures 9. The trend graphs illustrate the improvements in blood glucose control with increased learning cycles. Most notably, the 3-hour offset produces improved blood glucose control over both the 2-hour and 4-hour offsets. Cycle 0 in the trend graphs represent the *original* continuous blood glucose results prior to postprandial evaluation.

The postprandial evaluation results indicate that the postprandial blood glucose offset time is key to the quality of the bolus revision, with an offset time of 3 hours demonstrating the best revision results over 2-hour and 4-hour offsets. With a 3-hour offset, a decrease in BGRI of 27% is observed after three cycles of revision, pushing up to 28.7% after five cycles. This demonstrates the importance of revision to CBRs ability to learn, as shortcomings of the solution can only be observed through simulation or real-world use. The results demonstrate that an automated approach to bolus dose revision is possible within our CBR system, allowing the user to quickly evaluate the solution and obtain improved bolus insulin suggestions in future reuse.

Measure	Original	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
BGRI	3.94 $\pm$ 0.27	3.50 $\pm$ 0.30	3.29 $\pm$ 0.32	3.21 $\pm$ 0.32	3.18 $\pm$ 0.34	3.17 $\pm$ 0.34
LBGI	1.96 $\pm$ 0.23	1.59 $\pm$ 0.24	1.40 $\pm$ 0.25	1.31 $\pm$ 0.24	1.27 $\pm$ 0.23	1.25 $\pm$ 0.23
HBGI	1.98 $\pm$ 0.26	1.91 $\pm$ 0.31	1.89 $\pm$ 0.20	1.89 $\pm$ 0.29	1.91 $\pm$ 0.19	1.92 $\pm$ 0.20
< TR %	0.01 $\pm$ 0.12	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
> TR %	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
$\sigma$	0.81 $\pm$ 0.04	0.74 $\pm$ 0.04	0.71 $\pm$ 0.04	0.69 $\pm$ 0.04	0.69 $\pm$ 0.04	0.69 $\pm$ 0.04
$\mu$ mmol/L	6.30 $\pm$ 0.21	6.40 $\pm$ 0.17	6.46 $\pm$ 0.15	6.50 $\pm$ 0.15	6.53 $\pm$ 0.14	6.54 $\pm$ 0.14

Table 13: Bolus reuse following 2-hour offset postprandial evaluation

Measure	Original	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
BGRI	3.94 $\pm$ 0.27	3.32 $\pm$ 0.31	3.02 $\pm$ 0.41	2.87 $\pm$ 0.43	2.82 $\pm$ 0.44	2.81 $\pm$ 0.41
LBGI	1.96 $\pm$ 0.23	1.41 $\pm$ 0.17	1.14 $\pm$ 0.14	1.00 $\pm$ 0.14	0.95 $\pm$ 0.16	0.94 $\pm$ 0.15
HBGI	1.98 $\pm$ 0.26	1.91 $\pm$ 0.25	1.88 $\pm$ 0.28	1.87 $\pm$ 0.29	1.87 $\pm$ 0.28	1.87 $\pm$ 0.28
< TR %	0.01 $\pm$ 0.12	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
> TR %	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
$\sigma$	0.81 $\pm$ 0.04	0.72 $\pm$ 0.03	0.67 $\pm$ 0.04	0.65 $\pm$ 0.04	0.64 $\pm$ 0.04	0.64 $\pm$ 0.04
$\mu$ mmol/L	6.30 $\pm$ 0.21	6.44 $\pm$ 0.15	6.52 $\pm$ 0.13	6.56 $\pm$ 0.12	6.58 $\pm$ 0.12	6.58 $\pm$ 0.12

Table 14: Bolus reuse following 3-hour offset postprandial evaluation

Measure	Original	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
BGRI	3.94 $\pm$ 0.27	3.33 $\pm$ 0.26	3.04 $\pm$ 0.37	2.92 $\pm$ 0.41	2.89 $\pm$ 0.40	2.89 $\pm$ 0.37
LBGI	1.96 $\pm$ 0.23	1.41 $\pm$ 0.19	1.14 $\pm$ 0.13	1.02 $\pm$ 0.14	0.99 $\pm$ 0.14	0.99 $\pm$ 0.14
HBGI	1.98 $\pm$ 0.26	1.93 $\pm$ 0.25	1.90 $\pm$ 0.27	1.90 $\pm$ 0.27	1.90 $\pm$ 0.27	1.90 $\pm$ 0.37
< TR %	0.01 $\pm$ 0.12	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
> TR %	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
$\sigma$	0.81 $\pm$ 0.04	0.72 $\pm$ 0.03	0.68 $\pm$ 0.04	0.66 $\pm$ 0.04	0.65 $\pm$ 0.04	0.65 $\pm$ 0.04
$\mu$ mmol/L	6.30 $\pm$ 0.21	6.43 $\pm$ 0.16	6.50 $\pm$ 0.14	6.53 $\pm$ 0.13	6.54 $\pm$ 0.13	6.54 $\pm$ 0.13

