Sudan University of Science and Technology

Collage of Graduate Studies

A Thesis Submitted in Partial Fulfillment of the Requirements of
M.Sc. in Computer Science

SELF-DIAGNOSIS OF DIABETES USING CASE-BASED REASONING

التشخيص الذاتي لمرض السكري

بالإستنتاج من قاعدة بيانات الحالات

OCTOBER 2017

Sudan University of Science and Technology

Collage of Graduate Studies

A Thesis Submitted in Partial Fulfillment of the Requirements of
M.Sc. in Computer Science

SELF-DIAGNOSIS OF DIABETES USING CASE-BASED REASONING

التشخيص الذاتي لمرض السكري
بالإستنتاج من قاعدة بيانات الحالات

Prepared By:

Halima Mustafa Abdallah Ahmed

Supervisor:

Dr. Nadir Kamal Salih

الآية

قَالَ تَعَالَىٰ:

﴿ قَالُوا۟ سُبْحَـٰنَكَ لَا عِلْمَ لَنَآ إِلَّا مَا عَلَّمْتَنَآ ۖ إِنَّكَ أَنتَ ٱلْعَلِيمُ ٱلْحَكِيمُ ﴿٣٢﴾ ﴾

صدق الله العظيم

سورة البقرة (32)

الحمد لله

الحــمد لله رب العالمين ،أعطى اللسان،وعَلَّم البيان ،وخلق الإنسان ،فبإي ألآء ربكما تكذبان .. لك الحمد يا من هو للحِمد أهل ،أهل الثناء والمجد ،أحقُ ماقال العبد وكلنا لك عبد ،لك الحمد مادعوناك إلا حسنُ ظنٍ بك ومارجوناك إلا ثقة فيك ، وماخفناك إلا تصديقاً بوعدك ووعيدك لك الحمد حمداً كثيراً طيباً مباركاً فيه ، وصلى الله على سيدنا محمد خاتم الأنبياء والمرسلين أجمعين بشر وأنذر ووعد وأوعد ،أنقذ الله به البشر من الضلالة وهدى الناس الى صراط المستقيم ،صراط الله الذي له مافي السموات ومافي الأرض الا الى الله تصير.

DEDICATION

Special dedication to my family members especially to my beloved father and mother (Musta fa Abdallah and Fatima Hassabelrasoul) who always give me encouragement in my life, my study and to finish my Project .

To my Supervisor

Dr. Nadir Kamal Salih

To all SUST's lecturers

To all my classmate

To Dr. Sharaf Shuaib, Dr.Ibrahim, Dr. Roaa in Nile College

And all my friends out here

Thank You for your supporting and teaching

Thank You for everything that gave during my studies and the knowledge that we shared toge ther.

THANK YOU SO MUCH

ACKNOWLEDGEMENT

Thanks first and foremost to God Almighty who honored us in accomplishing this humble work, and I would like to express my deepest appreciation to all those who provided me the information to complete this research.

A special gratitude I give to my research supervisor, who contribution in stimulating suggestions and encouragement helped me to coordinate my project especially in writing this report.

Dr. Nadir Kamal Salih

Furthermore I would also like to acknowledge with much appreciation the crucial role of the.

Sudan University of Science and Technology Collage of Graduate Studies

Last but not the least, I would like to thank my family:, for help me and supporting me spiritually throughout my life.

Myparents, brothers and my sisters

ABSTRACT

The continuously rising cost of medical spending, population size is growing up and increasing their need for healthcare, which requires time saving for laboratory technicians at the examination is a great incentive to create a new approach that helps to provide health care to patients at lower costs with good management. We focus on design an autonomic environment to help management of healthcare service.

We started to apply the concept of the autonomic system that let the system work without intervention of the user. It has given by implemented and designed Case-Based Reasoning (CBR) algorithm in suitable way. We have proposed method can diagnose new patients according to the similar solution of stored cases. The system appears in attractive vision because the doctor just enters some parameters for new patient through the form of the system and the prediction result coming soon. The experimental results suggest that such a system is valuable both for less experienced clinicians and for experts where the system may function as a second option. The success of this work will permit to leverage the development of CBR systems in medicine. It will become possible to develop a web service to federate the CBR process across several domains of medicine.

Finally our application can be extended by developing a special method to diagnose diabetes mellitus for gestational women and it can be implemented as a mobile application due to the development of technology nowadays.

المستخلص

إن الإرتفاع المستمر في تكلفة النفقات الطبية، وإزدياد حجم السكان وزيادة حاجتهم إلى الرعاية الصحية، مما يتطلب توفير الوقت لفنيي المعامل هو حافز كبير لإنشاء نهج جديد يساعد على توفير الرعاية الصحية للمرضى بتكاليف أقل مع إدارة جيدة. إننا نركز على تصميم بيئة مستقلة للمساعدة في إدارة خدمة الرعاية الصحية.

لقد بدأنا بتطبيق مفهوم نظام الاستقلال الذاتي فيه يقوم النظام بالعمل دون تدخل المستخدم. وقد تم تصميمه بتطبيق خوارزمية الإستنتاج من قاعدة الحالات (Case-Based Reasoning) بطريقة مناسبة. وقد اقترحنا طريقة تمكن من التشخيص للمرضى الجدد وفقا للحلول المماثلة للحالات المخزنة. يظهر النظام في بصورة جذابة لأن الطبيب يدخل فقط بعض المعاملات للمريض الجديد من خلال استمارة النظام والتنبؤ بالنتيجة بسرعة. النتائج التجريبية تشير إلى أن مثل هذا نظام يعتبر مفيدا" للأطباء قليلي الخبرة والخبراء على حد سواء حيث قد يعمل النظام كخيار ثان للخبراء. و نجاح هذا العمل يسمح بتطوير انظمة للمجال الطبي بإستخدام خوارزمية الإستنتاج من قاعدة البيانات.

وأخيراً يمكن تمديد التطبيق بتطوير أسلوب خاص لتشخيص مرض السكري النساء الحوامل ويمكن تنفيذها كتطبيق الجوال نتيجة لتطور التكنولوجيا في الوقت الحاضر.

Table of Contents

List of Tables

List of Figures

Abbreviation table

Symbol	**Meaning**
DMDS	Diabetes Mellitus Diagnosis System
CBR	Case-Based Reasoning
GUI	Graphical User Interface
Mellitus	High blood glucose

CHAPTER I

INTRODUCTION

1.1 Introduction

This section briefly describes the diagnosis of diabetes mellitus using case-based reasoning (CBR) for predicting diabetic, pre-diabetic and normal patients that have been developed. This chapter consists of five sections: the first section describes the background of the project. The second section describes the problem statement and motivation of the project. The third section describes the objectives for the project. The fourth section describes the scope for the project. Finally in section five thesis organizations is described.

1.2 Background

In the medical domain, diagnostic, classification and treatment are the main tasks for a physician. The multi-faced and complex nature of the medical domain such as the psych- physiological domain often requires the development of a system applying several artificial Intelligence techniques for instance CBR [1].

Diabetes is a lifelong (chronic) disease increase at a rapid rate because of sedate life style, changes into urban culture, unhealthy foods and lacking of physical activity [2], it is a group of diseases characterized by high levels of blood glucose ("sugar") resulting from defects in insulin secretion, insulin action or both.

Insulin is a hormone that regulates carbohydrate metabolism by controlling blood glucose levels. The global prevalence of diabetes among adults over 18 years of age has risen from 4.7% in 1980 to 8.5% in 2014. In 2012, an estimated 1.5 million deaths were directly caused by diabetes and another 2.2 million deaths were attributable to high blood glucose (WHO report).

Case-based reasoning (CBR) is inspired by the way humans reasoning e.g. solve a new problem by applying previous experiences adapted to the current situation. An experience (a case) normally contains a, a diagnosis/classification, a solution and its results. For a new problem case, a CBR system matches the problem part of the case against cases in the so called case library and retrieves the solutions of the most similar cases that are suggested as solution after adapting it to the current situation [1].

We are going to develop a computer based diagnosis system for Diabetes disease by Compiling information and symptoms of diabetes disease .Diabetes mellitus is a disease in which the body's ability to produce or respond to the hormone insulin is impaired, resulting in abnormal metabolism of carbohydrates and elevated levels of glucose in the blood and urine.

From World Health Organization (WHO) in 2012 an estimated 1.5 million deaths were directly caused by diabetes and another 2.2 million deaths were attributable to high blood glucose.

CBR is an approach for solving problems based on solution of similar past cases. The purpose of CBR is to provide the decision maker with the ability to utilize the specific knowledge of previously experienced, concrete problem situation or specific patients" cases.. It is not used only in medical domain but also in the financial, agricultural, management and many more domains (Investopedia.com, 2011).

There are some different papers present the technique for diagnosis of diabetes mellitus such as Neutral Network, Expert Application, and Rough sets. By using the CBR technique, it can improve the solving problem performance through reuse and makes use of existing data. It also can reduce the knowledge acquisition efforts and require less maintenance effort.

1.3 Problem Statement

The number of specialists and expertise's in the medical domain about the diabetes mellitus is limited. The patients have to make appointments with them before doing the medical check-up. So many patients have to wait too long to get their result from the checkup.

CBR will help the doctors to make their works easy and provide the quick and correct medical reports to their patients

Research question 1: which method can be developed to self-diagnosis of diabetes mellitus?

We started to apply the concept of the autonomic system that let the system work without intervention of the user. It has given by implemented and designed Case-Based Reasoning (CBR) algorithm in suitable way.

Research question 2: How can we use CBR to develop a system that diagnoses diabetes mellitus?

We can use the test diagnosis of patients as cases of CBR, and then we can diagnose new patients according to the similar solution of stored cases.

1.4 Objectives

Medicine is a large and a complex domain, which makes it a difficult area to perform reasoning within. Instead of tackling the domain as a whole, we intend to propose a system, which is adaptive and can reason within a specific area of medicine, we are going to build a semantically intelligent CBR that mimic the expert thinking can diagnose diabetes disease.

The product of our work will be a software system that implements Case-Based Reasoning algorithm to test it on real life data. Our main goal is to create a system that through interaction with the patient is able to accurately predict the diagnosis of that patient which yields good performance within diagnosis. This can help to:

1) Diagnose the diabetes disease earlier
2) Discover if the person is prone to the diabetes disease
3) Reduce diabetes disease diagnosis cost and time
4) Increase the availability and the number of resources and activities for people with diabetes, their families and other interested parties

1.5 Scope

This autonomic system uses the CBR algorithm to solve the problem and uses dataset which is collected from military hospital. This system is based on stand-alone system and focus on checking if the patient is diabetic or he/she is prone to. The user for the system is expert doctors and medical staffs in order to help them diagnose the diabetes mellitus.

1.6 Thesis organization

This is about CBR for diagnosing diabetes mellitus and consists of five chapters as follows:

Chapter One: Introduction

In this chapter, we provide background information about the application which normally includes problem statement, objectives and scope.

Chapter Two: Literature Review

Consists of case study, literature review and evaluation of previous research of CBR technique in diagnosing diabetes mellitus.

Chapter Three: Methodology

Is a guideline for solving a problem, with specific components of CBR such as phases, tasks, methods, techniques and tools using in the project

Chapter Four: Design, Implementation, Analysis and Results

The implementation of the research project is presented, interfaces design, present the result of the project and discuss the outcome of the project

Chapter Five: Conclusion and Recommendation

The researcher makes the conclusion and suggests some recommendation in order improve the project in future. This chapter will briefly summarize the overall project

References

Chapter II

Literature Review

2.1 Introduction

This chapter describes the review on diagnosis of diabetes mellitus using expert systems. In this chapter, two sections are comprised: The first section briefly explain the background of diabetes mellitus and diabetes mellitus dataset. The second section describes the review on previous works.

2.2 Diabetes Mellitus

Diabetes mellitus is one of the common diseases in the world. It is a disease in which the body’s ability to produce or respond to the hormone insulin is impaired, resulting in abnormal metabolism of carbohydrates and elevated levels of glucose in the blood and urine. Many of the complications associated with diabetes, such as nephropathy, retinopathy (which leads to blindness), neuropathy, cardiovascular disease, stroke, and death, can be delayed or prevented with appropriate treatment of elevated blood pressure, lipids, and blood glucose[25,4,5].

The global prevalence of diabetes* among adults over 18 years of age has risen from 4.7% in 1980 to 8.5% in 2014. In 2012, an estimated 1.5 million deaths were directly caused by diabetes and another 2.2 million deaths were attributable to high blood glucose [25].