Table 15: Bolus reuse following 4-hour offset postprandial evaluation

## 7. Threats to Validity

We have only applied this approach for temporal sequences in case-based reasoning to insulin advice for diabetic subjects. This, along with other threats raises some potential issues as far as the validity of the approach concerned.

For construct validity we must ask if the BGRI measure is the right one to use. However, this is the de-facto standard measure for blood glucose risk levels, so we believe it is the best metric in this situation.

For internal validity we ask whether we achieved a good outcome as a result of chance. This is unlikely since the work took many years with the results being averages of several sample problem and case-base sets. The results were also compared against a control without temporal sequences.

As far as external validity is concerned, which asks whether we can generalise our results, the approach outlined in Section 3 requires evaluation

with different case-bases in other domains. With all feature weighting algorithms temporal sequences up to a length of 5 resulted in improved insulin predictions when compared no use of temporal sequences ( $t = 1$ ). In other domains factors from previous cases may also need to be tested before temporal sequences can be validated as a prediction aid.

Another threat to external validity is the need for continuous features. Whilst distance functions which cater for nominal values do exist, the approach cannot be applied in domains where retrieval is based upon heuristics or abstraction rules. The approach is also aimed towards case-bases where the features remain the same throughout the case-base; there is a potential difficulty to overcome in domains where the features may vary from case to case such as in cross-domain systems. The choice of temporal sequence length is likely to be highly domain dependant and may require trial and error following an initial hypothesis to determine the optimal length.

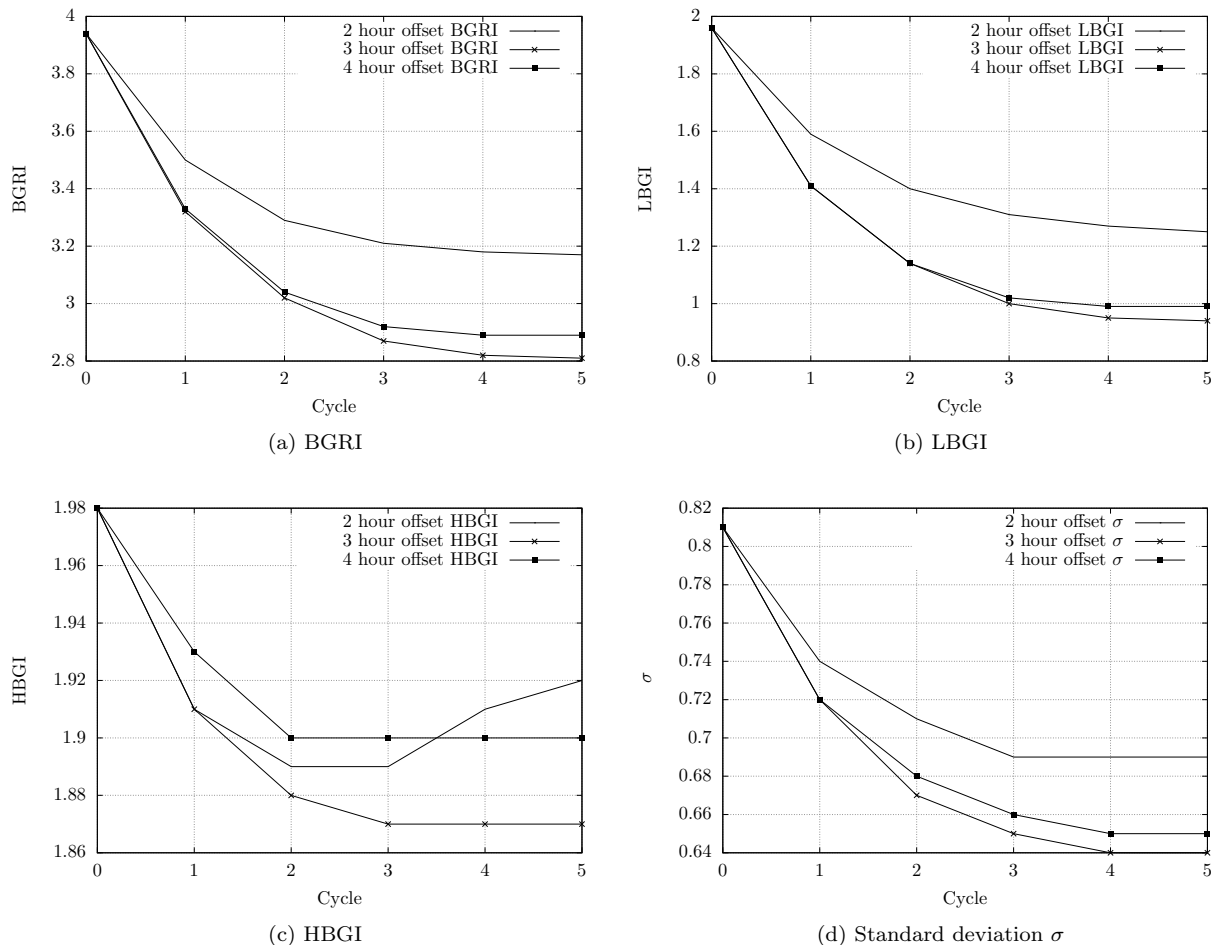


Figure 9: Results of reusing revised bolus suggestions over 0 to 5 cycles of revision

Finally, the approach is heavily reliant on feature weighting to ensure that all the features in the concatenated temporal case reflect their true weight in identifying the most similar case(s). As a result, the selection of a suitable feature weighting algorithm or expert advice is crucial to ensure that features are represented correctly.

## 8. Related Work

Case-based reasoning has been adopted by several research projects in the domain of T1DM. The majority of this research has focused on aiding clinicians with therapy adjustments as opposed to helping the patient directly as our work does. Such projects include the T-IDDM project [49], and more recently the IDSDM project [50]. A notable exception is the Advanced Bolus Calculator for Dia-

betes (ABC4D) [51, 52], which through clinical trials demonstrated the benefits of CBR for bolus advice. ABC4D's CBR approach involves the tuning of ISF and CIR for a small set of meal scenarios. The ISF and CIR are values from the most similar case are then plugged into a standard bolus calculator to suggest a bolus dose. Our approach instead retains all successful cases and derives the bolus suggestion from the most similar cases, allowing the introduction of the temporal retrieval algorithm presented in this work. This may introduce time complexity issues as the case-base grows; however, structures such as a  $k$ -dimensional tree can be introduced and case-base maintenance rules established to minimise this problem [53].

Several different approaches for temporal case-based reasoning have been explored, with the ma-

jority focusing on the prediction of future events. Jaczynski proposed a CBR framework with the ability to retrieve cases composed of time series features [10]. Each time series represents the variants of a variable over time, catering for both numerical samples and changes of state. This work presents a general framework for time-extended situations and focuses on domains with continuous data streams from multiple sources (e.g. sensors) as opposed to the recorded instances of a traditional CBR system.

Jære, et al. investigate the use of Allen’s theory of temporal intervals to avoid faults through prediction in the CBR system CREEK [11, 54]. In contrast to creating a sequence of events from concrete cases, CREEK uses interval relationships to form a temporal relationship.

Another approach to temporal reasoning is temporal projection, introduced by Branting and Hastings [55]. Temporal projection shifts retrieved cases forwards or backwards through a method such as simulation to match the new problem. The case with the greatest similarity following a shift is then selected. This approach represents cases very differently and uses time shifts instead of temporal sequences.

The aforementioned methods do not cater for the formation of temporal sequences from isolated cases. To overcome this Sánchez-Marré, et al. propose that a sequence of continuous cases can be merged into a singular case [12]. This method allows the temporal sequences to be compared using standard distance metrics without the need for additional rules. In the work conducted by Sánchez-Marré, et al., plausible episodes are generated from a new problem, which are then compared to similar retrieved episodes in order to solve the new problem [12]. This work uses a similar approach with the formation of episodes and serves as the foundations for our work.

## 9. Conclusion

Successful management of T1DM is a difficult task for subjects due to the complexity of the condition. As a result, subjects often seek methods to aid their ability to successfully manage the condition through good blood glucose control. Successful management of blood glucose levels reduces the risk of long-term complications.