The following figure(2.1) shows the percentage Of All DEATHS attributable TO HIGH blood glucose for adults AGED 20–69 years, By who region ND sex, for years 2000 AND 2012:

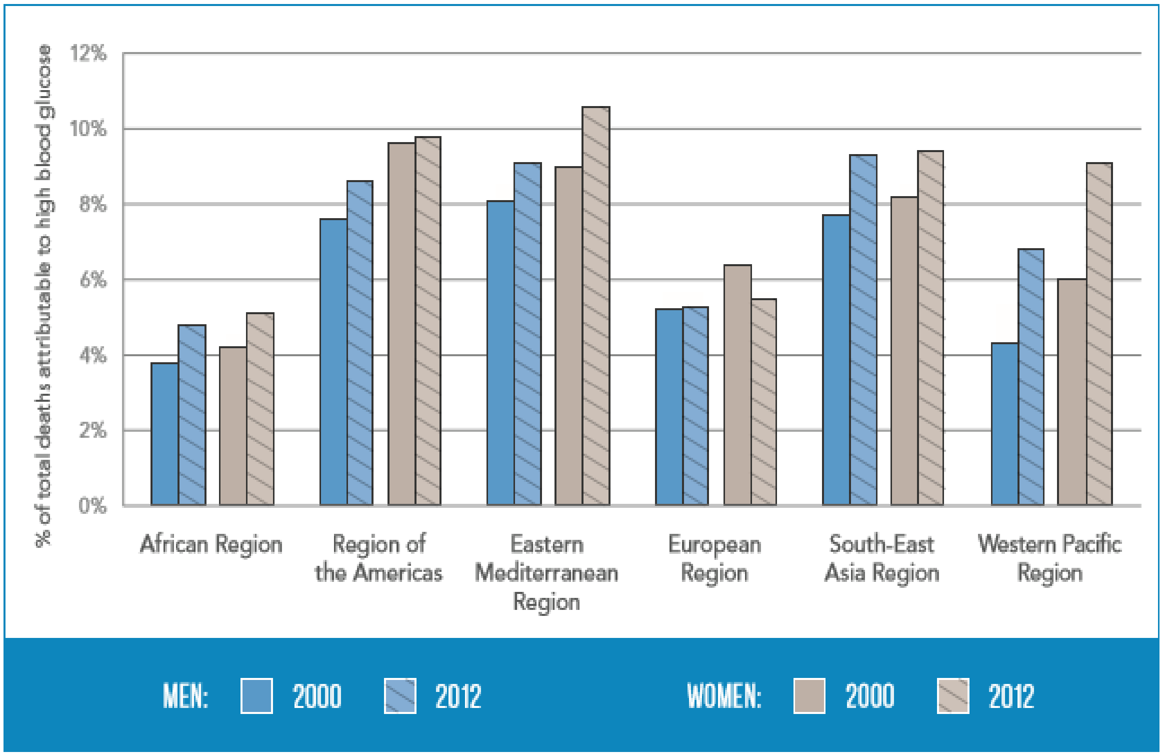


Figure 2.1 percentages of all deaths attributable to high blood glucose [25]

2.3 Diabetes types

There are three types of diabetes:

2.3.1 Type 1 diabetes

The exact cause of type 1 diabetes is unknown. What is known is that your immune system — which normally fights harmful bacteria or viruses — attacks and destroys your insulin-producing cells in the pancreas. This leaves you with little or no insulin. Instead of being transported into your cells, sugar builds up in your bloodstream.

Type 1 is thought to be caused by a combination of genetic susceptibility and environmental factors, though exactly what many of those factors are is still unclear[5].

2.3.2 Type 2 diabetes

In pre-diabetes — which can lead to type 2 diabetes — and in type 2 diabetes, your cells become resistant to the action of insulin, and your pancreas is unable to make enough insulin to overcome this resistance. Instead of moving into your cells where it's needed for energy, sugar builds up in your bloodstream.

Exactly why this happens is uncertain, although it's believed that genetic and environmental factors play a role in the development of type 2 diabetes. Being overweight is strongly linked to the development of type 2 diabetes, but not everyone with type 2 is overweight.[5]

2.3.3 Gestational diabetes

During pregnancy, the placenta produces hormones to sustain your pregnancy. These hormones make your cells more resistant to insulin.

Normally, your pancreas responds by producing enough extra insulin to overcome this resistance. But sometimes your pancreas can't keep up. When this happens, too little glucose gets into your cells and too much stays in your blood, resulting in gestational diabetes[6].

2.3.4 Diabetes Symptoms

The most consistent symptom of diabetes mellitus (Type 1 & Type 2) is elevated blood sugar levels. Type 1 is caused by the body not producing enough insulin to properly regulate blood sugar, while in Type 2 diabetes, is caused by the body developing resistance to insulin. Ignoring the diabetes symptom at early stage can lead to long-term serious health risk and complications that may lead to other fatal diseases. Below shows some common “early sign “of diabetes:

Type 1 Diabetes

a) Frequent urination

b) Unusual thirst

c) Extreme hunger

d) Unusual weight loss

e) Extreme fatigue and Irritability

Type 2 Diabetes

a) Any of the type 1 symptoms

b) Slow healing of wounds

c) Blurred vision

d) Cuts/bruises that are slow to heal

e) Tingling/numbness in the hands/feet

f) Dry or Itchy skin, gum, or bladder infections

2.4 Diabetes mellitus diagnosis tests:

There are several tests are used to diagnosis the diabetes such as[7]:

2.4.1 Fasting Plasma Glucose Test (FPG):

Measures the blood glucose in a person who has not eaten anything for at least 8 hours in order to detect diabetes or pre-diabetes

2.4.2 Oral Glucose Tolerance Test (OGTT):

In order to measure the blood glucose level after a person fasts at least 8 hours and 2 hour after the person drinks a glucose containing beverage and a random plasma glucose test to measures blood glucose level without regard to when the person being tested last ate.

2.4.3 HbA1c Test:

Also called the hemoglobin A1C, HbA1c, or glycohemoglobin test.

2.4.4 Random Plasma Glucose (RPG) Test:

Sometimes used to diagnose diabetes during a regular health check up. If the RPG measures 200 micrograms per deciliter or above.

Below figure (2.2) that shows the criteria of diabetes diagnosis using FPG, OGTT and HbA1c:

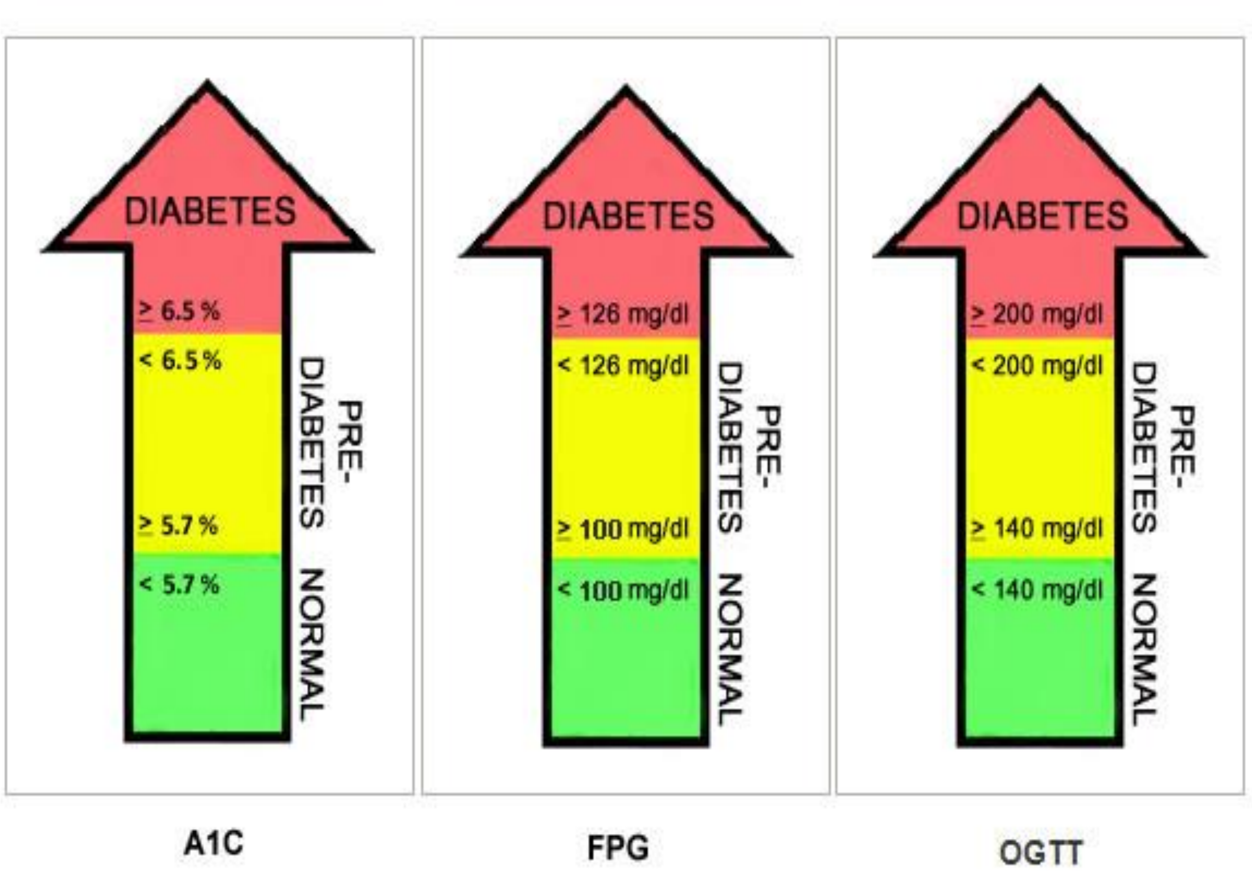


Figure 2.2 Criteria of the Diagnosis of Diabetes[25]

2.5 Previous works

2.5.1 Artificial Neural Networks

R. P. Ambilwade et al in [8] discussed that medical expert systems being used for diabetes diagnosis where the patient's symptoms and other details are inputs and the system diagnose the disease, recommend treatment or drugs which may be prescribed. Artificial neural network used. In this research work, the highest accuracy above 89% is ANN.

Ebenezer Obaloluwa et al in [9] diagnosed diabetes by creating a multilayer feed-forward and trained with back-propagation algorithm which classify patient that are tested positive as binary 1 and patient that are tested negative as binary 0.The use of trained neural network gave recognition rate of 82% on test.

The advantage using neural network is a neural network learns application behavior by using application input-output data. Neural networks have good capabilities. It is effective in time variant problems, even under noisy conditions. Thus, neural nets-can solve many problems that are either unsolved or inefficiently solved by existing techniques, including fuzzy logic.

Finally, neural networks can develop solutions to meet a pre-specified accuracy. It is difficult, if not impossible, to determine the proper size and structure of a neural networks to solve a given problem. Neural network also do not scale well.

Manipulating learning parameters for learning and convergence becomes increasingly difficult. Artificial neural network are still far away from biological neural networks, but what we know today about artificial neural networks is sufficient to solve many problems that were previously unsolvable or inefficiently solvable at best.

2.5.2 Rule Based Systems

Akteretal in [10] have provided a knowledge-based system for diagnosis and management of diabetes mellitus. They believed that preventive care helps in controlling the severity of chronic disease of diabetes. In addition, preventive measures require proper educational awareness and routine health checks. The main purpose of this research was developing a low-cost automated knowledge-based system with easy computer interface. This system performs the diagnostic tasks using rules achieved from medical doctors on the basis of patients' data.

Tawfik Saeed et al in [11] have provided a rule-based expert system to diagnose all types of diabetes, coded with VP_Expert Shell and tested in Shahid Hasheminezhad Teaching Hospital affiliated to Tehran University of Medical Sciences and final expert system has been presented. The mentioned that some of these patients do not access to the physicians during necessary times. Therefore, such a system can provide necessary information about the indications, diagnosis and primary treatment advices to the diabetics. Since this expert system gathers its knowledge from several medical specialists, the system has a broader scope and can be more helpful to the patients -- in comparison to just one physician.

Rules are the popular paradigm for representing knowledge. A rule based expert system is one whose knowledge base contains the domain knowledge coded in the form of rules. It is efficient because of modular nature which means encapsulating knowledge and expansion of the expert system done in an easy way.

Also rules make it easy to build explanation facilities. But Rule-based systems need to be specially crafted to avoid infinite loops; the modification of Knowledge Base can be complicated possibility of contradictions. Following table (2.1) shows the analysis of previous works:

Table 2.1 previous works

Study	Technique	Results	Open issue
[8]	Artificial neural networks	In this research work, the highest accuracy above 89%	It is difficult to determine the proper size and structure of a neural networks to solve a given problem
[9]	Artificial neural network	The use of trained neural network gave recognition rate of 82% on test	It is difficult to determine the proper size and structure of a neural networks to solve a given problem
[10]	Rule-based system	This system performs the diagnostic tasks using rules achieved from medical doctors on the basis of patients’ data	Needs to be crafted to avoid infinite loops
[11]	Rule-based system	the system has a broader scope and can be more helpful to the patients -- in comparison to just one physician	Needs to be crafted to avoid infinite loops

We have discussed that It is difficult, if not impossible, to determine the proper size and structure of a neural networks to solve a given problem. Neural network also do not scale well, and Rule-based systems need to be specially crafted to avoid infinite loops, the modification of Knowledge Base can be complicated, possibility of contradictions.

So we are going to develop a new method to avoid these problems.

2.6 Summary

The number of expertise in the medical domain about diabetes mellitus is limited. Many patients have to wait too long for getting the check up result. The experience medical staffs are decreasing in number due to retire, the new staffs will replacing their places. They have to learn more about their works. This application is very useful in the management application and aids the inexperienced physicians to check their diagnosis. CBR seems to be a suitable technique for medical knowledge based application. This technique will be more effective at applying the existing cases to new situation. It will be as the doctor diagnostic assistant as the aim of the research is to classify blood glucose of patient to normal, prediabetic or diabetic. In order to achieve the aim of the research, the objectives such as developing an application to diagnose diabetes mellitus applying the CBR algorithm must be met. The methodology of the application will be discussed in the next chapter.

CHAPTER III

METHODOLOGY

3.1 Introduction

This chapter discusses the methodology of our system, diagnosis of diabetes mellitus using case-based reasoning. This application is called as DIABETES MELLITUS DIAGNOSIS APPLICATION (DMDA).

3.2 Case-Based Reasoning (CBR)

The most important research agendas of AI are scientific and technological. Scientific agenda is to understand the nature of intelligence and human thought. And in technological agenda AI researchers seek to develop the technology of intelligence which leads to create machines that can perform useful tasks and intelligent artifacts.So they will be able to design and build computer programs that can solve problems and adapt to new situations. In this article, we discuss case-based reasoning, an AI paradigm that addresses both research agendas.

Case-based reasoning is a psychological theory of human cognition[15].Case-based reasoning (CBR) was first formalized in the 1980s following from the work of Schank and others on memory [12], and is based upon the fundamental premise that similar problems are best solved with similar solutions [13]. Its idea is to learn from experience.

3.2.1 An Overview of CBR

CBR enables utilization of knowledge of previously experienced, concrete problem situations. A CBR system requires a good supply of cases in its case database. The retrieval task starts with a problem description, and ends when a best matching previous case has been found. A new problem is solved by finding a similar past case, and reusing it in the new problem situation. Sometimes a modification of the solution is done to adapt the previous solution to the unsolved case. It is important to emphasize that CBR also is an approach to incremental and sustained learning; learning is the last step in a CBR cycle [14, 16].

3.2.2 Case-based problem solving

When patterns are complex, CBR could be used as a pattern recognition technique without the need for defining explicit patterns [18].

The basic reasoning cycle of a CBR agent can be summarized by a schematic cycle (see Fig. 1) and detailed in the following steps [16]:

a) Retrieve the most similar case(s) to the new case. Similarity measures are involved in this step.

b) Adapt or reuse the information and knowledge in that case to solve the new case. The selected best case has to be adapted when it does not match perfectly the new case.

c) Evaluate or revise the proposed solution. A CBR agent usually requires some feedback to know what is going right and what is going wrong. Usually, it is performed by simulation or by asking a human. Learn or retain the parts of this experience likely to be useful for future problem solving. The agent can learn both from successful solutions and from failed ones (repair).

3.2.3 CBR Components

There are several components of CBR:

i. Case

The case has two components: the problem description and the solution. So it is defined as an instance of a problem [18].

ii. Case Base

The case base contains the experiences and conforms to one of the four sources of knowledge required in a CBR. They are the vocabulary, the case-base, the similarity measure and adaptation containers.

1. The first, the vocabulary, contains the terms which support the others. The case-base comprehends what is in a case and how cases are organized.

2. The similarity measure container contains knowledge to determine the similarity between two cases in the retrieval phase.

3. The solution adaptation container contains knowledge to adapt past solutions to new problems in the reuse stage [18], [24].

iii. Case index:

Kolodner identifies indexing with an accessibility problem [22], that is, with the whole set of issues inherent in setting up the case base and its retrieval process so that the right cases are retrieved at the right time. Thus, case indexing involves assigning indices to cases to facilitate their retrieval. CBR researches proposed several guidelines on indexing [23]. Indexes should be:

1. Predictive of the case relevance
2. Recognizable in the sense that it should be understandable why they are used
3. Abstract enough to allow for widening the future use of the case base concrete (discriminative) enough to facilitate efficient and accurate retrieval.

3.2.4 Representation of Cases

The retrieval of previous cases which leads to solve the target problem is considered as an important step in CBR cycle [17].

The Retrieve task starts with a (partial) problem description, and ends when a best matching previous case has been found. Its subtasks are referred to as Identify Features, Initially Match, Search, and Select, executed in that order. The identification task basically comes up with a set of relevant problem descriptors, the goal of the matching task is to return a set of cases that are sufficiently similar to the new case - given a similarity threshold of some kind, and the selection task works on this set of cases and chooses the best match (or at least a first case to try out) [14]. In addition, source of the cases needs to be decided.

In the medical domain, they can for instance be created with help from expert physicians or be the result of data mining from existing electronic medical records. Maintenance of these libraries also becomes important when the number of cases grows large.

Although these are general problem areas that are relevant for all types of CBR systems, a special consideration needs to be made when working with CBR systems.

3.2.5 CBR phases

Cycle Aamodt and Plaza [14] identified four stages of CBR — sometimes referred to as the R4 model — that combine to make a cyclical process:

i. Retrieve similar cases to the target problem
ii. Reuse past solutions
iii. Revise or adapt the suggested solutions to better fit the target problem
iv. Retain the target and solution in the case-base

Below figure (3.1) shows these phases of CBR:

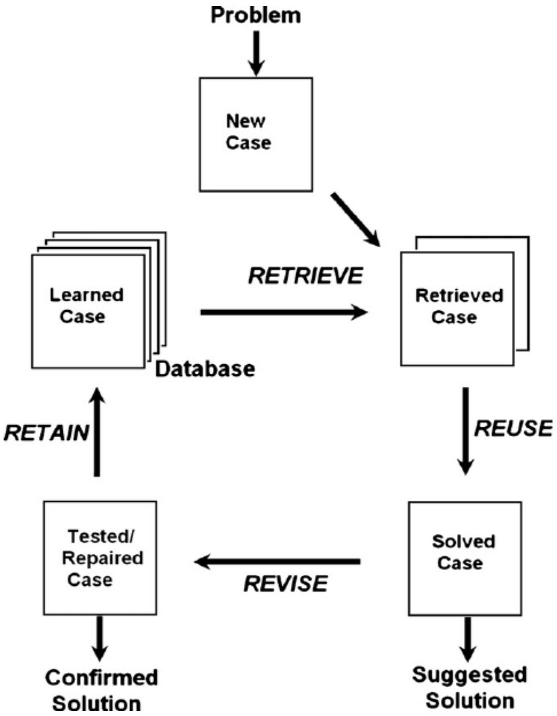


Figure 3.1 The CBR Process [14]

i. Retrieval phase:

This is an important stage where a measure of how a present case is similar to past cases is done. The most similar case to the present case is then retrieved from the case base using some similarity metrics.

The similarity measure computes the similarity between a new case and previous cases restored in the case base. Depending on the application domain and features used for describing cases, a simple or more complex measure can be applied [19][20].

Possibly the most widely used technique used for retrieval in CBR are nearest neighbor techniques [3].

Nearest neighbor algorithms all use a similar technique. The similarity between a target case, q, and a case in the case-base, x, are determined by calculating the similarity δ between each feature, f, in both cases. This similarity may then be scaled using a weighting factor, wf. A sum of all scaled similarities is calculated to provide a measure of similarity between the two cases (the target case and the case in the case-base).

Nearest match can be represented by the following equation:

$$\mathrm{Sim}(T,S) = \sum_{i=1}^{n} f(T_j\,, S_i\,) * W_i \ldots\ (3.1)$$

Where:

T= target case

S= source case

n= number of attributes in each case

I= individual attribute from 1 to n

f= similarity function for attributes I in cases T and S

w= importance weighting of attribute I

Nearest-neighbor is not efficient technique. Because, when new case is introduced, indexing should be performed and it could affect efficiently.

Similarities are usually normalized to fall within the range [0, 1], where zero is dissimilar and one being an exact match. Most CBR tools that use nearest neighbor techniques use algorithms similar to this, for example the Wayland System (Price and Pegler, 1995).

ii. Reuse phase

This stage allows reusing and adapting the suggested solution (retrieved most similar case) to the target problem. Which means proposing a solution for a new problem from the solutions of the retrieved cases.

In the “4 REs” of Aamodt& Plaza’s (1994) classic CBR cycle (Figure 1), reuse appears second, after retrieve, and is followed by revise and retain. Reusing a retrieved case can be as easy as returning the retrieved solution, unchanged, as the proposed solution for the new problem. This is often appropriate for classification tasks, where each solution (or class) is likely to be represented frequently in the case base, and therefore the most similar retrieved case, if sufficiently similar, is likely to contain an appropriate solution. But reuse becomes more difficult if there are significant differences between the new problem and the retrieved case’s problem. In these circumstances the retrieved solution may need to be adapted to account for these important differences. Medical decision making is one domain in which adaptation is commonly required.

iii. Revise phase

Typically, the revision phase consists of evaluating the case solution generated by the reuse phase and learning about it. If the result is successful, then the system learns from the success (case retainment), otherwise it is necessary to repair the case solution using domain-specific knowledge. With regard to our approach, revision must be done from experts and specialists of diabetes mellitus.

In CBR systems the solution is successful or wrong. If it is successful, case can be retained, inserting it into the case base if necessary, or it should not be. But when the solution fails, the system is also interested in retaining the reason for the failure thus; there is an investigation task to find out additional information about the case.

iv. Retain phase

This stage retains the solution and adds it to the case base once such solution has been validated. This allows the system to learn from its experiences, there are two types for adaptation:

1. **Structural Adaptation**

In structural adaptation, formulas and rules are directly applied to stored solution in CBR library. When case is applies on these rules and formulas. Then, CBR system adapts this case and match with new problem.

2. **Derivational Adaptation**

It is a technique to reuse the rules and formulas to produce a new solution of a current problem. Solutions which are retrieved must be stored as additional case in the CBR library so it reproduces new solution to new case.

Several techniques are used in CBR for simple to complex. Techniques are following [3]:

a) Null Adaptation

It uses no adaptation at all. It just applies whatever solution is retrieved to current problem without adapting it. Null adaptation is useful for problems which involve complex reasoning.

b) Parameter adjustment

It is a structured technique which compares specified parameters of retrieve and current case to give a solution in the right direction. This technique used in CBR called JUDGE.

c) Derivational Reply

It is a technique of retracing the method to arrive at old situation which is used to give a new solution in new situation.

d) Model-guided Repair

This technique uses a causal model to guide adaptation. In this technique, also require good understanding of problem domain.

3.2.6 Case base organization

When there is a new problem, we are going to retrieve all relevant cases from case base, the take the appropriate solution of the retrieved case and evaluate this solution, if it is good we will save this problem in the case base, otherwise evaluate it again and do the same stages.

The following figure (3.2) shows the organization of case base:

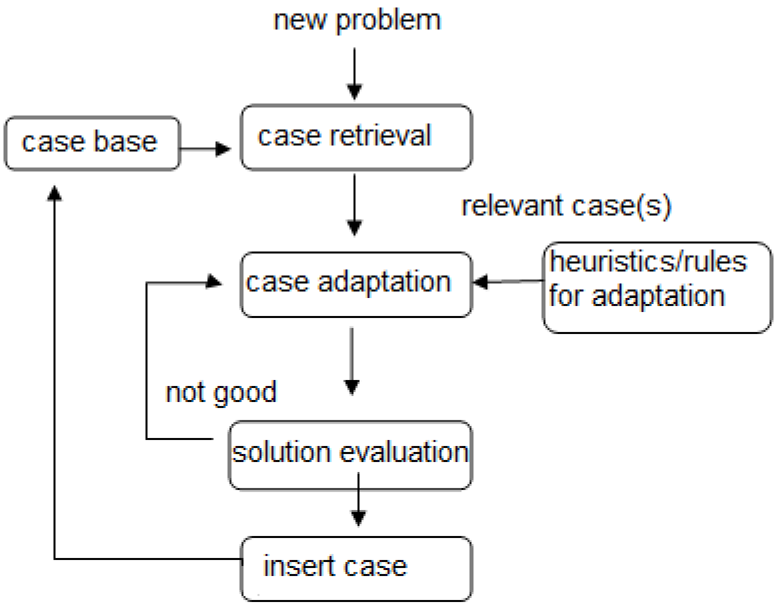


Figure 3.2 case base organization [26]

3.2.7 Advantages and limitations of CBR

CBR is a lazy problem-solving method and shares many characteristics with other lazy problem-solving methods, including advantages and disadvantages. Aha [21] defines the peculiarities of lazy problem-solving methods in terms of three Ds:

a) Defer: lazy problem solvers simply store the presented data and generalizing beyond these data is postponed until an explicit request is made.

b) Data-driven: lazy problem solvers respond to a given request by combining information from the stored data.

c) Discard: lazy problem solvers dismiss any temporary (intermediate) result obtained during the problem solving process.