State-of-the-art bolus calculators for T1DM provide a means for bolus advice, but are reliant on

continuous medical advice from a doctor for optimal configuration. This research demonstrates positive *in silico* results for the use of temporal CBR as an alternative for bolus decision support.

The temporal CBR method proposed in this research enhances the retrieval step through the introduction of temporal sequences and dynamic feature weighting, resulting in better case retrieval in comparison to the traditional approach of single case comparison. Temporal sequences allow factors from previous events to be considered when identifying the most similar case, which is an important consideration in temporal domains. Additionally, the research utilises domain specific rules to enable automatic adaptation and revision, allowing the system to both improve suggestions and optimise future advice. Results of these domain specific adaptation and revision rules show significant improvement in simulated blood glucose control and highlight the potential of CBR for bolus advice.

The method can be adopted by insulin pumps, blood glucose monitors, personal computers, and as a web service accessible from any device with internet access. The potential versatility allows for a widely accessible intelligent solution to bolus advice.

The CBR system could be extended further to include features (due to the limitations of the T1DM simulator). Some features omitted in this research which should be explored include exercise, stress and alcohol. An expanded system with clinical trials would provide a real-world insight into the viability of temporal CBR for T1DM self-management.

There is potential to increase the performance of the proposed retrieval process through the inclusion of dimensional matching to reduce the number of cases requiring an aggregate match. Improvements to efficiency will increase the range of devices for which CBR in this domain would be viable. As the majority of efficiency concerns reside in the retrieval process, a cloud service would be an approach to consider. Such an approach would have to be implemented carefully to prevent issues associated with network access, which is an area of concern for quality of experience.

A CBR service in the cloud also opens up the possibility of case sharing between subjects. This would reduce the need for a case-base to be created using a traditional bolus calculator or similar method, potentially allowing CBR to be used without the need to create a case-base. For case sharing to be reliable, a suitable method for identifying the

similar subjects would be required. Additionally, steps would need to be taken to ensure the security and confidentiality of any stored user information and address the issue of patient trust.

Some elements of this research also have the potential for use in other domains. Most notably the use of temporal sequences to improve case retrieval in temporal and sequential domains. Research into the use of temporal sequences in other domains will help provide evidence to validate the method. Examples of such use may be other medical conditions which require self-management, project cost-estimation for software-engineering, and the prediction of future events in a number of domains based on sequential events. There are certainly questions to be asked about how to determine optimal temporal sequence length, possibly through a hypothesis by domain experts which can then be validated through simulated or real-world testing. Additionally, the use of temporal sequences is highly dependent on feature weighting, so future research could explore the use of different algorithms, or explore how the algorithms used in this research perform in other domains.