3.3 Diabetes mellitus dataset

The diabetes mellitus dataset used was collected from military hospital. The dataset contains 140 samples with 70 samples have attributes with missing values and 70 samples have complete data. Each sample record has six attributes. The attributes of data are:

1. Age
2. Gender
3. Fasting glucose test
4. Two-hour OGTT
5. HbA1c measurement
6. Status (normal, pre-diabetic or diabetic)

The attributes data types are shown in below table (3.1):

Table 3.1 five important attributes in diagnosing diabetes mellitus

Name of attributes	Type of value
Age	Integer
Gender	String
Fasting glucose test	Integer
Two-hour OGTT	Integer
HbA1C measurement	Double

3.3.1 Software

Software that will be used for developing this application is eclipse for creating the interface and codes the function of the application. The eclipse use java as the programming language.

3.4 User design

In user design, the flowchart is used to describe the flow of DMDA. When the new diabetes mellitus case is determined, the application will search in the case base for the similar cases and retrieve the most similar case from it. If the case base is matched with the new case, the application will reuse the solution from the similar case as the new solution. If not, the process is going back to retrieve the similar case.

In this application, some part of CBR is used which are retrieved and reused. The flow chart of diabetes mellitus diagnosis system is shown in figure 3.3

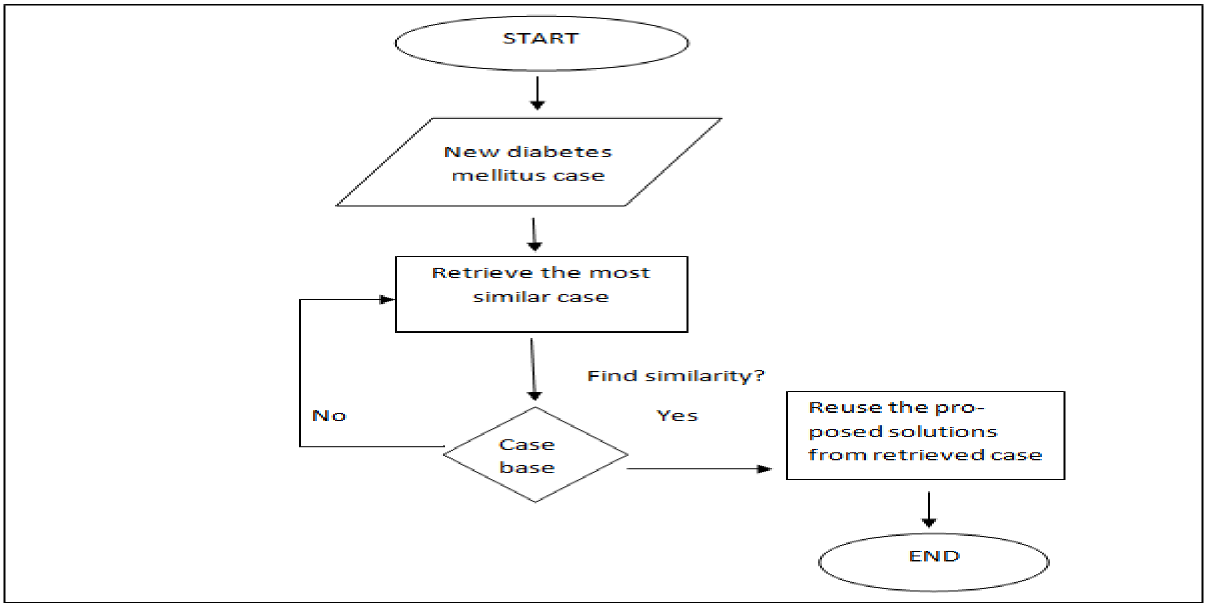

START
New diabetes mellitus case
Retrieve the most similar case
Find similarity?
No
Yes
Case base
Reuse the pro-posed solutions from retrieved case
END

Figure 3.3The flow chart of diabetes mellitus diagnosis system

The user is expert doctors or medical staff whose are responsible to do the diagnosis. When the new diabetes mellitus case is determined, the admin generated the application to case base for search the similar case. Then the application retrieved the most similar case and match with the new case. If the similar case is matched with the new case, the expert doctors reused the proposed solution from the retrieved case as the solution of the new case.

3.5 Summary

CBR reuse the existing data that have been stored in case base as the solution for the new case that are similar. By the end of developing the application, hopefully it will functioning well in assign the patient’s blood glucose to normal that does not have diabetes mellitus, prediabetic that patient is prone to be or diabetic that has strong evidence of having diabetes mellitus. The implementation of the application will be discussed in next chapter

CHAPTER IV

DESIGN, IMPLEMENTATION, TESTING ANDRESULTS

4.1 Introduction

This chapter will discuss about how the implementation stage has been done in developing the diabetes diagnosis system, and discussing the results. The development is involving the Graphical User Interface (GUI) design, case base, and the coding development for entire application. The method used in developing the application and database also is discussed here. Diabetes diagnosis system has been developed using Eclipse. The source code of the application using the Java programming language and the case-base was created using text files.

Below figure (4.1) shows the implemented framework of the DIABETES MELLITUS DIAGNOSIS system (DMDS) using CBR:

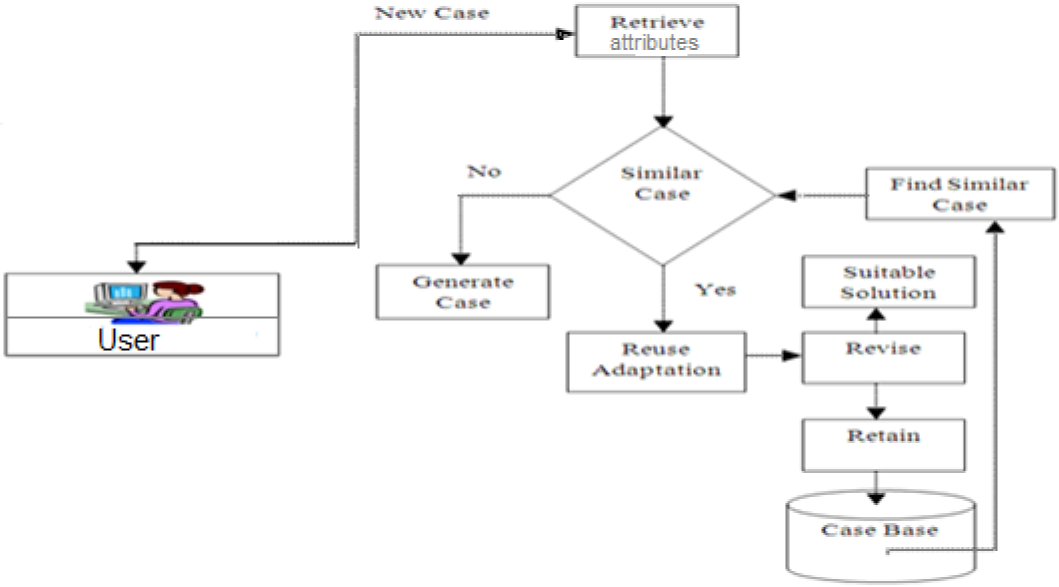


Figure 4.1 DMDS framework using CBR

Bellow figure shows the phases of our developed system:

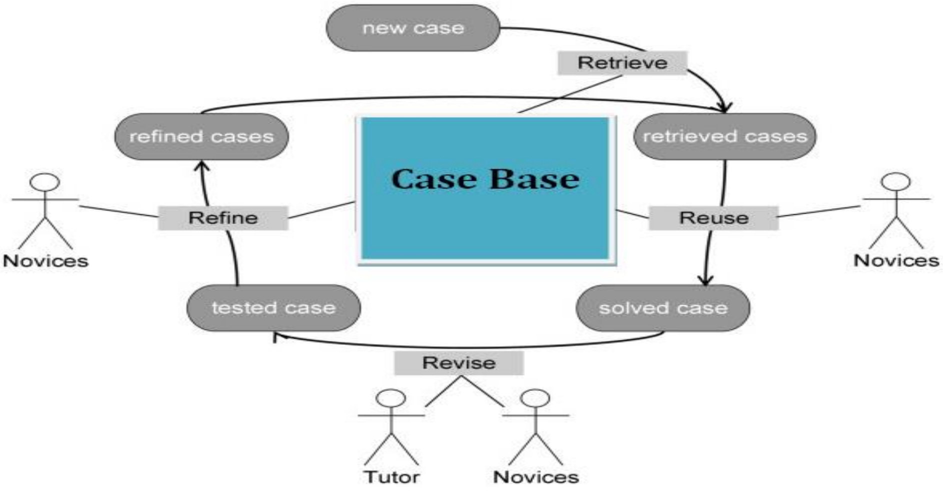


Figure 4.2 DMDS phases

4.2 Development Environment

For this application, it is developed in Eclipse using the Java programming language. Window 7 is used as the operating system with Intel(R) processor and 3 GB of RAM for develop the application environment. This application used diabetics patients ‘dataset which has been collected from Military Hospital for evaluating the CBR algorithm. This dataset is used as the casebase for the application. Table (4.1) below shows the environmental needs for the application development.

Table 4.1 Environmental needs for the application development

Type	Tool	Platform
Programming Platform	Eclipse	Windows
Programming Language	Java	Windows
Operating System	-	Windows 7
Hardware	-	Samsung R430 labtop
Processor	-	Intel (R) Celeron (R) processor with 2.20GHz
RAM	-	3 GB
Case-base	Text file	-

4.3 Designing of Interface

Interface is the layer of the application or system that used to interacts with others. It is used by the user to interact between interfaces.

For our system, it consists of six interfaces. The first interface is weights of diabetes mellitus tests, second interface is values of patient's tests, third one is the diagnosis result , fourth one displays all cases stored in case base, fifth one displays information of system and diabetes mellitus, and the last interface is for retaining cases. Below figures shows the model of diabetes diagnosis system using Case-Based Reasoning:

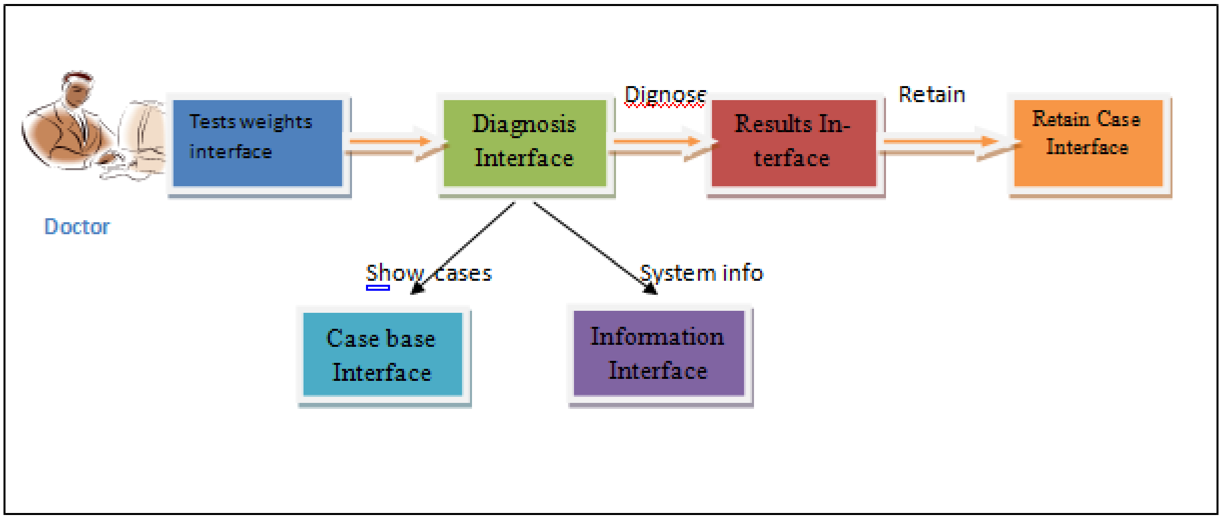


Figure 4.3DMDSinterfaces model

4.3.1 Tests Weights Interface

It's the first interface in the model. It consists of three text fields which are used as the input of weights for fasting test, 2Hours test and HbA1c test, and a button to handle this action. Age and gender have constant weights, age=3, gender=4. Weight is used in Manhattan and Euclidian functions to calculate the distance between new cases and stored cases.Following figure (4.4) is the test weight interface in our system:

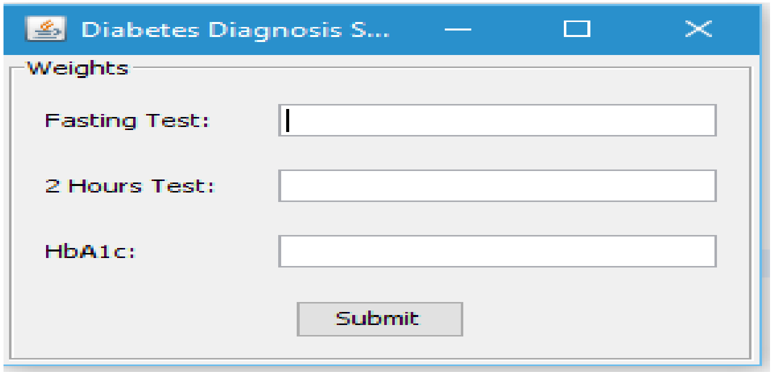


Figure 4.4Tests Weights Interface of DMDS

4.3.2 Diagnosis Interface

It’s the second interface in the model. It’s a page for the user to input the patients‟ information about their Diabetes mellitus. From the diagnose interface, after click the diagnose button, the input will compared with the case-base, retrieve the data from the case-base to do the calculation of diagnosis ‘similarity. After the calculation, the result will display the result of diabetes mellitus with the information of patient and the similarity from the previous case. Figure (4.5) below shows diagnosis interface:

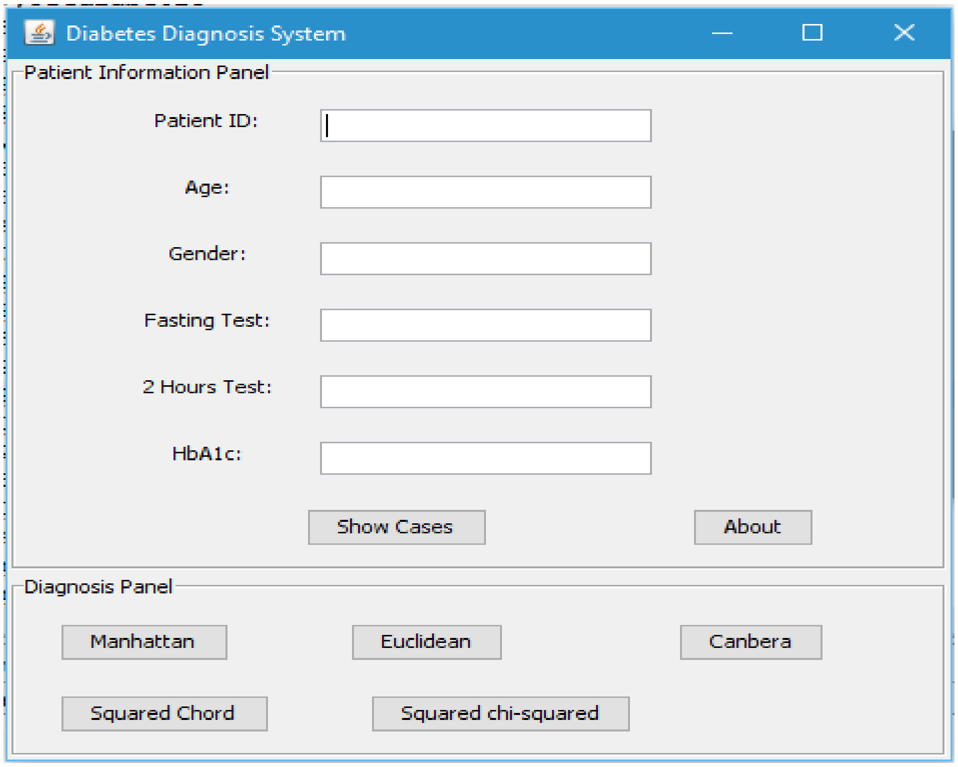
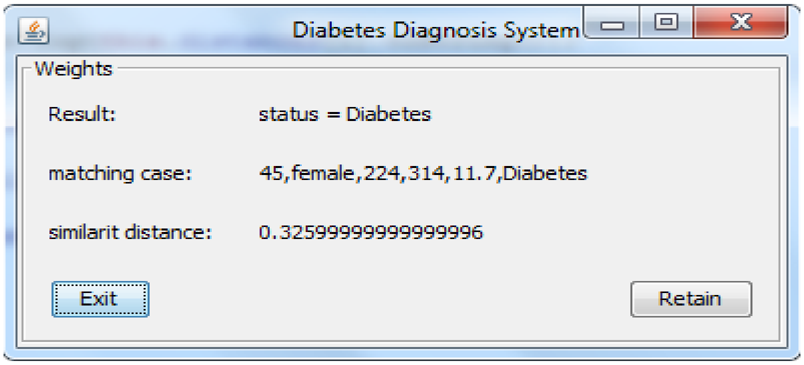


Figure 4.5 Diagnosis Interfaces of DMDS

4.3.3 Results Interface

This is the result interface where the result of the diagnosis will display. In the result interface, it will display the result of diagnosis, best matching case and similarity of the case to the previous case. User can choose either to retain the case or to the end page which is exit interface.

The similarity will be calculated using Manhattan, Euclidean, Canberra, Squared Chord and Squared chi-squared functions. Shown in figure (4.6):



Diabetes Diagnosis System

Weights

Result: status = Diabetes

matching case: 45,female,224,314,11.7,Diabetes

similarit distance: 0.32599999999999996

Figure 4.6 Result Interfaces of DMDS

Following tables' shows the functions' codes which are implemented, table (4.2) shows the code of reading the value of the new attribute value from the user and store in variable newAttributeValue, and reading the older attribute value from the case-base and store it in caseAttributeValue:

Table 4.2 values of new and older cases

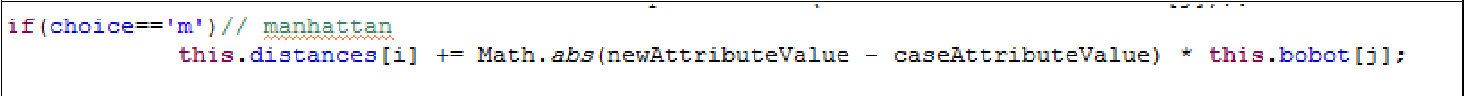
new and older case

`newAttributeValue = Double.parseDouble(this.newCaseAttributeValues[j]);`
`caseAttributeValue = Double.parseDouble( theCase.attributeValues[j]);}`

The second step is to calculate the similarity between these cases, the first function used to calculate the similarity distance is Manhattan function. The variable this.distance stores the similarity distance which is calculated by subtract newAttributeValue form

CaseAttributeValue multiplied by weight of the new case attribute. The following table (4.3) shows the code that calculates the distance using Manhattan function:

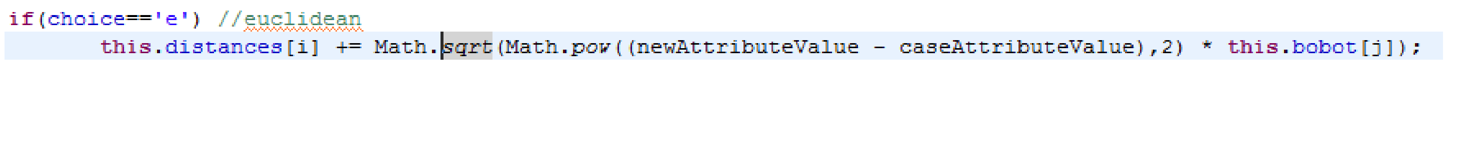
Table 4.3 Manhattan function calculation code

Manhattan function calculation


```
if(choice=='m')// manhattan
            this.distances[i] += Math.abs(newAttributeValue - caseAttributeValue) * this.bobot[j];
```

The second function used to calculate the similarity distance is Euclidean function. The following table (4.4) shows the code that calculates the distance using Euclidean function:

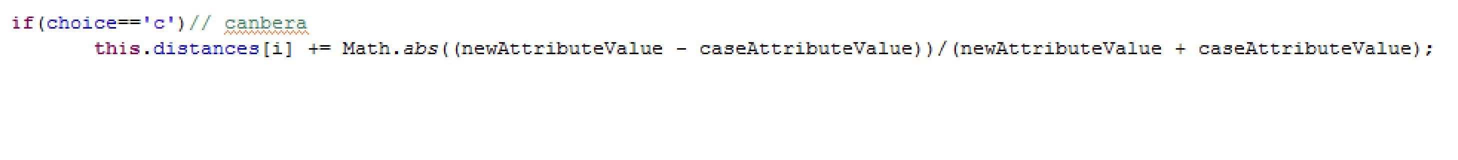
Table 4.4 Euclidean function calculation code

Euclidean function calculation


```
if(choice=='e') //euclidean
        this.distances[i] += Math.sqrt(Math.pow((newAttributeValue - caseAttributeValue),2) * this.bobot[j]);
```

The third function used to calculate the similarity distance is Canberra function. The following table (4.5) shows the code that calculates the distance using Canberra function:

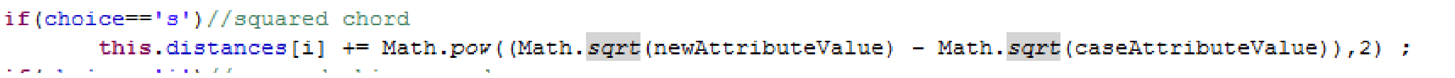
Table 4.5 Euclidean function calculation code

Canberra function calculation


```
if(choice=='c')// canbera
        this.distances[i] += Math.abs((newAttributeValue - caseAttributeValue))/(newAttributeValue + caseAttributeValue);
```

The forth function used to calculate the similarity distance is Squared Chord function. The following table (4.6) shows the code that calculates the distance using Squared Chord function:

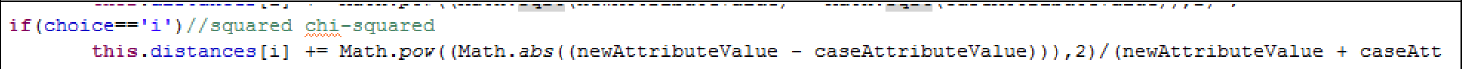
Table 4.6 Squared Chord function calculation code

Squared Chord function calculation


```
if(choice=='s')//squared chord
      this.distances[i] += Math.pow((Math.sqrt(newAttributeValue) - Math.sqrt(caseAttributeValue)),2) ;
```

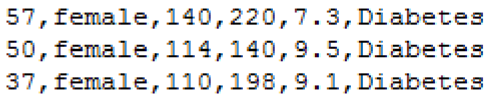
The fifth function used to calculate the similarity distance is Squared chi-squared function. The following table (4.7) shows the code that calculates the distance using Squared chi-squared function:

Table 4.7 Squared chi-squared function calculation code

Squared chi-squared function calculation


```
if(choice=='i')//squared chi-squared
      this.distances[i] += Math.pow((Math.abs((newAttributeValue - caseAttributeValue))),2)/(newAttributeValue + caseAtt
```

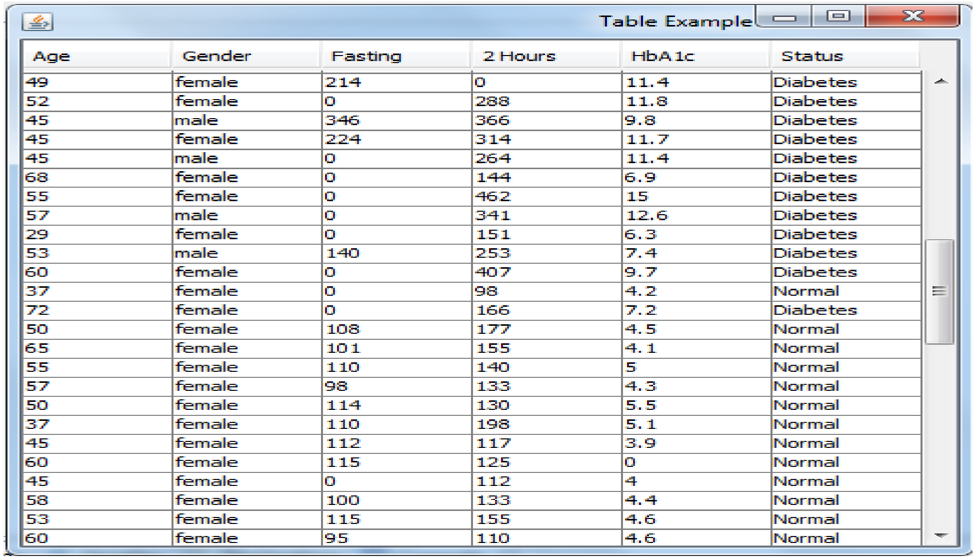
Figure 4.7 shows the list of Diabetic patients' case-base in the text file. There is one file which contains all the dataset from previous cases. This dataset contains six attributes with which are age, gender, fasting testing, two-hour test, HbA1c test and the result of diagnosis. The table contains 140 records of patients from military hospital. age, fasting test, two-hour test and HbA1c are defined as integer, gender and result are defined as String .the attributes are separated with comma. The following figure (4.7) shows cases from the text file:



```
57,female,140,220,7.3,Diabetes
50,female,114,140,9.5,Diabetes
37,female,110,198,9.1,Diabetes
```