## References

- [1] J. Borus, L. Laffel, Adherence challenges in the management of type 1 diabetes in adolescents: Prevention and intervention, *Current opinion in pediatrics* 22 (4) (2010) 405.
- [2] K. Barnard, C. Parkin, A. Young, M. Ashraf, Use of an automated bolus calculator reduces fear of hypoglycemia and improves confidence in dosage accuracy in patients with type 1 diabetes mellitus treated with multiple daily insulin injections, *Journal of Diabetes Science and Technology* 6 (1) (2012) 144–149.
- [3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, The Weka data mining software: An update, *ACM SIGKDD Explorations Newsletter* 11 (2009) 10–18.
- [4] J. Kolodner, *Case-based Reasoning*, Morgan Kaufmann, 1993.
- [5] R. Schank, *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press, 1983.
- [6] K. Hammond, CHEF: A model of case-based planning, in: *Proceedings of the 5th National Conference on Artificial Intelligence*, Vol. 1 of AAAI-86, AAAI Press/MIT Press, Philadelphia, PA, USA, 1986, pp. 267–271.
- [7] J. Kolodner, R. Simpson, The MEDIATOR: Analysis of an early case-based problem solver, *Cognitive Science* 13 (1989) 507–549.
- [8] P. Koton, Reasoning about evidence in casual explanations, in: *Seventh National Conference on Artificial Intelligence*, AAAI-88, AAAI Press, St. Paul, MN, USA, 1988, pp. 256–261.
- [9] A. Aamodt, E. Plaza, Case-based reasoning: Foundational issues, methodological variations, and system approaches, *AI Communications* 7 (1994) 39–59.
- [10] M. Jaczynski, A framework for the management of past experiences with time-extended situations, in: *Proceedings of the sixth international conference on Information and knowledge management*, ACM, Las Vegas, NV, USA, 1997, pp. 32–39.
- [11] M. Jære, A. Aamodt, P. Skalle, Representing temporal knowledge for case-based prediction, in: *Advances in case-based reasoning*, Springer, 2002, pp. 174–188.
- [12] M. Sánchez-Marré, U. Cortés, M. Martínez, J. Comas, I. Rodríguez-Roda, An approach for temporal case-based reasoning: Episode-based reasoning, in: *Case-Based Reasoning Research and Development*, Springer, 2005, pp. 465–476.
- [13] D. Brown, I. Bayley, R. Harrison, C. Martin, Developing a mobile case-based reasoning application to assist type 1 diabetes management, in: *2013 IEEE 15th International Conference on e-Health Networking, Applications & Services (Healthcom)*, IEEE, Lisbon, Portugal, 2013, pp. 1–3.
- [14] D. Brown, *Temporal case-based reasoning for insulin decision support*, Ph.D. thesis, Department of Computing and Communication Technologies, Oxford Brookes University (2015).
- [15] C. Martin, D. Flood, D. Sutton, A. Aldea, R. Harrison, M. Waite, A systematic evaluation of mobile applications for diabetes management, in: *INTERACT 2011, 13th IFIP TC 13 International Conference, Proceedings, Part IV, Human-Computer Interaction*, Springer, Lisbon, Portugal, 2011, pp. 466–469.
- [16] B. Kovatchev, M. Breton, C. Dalla Man, C. Cobelli, In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes, *Journal of Diabetes Science and Technology* 3 (2009) 44–55.
- [17] J. Tukey, *Exploratory data analysis*, Pearson, 1977.
- [18] I. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition (Morgan Kaufmann Series in Data Management Systems), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [19] R. Campbell, A. Abramovich, Calculating insulin on board for extended bolus being delivered by an insulin delivery device, last accessed: 2017-01-24 (2012). URL <https://www.google.com/patents/US8140275>
- [20] Becton, Dickinson and Company, *Staying on Target™: Your Insulin Adjustment Workbook. Yes, You Can Do It!*, Becton, Dickinson and Company, 2005.
- [21] P. Davidson, H. Hebblewhite, B. Bode, R. Steed, N. Welch, M. Greenlee, P. Richardson, J. Johnson, Statistically based CSII parameters: correction factor, CF (1700 rule), carbohydrate-to-insulin ratio, CIR (2.8 rule), and basal-to-total ratio, *Diabetes Technology and Therapeutics* 5 (5) (2003) A237.
- [22] University College London Hospitals, *Correcting a high blood glucose level*, NHS, 2013.
- [23] P. C. Davidson, H. R. Hebblewhite, R. D. Steed, B. W. Bode, Analysis of guidelines for basal-bolus insulin dosing: basal insulin, correction factor, and carbohydrate-to-insulin ratio, *Endocrine Practice* 14 (9) (2008) 1095–1101.
- [24] A. B. King, D. U. Armstrong, A prospective evaluation of insulin dosing recommendations in patients with type 1 diabetes at near normal glucose control: Bolus dosing,