Figure 4.7 cases in text file

We can retrieve all cases from the case-base and display them in the interface of case-base, omitting the comma and put every attribute of case in a cell of table interface. Following figure(4.8) shows cases retrieved from case-base and displayed in interface:

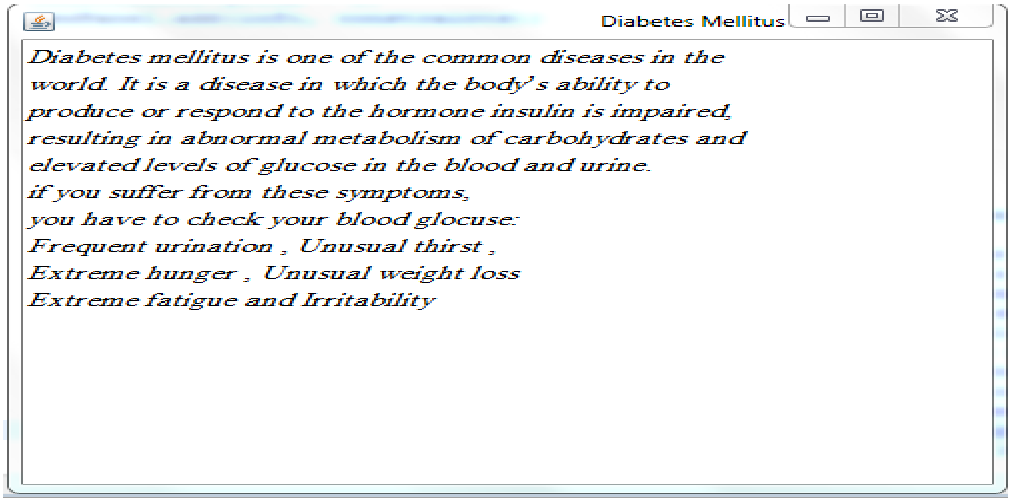


Age	Gender	Fasting	2 Hours	HbA1c	Status
49	female	214	0	11.4	Diabetes
52	female	0	288	11.8	Diabetes
45	male	346	366	9.8	Diabetes
45	female	224	314	11.7	Diabetes
45	male	0	264	11.4	Diabetes
68	female	0	144	6.9	Diabetes
55	female	0	462	15	Diabetes
57	male	0	341	12.6	Diabetes
29	female	0	151	6.3	Diabetes
53	male	140	253	7.4	Diabetes
60	female	0	407	9.7	Diabetes
37	female	0	98	4.2	Normal
72	female	0	166	7.2	Diabetes
50	female	108	177	4.5	Normal
65	female	101	155	4.1	Normal
55	female	110	140	5	Normal
57	female	98	133	4.3	Normal
50	female	114	130	5.5	Normal
37	female	110	198	5.1	Normal
45	female	112	117	3.9	Normal
60	female	115	125	0	Normal
45	female	0	112	4	Normal
58	female	100	133	4.4	Normal
53	female	115	155	4.6	Normal
60	female	95	110	4.6	Normal

Figure 4.8 Diabetic patients Dataset of DMDS

4.3.4 Diabetes mellitus Info Interface

This page consists of briefly explanation about the diabetes mellitus. it is for the general view of user about the diabetes and symptoms of diabetes mellitus. Figure (4.9) shows the info interface:



Diabetes mellitus is one of the common diseases in the
world. It is a disease in which the body's ability to
produce or respond to the hormone insulin is impaired,
resulting in abnormal metabolism of carbohydrates and
elevated levels of glucose in the blood and urine.
if you suffer from these symptoms,
you have to check your blood glocuse:
Frequent urination , Unusual thirst ,
Extreme hunger , Unusual weight loss
Extreme fatigue and Irritability

Figure 4.9 Diabetes mellitus Info Interface of DMDS

4.3.5 Retain Case Interface

This interface is responsible of retaining cases; if result is incorrect the user can correct it then store the problem case as a new case in the case-base, otherwise if the solution is appropriate we will not retain the case. Figure(4.10) shows case retaining interface in our application:

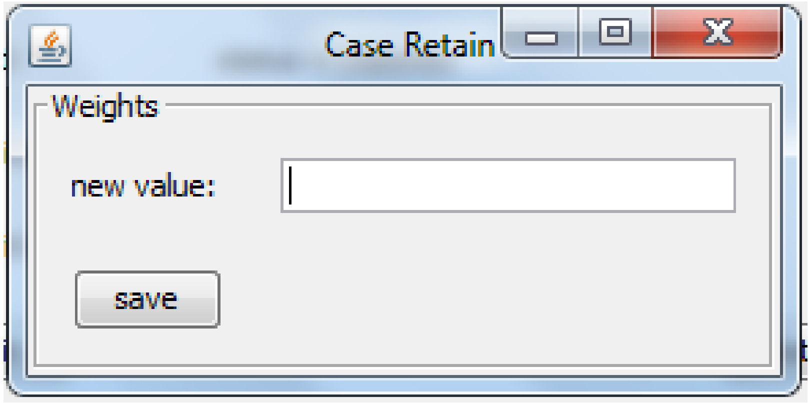
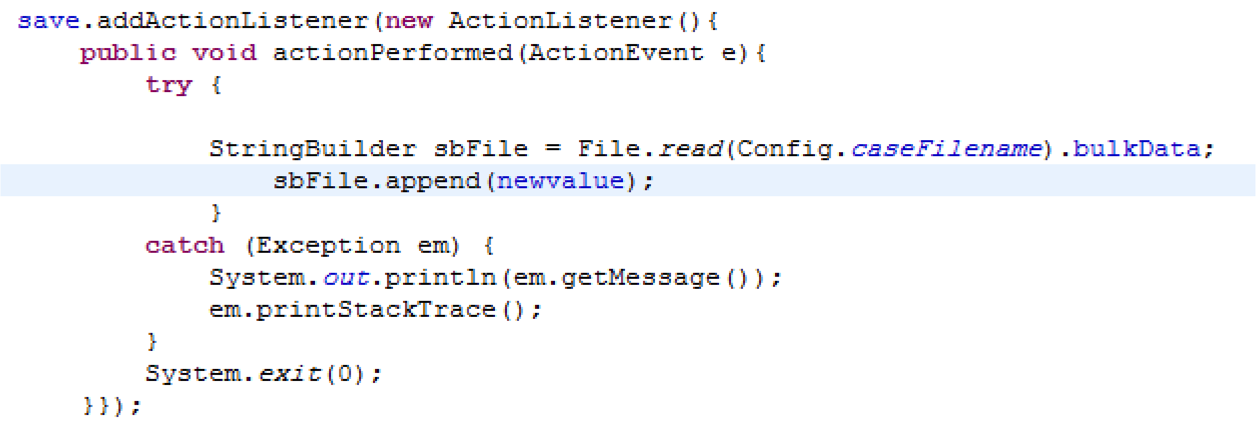


Figure 4.10 Cases Retaining in DMDS

The variable newvalue is the problem case which we want to retain in case-base. Below table(4.8) shows the code of case retaining:

Table 4.8 Case Retain

Retain Case



```
save.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try {

            StringBuilder sbFile = File.read(Config.caseFilename).bulkData;
                sbFile.append(newvalue);
            }
        catch (Exception em) {
            System.out.println(em.getMessage());
            em.printStackTrace();
        }
        System.exit(0);
    }});
```

4.4 Diabetes diagnosis application engine module

In CBR, there are four components that are important during the prediction which are Retrieve, Reuse, Revise and Retain. In developing of the diabetes diagnosis system, retrieve is referring to given a target problem, retrieve cases from memory that are relevant to solve it. A case consists of a problem, its solution and about how the solution is derived. In this application, case retrieval refers to process of finding the nearest case, which includes the solution for the new case within the case-base. After the nearest case is retrieve, the solution from the previous case is reused to solve the new case.

4.4.1 Similarity Measure

Similarity measure is used in problem solving and reasoning to match a previous case (case-base) with the new case to find solution. It select cases that have nearly the same solution that the new case. These similarities can be calculated using these functions describe this function and their equations:

1. **Manhattan distance**

The Manhattan distance is the shortest distance a car would have to drive in a city block structure to get from x to y. since it takes the absolute distance in each dimension before we sum them up, the Manhattan distance will always be bigger or equal to the Euclidean distance, which we can imagine as the linear distance between two points. The following equation (4.1) describes this function:

$$\mathrm{d} \quad = \sum_{\mathrm{i}=1}^{\mathrm{n}} |\mathrm{x}_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}}| \ldots . \text{(4.1)}$$

Where:

d: Manhattan distance

x_{i}: New case

y_{i} : Old case

n: number of compared cases

When we applied this function for calculating the similarity distance between problem case and older case it has scored 76% accuracy and error rate percentage in cases: diabetic, pre-diabetic and normal was 7.1%, 2.6%, and 2.7% sequentially. the reason behind low accuracy rate is when tests has large attributes values, which will be compared with medium or low values attributes of cases values it will result in a large distance.

2. **Euclidean distance**

This is pretty much the most common distance measurement. Its so common, in fact, that it’s often called the Euclidean distance, even though there’s many Euclidean distance measures, as we just learned. It’s defined as equation (4.2):

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \;\ldots.\; (4.2)$$

d: Euclidean distance

x_i: New case

y_i : Old case

n: number of compared cases

This Euclidean distance adds up all the squared distances between corresponding data points and takes the square root of the result. Remember the Pythagorean Theorem? If you look closely, the Euclidean distance is just a theorem solved for the hypotenuse which is, in this case, the distance between x and y. It can get arbitrarily large and is only zero if the data points are all exactly the same.

When we applied this function for calculating the similarity distance between problem case and older case it has scored 76% accuracy and error rate percentage in cases: diabetic, pre-diabetic and normal was 3.2%, 1.7%, and 1.8% sequentially. The reason behind these percentages is the Euclidean distance is pretty solid: it’s bigger for larger distances, and smaller for closer data points.

3. **Canberra Distance**

(Lance and Williams, 1967) examines the sum of series of fraction differences between coordinates of pair of objects. Each term of fraction difference has value between 0 and 1. This distance is very sensitive to a small change when both coordinate near to 0.It’s defined as equation (4.3)

$$d_{ij} = \sum_{k=1}^{n} \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|} \ldots.\; (4.3)$$

Where:

d_{ij}: Canberra distance

x_{ik} : New case

x_{jk} : Old case

n: Number of compared cases

This function scored the maximum accuracy rate when applied to calculate similarity distance. Its accuracy was 92%. And the error rate percentage for cases: diabetic, pre-diabetic and normal was 0.73%, 0.21%, 0.01% sequentially. We consider this function is the best one among the other functions.

4. Squared Chord

Squared chord distance values can range from 0.0 to 2.0, with 0.0 indicating identical proportions of species within the samples being compared. It’s defined as equation (4.4):

$$d \quad = \sum_{i=1}^{n} (\sqrt{x_i} - \sqrt{y_i})^2 \ldots . \, (4.4)$$

Where:

d: Squared chord distance

x_{ik} : New case

x_{jk} : Old case

n: Number of compared cases

When we applied this function for calculating the similarity distance between problem case and older case it has scored 78% accuracy and error rate percentage in cases: diabetic, pre-diabetic and normal was 0.87%, 0.66%, and 0.93% sequentially. the reason behind low accuracy rate is when tests has large attributes values, which will be compared with medium or low values attributes of cases values it will result in a large distance.

5. Squared Chi-Squared

A chi-squared test, also written as χ2 test, is any statistical hypothesis test wherein the sampling distribution of the test statistic is a chi-squared distribution when the null hypothesis is true. Without other qualification, 'chi-squared test' often is used as short for Pearson's chi-squared test.