- Journal of Diabetes Science and Technology 1 (1) (2007) 42–46.
- [25] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *Journal of Machine Learning Research* 3 (2003) 1157–1182.
- [26] M. A. Hall, Correlation-based feature selection for machine learning, Ph.D. thesis, Department of Computer Science, The University of Waikato, Hamilton, Waikato, New Zealand (1999).
- [27] H. Liu, R. Setiono, Chi2: Feature selection and discretization of numeric attributes, in: *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*, 1995, IEEE, Herndon, VA, USA, 1995, pp. 388–391.
- [28] R. Kerber, ChiMerge: Discretization of numeric attributes, in: *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI-92, AAAI Press, 1992, pp. 123–128.
- [29] U. M. Fayyad, K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: *Proceedings of the the Thirteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Chambéry, France, 1993, pp. 1022–1027.
- [30] J. Rissanen, Modeling by shortest data description, *Automatica* 14 (5) (1978) 465–471.
- [31] H. Liu, F. Hussain, C. L. Tan, M. Dash, Discretization: An enabling technique, *Data mining and knowledge discovery* 6 (4) (2002) 393–423.
- [32] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: *Proceedings of the Twentieth International Conference on Machine Learning*, Vol. 3, Washington, DC, USA, 2003, pp. 856–863.
- [33] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.
- [34] M. A. Hall, L. A. Smith, Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, in: *FLAIRS conference*, Vol. 1999, Orlando, FL, USA, 1999, pp. 235–239.
- [35] J. Mingers, An empirical comparison of selection measures for decision-tree induction, *Machine Learning* 3 (1989) 319–342.
- [36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.
- [37] R. C. Holte, Very simple classification rules perform well on most commonly used dataset, *Machine Learning* 11 (1993) 63–91.
- [38] C. G. Nevill-Manning, G. Holmes, I. H. Witten, The development of Holte’s 1R classifier, in: *Proceedings of the Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, IEEE, Dunedin, New Zealand, 1995, pp. 239–242.
- [39] I. Kononenko, Estimating attributes: Analysis and extensions of relief, in: *Proceedings of the Seventh European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence, Springer-Verlag Berlin Heidelberg, Catania, Italy, 1994, pp. 171–182.
- [40] K. Kira, L. Rendell, A practical approach to feature selection, *Proceedings of the Ninth International Conference on Machine Learning* (1992) 249–256.
- [41] T. Farmer, T. Edgar, N. Peppas, The future of open and closed-loop insulin delivery for diabetes mellitus, *The Journal of Pharmacy and Pharmacology* 60 (2008) 1–13.
- [42] Roche, Accu-Chek® Aviva Expert Blood Glucose Meter - Standard Owner’s Booklet (2012).
- [43] W. Clarke, B. Kovatchev, Statistical tools to analyse continuous glucose monitor data, *Diabetes Technology and Therapeutics* 11 (2009) S-45–S-55.
- [44] B. Kovatchev, D. Cox, L. A. Gonder-Frederick, D. Young-Hyman, D. Schlundt, W. Clarke, Assessment of risk for severe hypoglycemia among adults with IDDM: validation of the low blood glucose index., *Diabetes care* 21 (11) (1998) 1870–1875.
- [45] B. Kovatchev, E. Otto, D. Cox, L. Gonder-Frederick, W. Clarke, Evaluation of a new measure of blood glucose variability in diabetes, *Diabetes Care* 29 (11) (2006) 2433–2438.
- [46] Roche, Blood glucose index, last accessed: 2017-01-24 (2015).  
URL <https://hcp.accu-chek.co.uk/gbconnect/blood-glucose-index.html>
- [47] D. Rodbard, Interpretation of continuous glucose monitoring data: Glycemic variability and quality of glycemic control, *Diabetes Technology and Therapeutics* 11 (2009) S-55–S-67.
- [48] NICE, Type 1 diabetes: Diagnosis and management of type 1 diabetes in children, young people and adults, NICE, 2004.
- [49] R. Bellazzi, C. Larizza, S. Montani, A. Riva, M. Stefanelli, G. d’Annunzio, R. Lorini, E. Gómez, et al., A telemedicine support for diabetes management: the T-IDDM project, *Computer methods and programs in biomedicine* 69 (2) (2002) 147–161.
- [50] R. Marling, J. Shubrook, F. Schwartz, Case-based decision support for patients with type 1 diabetes on insulin pump therapy, *Advances in Case-Based Reasoning, Lecture Notes in Computer Science* 5239 (2008) 325–339.
- [51] P. Herrero, P. Pesi, M. Reddy, N. Oliver, P. Georgiou, C. Toumazou, Advanced insulin bolus advisor based on run-to-run control and case-based reasoning, *IEEE Journal of Biomedical and Health Informatics* 19 (2015) 1087–1096.
- [52] P. Pesi, P. Herrero, M. Reddy, N. Oliver, D. Johnston, C. Toumazou, P. Georgiou, Case-based reasoning for insulin bolus advice: Evaluation of case parameters in a six-week pilot study., *Journal of Diabetes Science and Technology* doi:10.1177/1932296816629986.
- [53] S. Wess, K.-d. Althoff, G. Derwand, Using k-d trees to improve the retrieval step in case-based reasoning, in: *EWCBR ’93 Selected papers from the First European Workshop on Topics in Case-Based Reasoning*, Springer-Verlag, Kaiserslautern, Germany, 1993, pp. 167–181.
- [54] J. Allen, Maintaining knowledge about temporal intervals, *Communications of the ACM* 26 (11) (1983) 832–843.
- [55] L. Branting, J. Hastings, An empirical evaluation of model-based case matching and adaptation, in: *AAAI-94 Workshop: Case-Based Reasoning*, Seattle, WA, USA, 1994, pp. 72–78.