Chi-squared tests are often constructed from a sum of squared errors, or through the sample variance. Test statistics that follow a chi-squared distribution arise from an assumption of independent normally distributed data, which is valid in many cases due to the central limit theorem. A chi-squared test can be used to attempt rejection of the null hypothesis that the data are independent. It's defined as equation (4.5):

$$d = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{(x_i + y_i)} \ldots. (4.5)$$

Where:

n:number of compared cases

x_i : New case

y_i: Older case

This function scored the minimum accuracy rate when applied to calculate similarity distance. Its accuracy was 72%. And the error rate percentage for cases: diabetic, pre-diabetic and normal was 0.44%, 0.33%, and 0.86% sequentially.

4.5 Testing and result

The five attributes used are Age, Gender, Fasting glucose test, two-hour OGTT and HbA1C measurement as the input of the system. Similarity measure is used in problem solving and reasoning to match a previous case of diabetes mellitus with the new problem to find solution. After find the similarity, the similarity will be calculated using Manhattan, Euclidean, Canberra, Squared Chord and Squared chi-squared functions. The most important features (weight) are determined and it will be used in similarity computation (for Manhattan and Euclidean).

4.5.1 Accuracy measurement

The measure accuracy is computed in Equation 4.6 as follow:

$$\text{Accuracy} = \frac{\text{Correct Diagnosed}}{\text{Total Testing Cases}} * 100\ldots. (4.6)$$

For estimating the accuracy rate of the CBR model, dataset is divided into two sets. One of them is training set that is used for model training and another is test set that is used for estimating accuracy of the model. So, 140 of data are allocated to training data and the 50 is allocated to testing data. The output of the system is normal, prediabetic or diabetic.

CBR algorithm in this application will bring more than 72% accuracy of diagnose of diabetes mellitus, and maximum 94%. The accuracy for all functions which have been used to calculate similarity distance is explained in the table (4.9) below:

Table 4.9 Accuracy rate of similarity functions

Function	Accuracy
Manhattan	76%
Euclidian	76%
Canberra	94%
Squared Chord	78%
Squared chi-squared	72%

The following figure (4.11) is a chart that shows the accuracy rate built in x and y axes, x axe determines number of test cases which have been tested using the similarity functions that discussed in previous section of this thesis, and y axe determines the result of testing either correct or not.

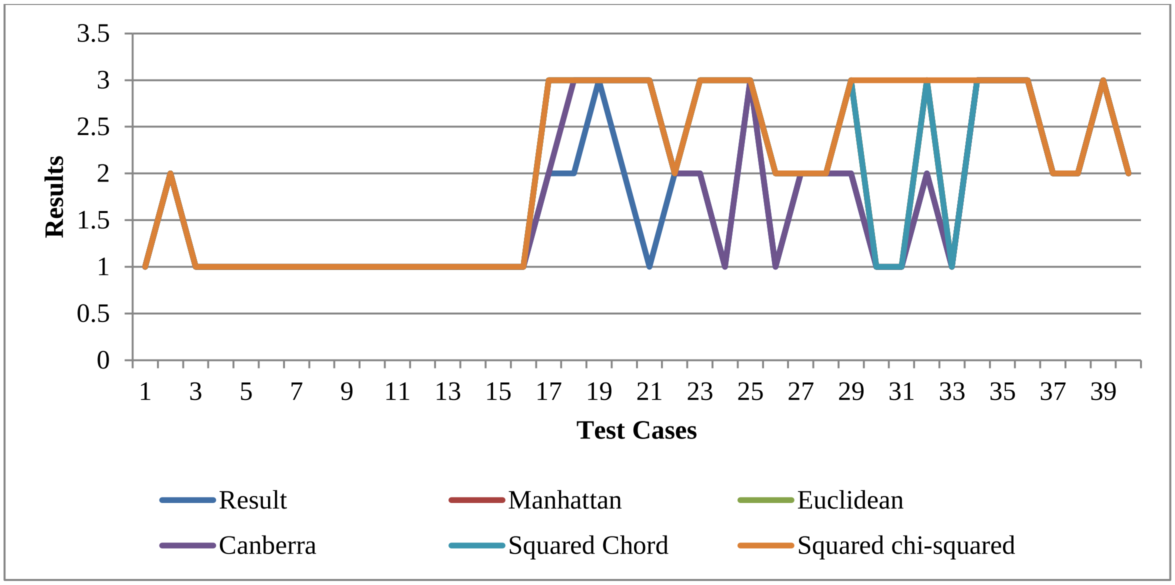


Figure 4.11 similarity functions accuracy rate

The following figure (4.12) is a chart that shows the accuracy rate built in x and y axes, x axe determines number of test cases which have been tested using Manhattan similarity functions that discussed in previous section of this thesis, and y axe determines the correct result.

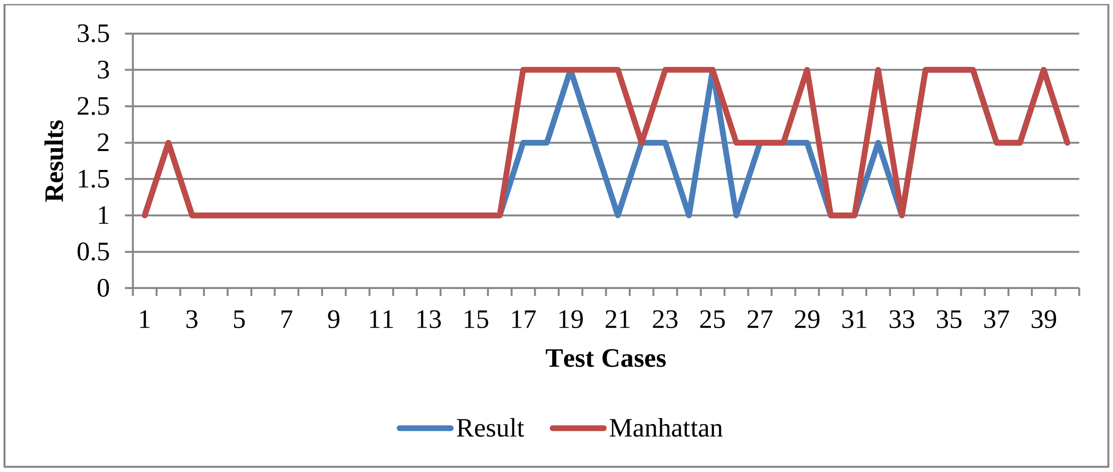


Figure 4.12 Manhattan function accuracy

The following figure (4.13) is a chart that shows the accuracy rate built in x and y axes, x axe determines number of test cases which have been tested using Euclidean similarity functions that discussed in previous section of this thesis, and y axe determines the correct result.

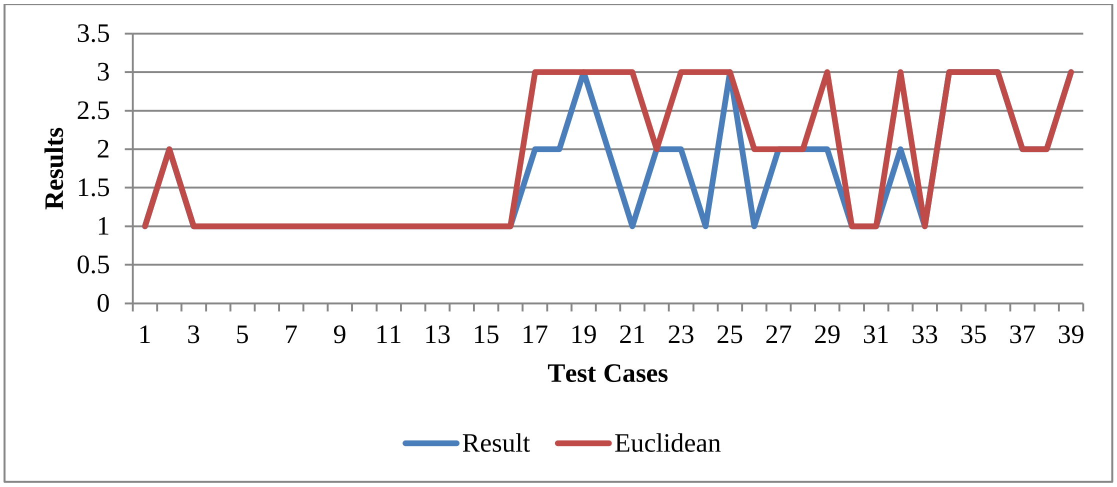


Figure 4.13Euclidean function accuracy

The following figure (4.14) is a chart that shows the accuracy rate built in x and y axes, x axe determines number of test cases which have been tested using Canberra similarity functions that discussed in previous section of this thesis, and y axe determines the correct result.

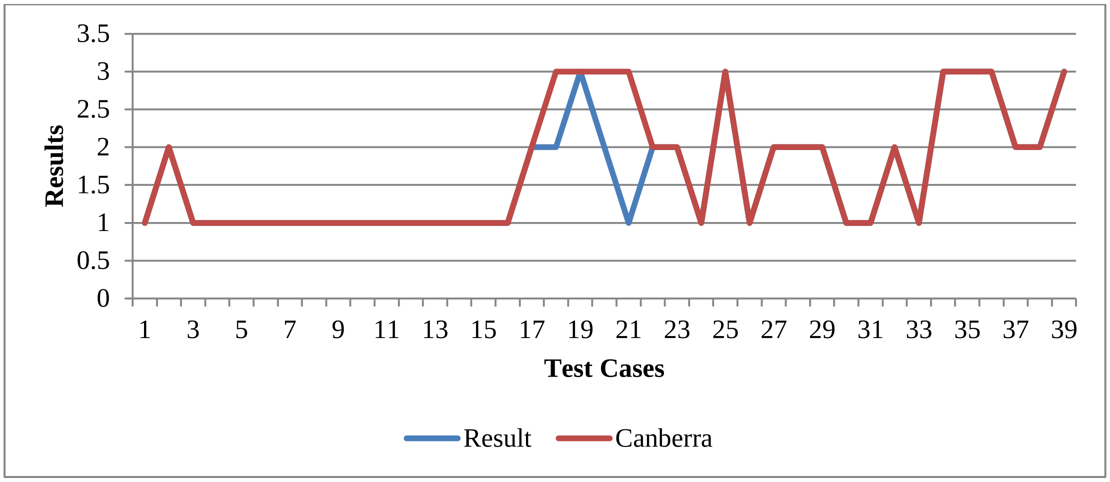


Figure 4.14 Canberra function accuracy

The following figure (4.15) is a chart that shows the accuracy rate built in x and y axes, x axe determines number of test cases which have been tested using Squared Chord similarity functions that discussed in previous section of this thesis, and y axe determines the correct result.

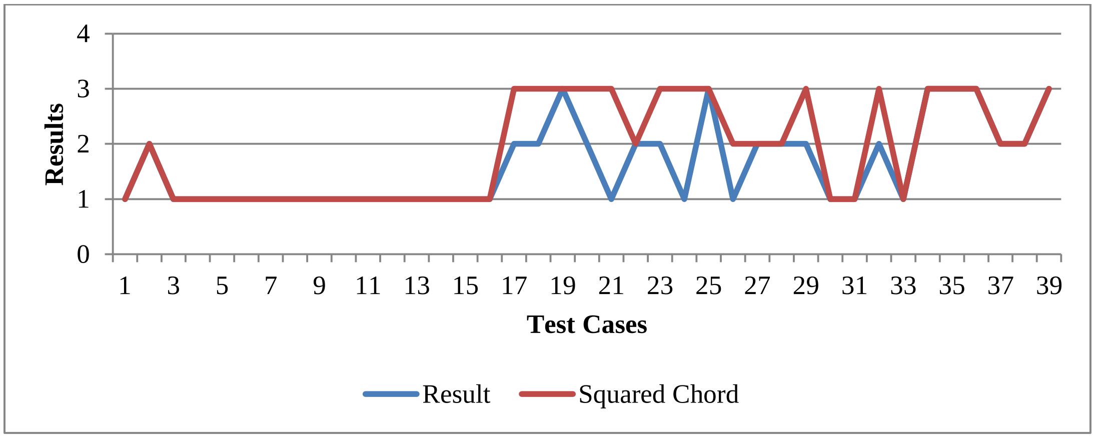


Figure 4.15 Squared Chord function accuracy

The following figure (4.16) is a chart that shows the accuracy rate built in x and y axes, x axe determines number of test cases which have been tested using Squared Chi-squared similarity functions that discussed in previous section of this thesis, and y axe determines the correct result.

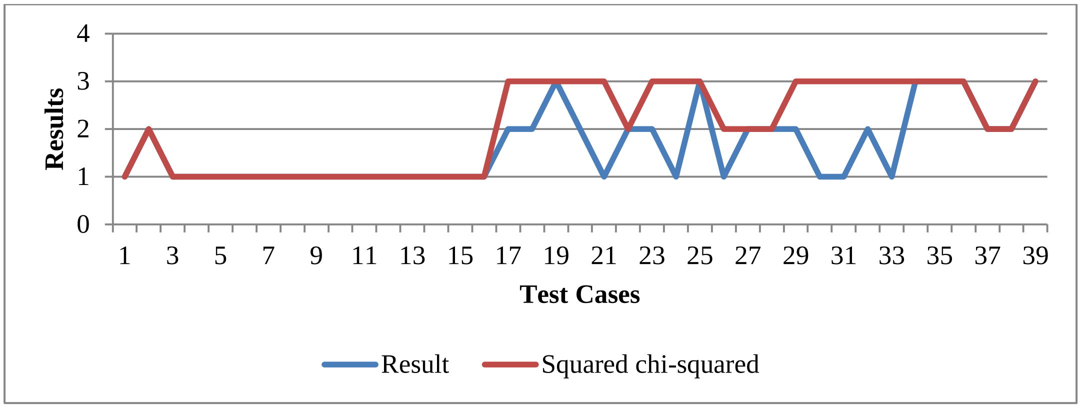


Figure 4.16 Squared chi-squared function accuracy

4.5.2 Root-mean-square deviation (RMSD) formula

Also called root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed. The RMSD represents the sample standard deviation of the differences between predicted values and observed values. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a measure of accuracy, to compare forecasting errors of different models for a particular variable and not between variables, as it is scale-dependent.

Although RMSE is one of the most commonly reported measures of disagreement, some scientists misinterpret RMSD as average error, which RMSD is not. RMSD is the square root of the average of squared errors, thus RMSD confounds information concerning average error with information concerning variation in the errors. The effect of each error on RMSD is proportional to the size of the squared error thus larger errors have a disproportionately large effect on RMSD.

$$\mathrm{RMSE}(X_1, X_2) = \sqrt{\frac{\sum_{i=1}^{n}(X_{1,i} - X_{2,i})^2}{n}} \quad \ldots (4.7)$$

Where:

(x_{1i} – x_{2i})Sup>2 = similarities difference, squared

n = sample size.

The following table (4.10) helps us estimating error rate percentage for all similarity functions which have been applied in this research. C# is the case number from case-base and it value is its similarity value; we took for every status sample of cases.

Table 4.10 cases similarities to calculate error rate

#	C1	C2	C3	C4	C5	C6	C7	C8	chosen	function
Diabetic	3.66	1.56	1.51	3.2	0.24	7.2	1.46	3.28	5	Manhattan
	2.6	1.15	1.1	2.29	0.20	5.11	1.06	2.3	5	Euclidean
	0.45	0.38	0.37	0.43	0.13	0.75	0.29	0.31	5	Canberra
	0.19	0.05	0.05	0.15	0.007	0.61	0.06	0.189	5	S-Chord
	0.38	0.103	0.106	0.31	0.015	1.19	0.12	0.37	5	S-chi-s
Prediabetic	0.751	1.362	0.261	0.85	1.766	1.361	1.726	1.491	3	Manhattan
	0.4495	0.9325	0.177	0.597	1.118	0.892	1.082	1.0039	3	Euclidean
	0.0029	0.00306	0.0008	0.00148	0.0053	0.0039	0.00536	0.0048	3	Canberra
	0.0288	0.04628	0.0016	0.02143	0.0862	0.039	0.0852	0.0487	3	S-Chord
	0.0567	0.09202	0.0032	0.04273	0.1678	0.0764	0.1649	0.0965	3	S-chi-s

Normal	0.936	0.541	2.33	0.522	0.495	2.64	3.14	1.23	5	Manhattan
	0.5585	0.367	1.5	0.33	0.347	1.84	2.15	0.7638	4	Euclidean
	0.0038	0.0017	0.0065	0.00249	0.0019	0.0117	0.01318	0.0046	2	Canberra
	0.0407	0.00673	0.1328	0.01122	0.0089	1.106	1.128	0.0502	2	S-Chord
	0.0796	0.01346	0.2592	0.0223	0.0179	1.11	1.15	0.0983	2	S-chi-s

The following table (4.11) shows the error rate for all similarity functions when diagnosis case is for diabetic patients. The error rate percentage has been calculated using RMSE formula.

Table 4.11error rate (diabetic case) calculated using RMSE formula

Function	Error rate
Manhattan	7.14%
Euclidean	3.2%
Canberra	0.73%
Squared Chord	0.44%
Squared chi-squared	0.87%

The following table (4.12) shows the error rate for all similarity functions when diagnosis case is for pre-diabetic patients. The error rate percentage has been calculated using RMSE formula.

Table 4.12 error rate (Prediabetic case) calculated using RMSE formula

Function	Error rate
Manhattan	2.6%
Euclidean	1.7%
Canberra	0.21%
Squared Chord	0.33%
Squared chi-squared	0.66%

The following table (4.13) shows error rate for similarity functions when diagnosis is for normal patients. The error rate percentage has been calculated using RMSE formula:

Table 4.13 error rate (Normal case) calculated using RMSE formula

Function	Error rate
Manhattan	2.783879398
Euclidean	1.844240551
Canberra	0.011447917
Squared Chord	0.859422885
Squared chi-squared	0.934491109

Error rate has been calculated using RMSE formula, the results shown in figure (4.17) below mentions that the function which gives minimum error is Canberra. The x axe is the number of tested cases and y axe is the error rate percentage for all similarity functions:

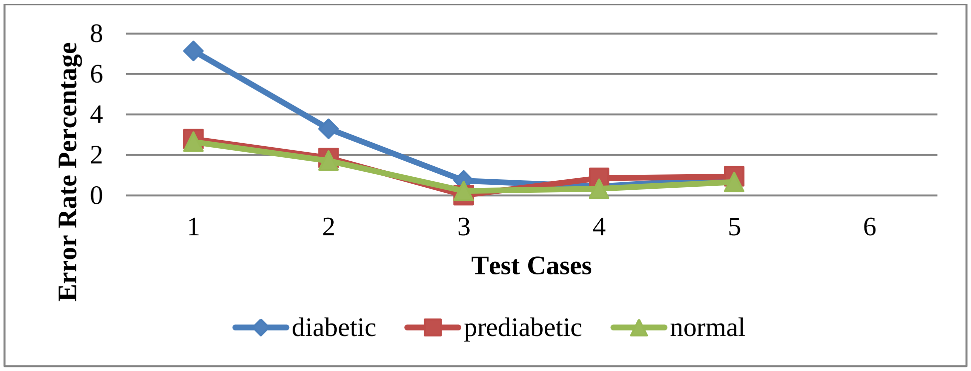


Figure 4.17 error rate of similarity functions using RMSE formula

4.6 Discussion

The success of this work will permit to leverage the development of CBR systems in medicine. It will become possible to develop a web service to federate the CBR process across several domains of medicine. This work will permit patients reuse of CBR systems and develop them. It will also provide the basis for developing a CBR shell for rapid development of CBR systems in medicine.

The above results have provided some indications on the factors affecting the performance of the CBR-system, such as the range of values affects the similarity distance between two cases.

Our work will also help new doctor diagnosing this dangerous mellitus, also increase the availability and the number of resources and activities for people with diabetes, their families and other interested parties.

4.7 Summary

Diabetes diagnosis system design is presented with the development environment, designing of interface is discussed. The application applies the CBR technique. The next chapter will discuss about the testing and discussion about the result of the testing.

CHAPTER V

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The purpose of this algorithm is to serve as doctor diagnostic assistant and aid the young physicians to check their diagnosis.

As mentioned in Chapter 1 Introduction, The objectives for this application are:

1) To develop an intelligent decision support application for diagnosis the diabetes mellitus in order to classify the patient status normal, prediabetic or diabetic.

2) To apply the CBR algorithm in the diabetes mellitus diagnosis application.

The objectives of this application that stated are achieved. The result of the testing does not achieved 100%; but we achieved a high percentage of accuracy.

CBR methods help to compensate for lack of experience of young medical staffs. The inexperience staffs need the guidance from the experience staffs to improve their skill in handling the diagnosis.We have many contributions we have achieved:

1- A survey on trends and developments of recent medical CBR systems has been done.

2- A case-based reasoning system is developed to prove that it is possible to diagnose diabetes mellitus previously only manually diagnosed

3- how a similarity matching algorithms improves system performance.

4- Reduce the time required to come to a decision particularly in an emergency case.

There are some suggestions and recommendations that should be done in order to improve the application as follow:

1. Meet the expertise in the medical domain about the Diabetes mellitus to find out the most important attributes that they used in doing the diagnosis of Diabetes mellitus.
2. Develop a special application to diagnose diabetes mellitus for gestational women.
3. For next version this application can be implementing inside the mobile application due to the development of technology nowadays.

REFERENCES

1. Begum, S., Ahmed, M.U. and Funk, P., 2009. Case-based systems in health sciences: a case study in the field of stress management. Wseas transactions on systems, 8(3), pp.344-354.
2. Jha, M.K., Pakhira, D. and Chakraborty, B., 2013. Diabetes detection and care applying CBR techniques. Int. J. Soft Comput. Eng, 2(6), pp.132-137.
3. Watson, I., 1998. Applying case-based reasoning: techniques for enterprise systems. Morgan Kaufmann Publishers Inc..
4. Mathers, C.D. and Loncar, D., 2006. Projections of global mortality and burden of disease from 2002 to 2030. PLoS medicine, 3(11), p.e442.
5. Mathers, C.D. and Loncar, D., 2006. Projections of global mortality and burden of disease from 2002 to 2030. PLoS medicine, 3(11), p.e442.
6. World Health Organization, 2013. Diagnostic criteria and classification of hyperglycaemia first detected in pregnancy.
7. NIH Publication No. 14ion, 2013. Report Number: WHO/NMHPrediabetes
8. Ambilwade, R.P., Manza, R.R. and Gaikwad, B.P., 2014. Medical expert systems for diabetes diagnosis: a survey. International Journal of Advanced Reserach in Computer Science and Software Engineering, 4(11).
9. Olaniyi, E.O. and Adnan, K., 2014. Onset diabetes diagnosis using Artificial neural network. International Journal of Scientific and Engineering Research,5(10).
10. Akter, M., Uddin, M.S. and Haque, A., 2009. Diagnosis and management of diabetes mellitus through a knowledge-based system. In 13th International Conference on Biomedical Engineering (pp. 1000-1003). Springer Berlin Heidelberg.
11. Zeki, T.S., Malakooti, M.V., Ataeipoor, Y. and Tabibi, S.T., 2012. An expert system for diabetes diagnosis. American Academic & Scholarly Research Journal, 4(5), p.1.
12. Schank, R.C., 1983. Dynamic memory: A theory of reminding and learning in computers and people. cambridge university press.
13. Leake, D.B., 1996. CBR in context: The present and future. Case-Based Reasoning, Experiences, Lessons & Future Directions, pp.1-30.
14. Aamodt, A. and Plaza, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI communications, 7(1), pp.39-59.
15. Slade, S., 1991. Case-based reasoning: A research paradigm. AI magazine,12(1), p.42.

16. Kolodner, J., 1993. Case based reasoning. Morgan Kauffman. San Mateo CA.

17. De Mantaras, R.L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., T COX, M.I.C.H.A.E.L., Forbus, K. and Keane, M., 2005. Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review, 20(3), pp.215-240.

18. López, B., 2013. Case-based reasoning: a concise introduction. Synthesis Lectures on Artificial Intelligence and Machine Learning, 7(1), pp.1-103.

19. Finnie, G. and Sun, Z., 2002. Similarity and metrics in case-based reasoning. International journal of intelligent systems, 17(3), pp.273-287.

20. Cunningham, P., 2009. A taxonomy of similarity mechanisms for case-based reasoning. IEEE Transactions on Knowledge and Data Engineering, 21(11), pp.1532-1543.

21. Aha, D.W., 1998. The omnipresence of case-based reasoning in science and application. Knowledge-based systems, 11(5), pp.261-273.

22. Kolodner, J.L., 1996. Making the implicit explicit: Clarifying the principles of case-based reasoning. Case-based reasoning: Experiences, lessons & future directions, pp.349-370.

23. Watson, I. and Marir, F., 1994. Case-based reasoning: A review. The knowledge engineering review, 9(4), pp.327-354.

24. Richter, M.M., 1995. The knowledge contained in similarity measures.

25. Global report on diabetes, 2016.World Health Organization, Geneva.

26. Kirsh, D., 1991. Foundations of AI: the big issues. Artificial intelligence,47(1-3), pp.3-30.