Development of a Knowledge-Based System for Diagnosis and Treatment of Obesity

Abebaw Agegne[1,†], Ayodeji Olalekan Salau[2,†], Ayalew Belay[3,†], Anupam Singh[4,*,†], Nigus Wereta Asnake[5,†] and Sepiribo Lucky Braide[6,†]

[1]*Department of Computer Science, Debark University, Debark, Ethiopia*
[2]*Department of Electrical/Electronics and Computer Engineering, Afe Babalola University, Ado-Ekiti, Nigeria*
[3]*Department of Computer Science, AAU University, Addis Ababa, Ethiopia*
[4]*Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun, India*
[5]*Department of Information Technology, Debark University, Debark, Ethiopia*
[6]*Department of Electrical and Electronics Engineering, Rivers State University, Port Harcourt, Nigeria*

Abstract
Obesity is a major public health issue that affects both industrialized and developing countries. Obesity is a varied and complex issue that necessitates diagnosis and treatment. Various research projects have attempted to create and develop a knowledge-based system (KBS). Physicians' workload and medical errors could be reduced by using a KBS. The goal of this study was to create a KBS that can guide clinicians and patients through the diagnosis and treatment of obesity. A case-based reasoning approach is applied in this study. Case-based reasoning is a general artificial intelligence paradigm that has been studied in the context of improving human decision-making and has gotten a lot of interest in the development of KBSs. To acquire data, a purposive sampling technique is employed. The explicit knowledge is collected from the literature. To develop the prototype, JCOLIBRI software is used. In the development of the knowledge base system, 243 obesity cases were used. During evaluation, six experts were involved. User acceptance testing, case retrieval in terms of precision and recall, and case similarity testing are all used to assess the system's performance. The user acceptance testing conducted showed that 90% of the evaluators found the system to be easy to use, efficient with respect to time and speed of the system. With precision and recall, the performance of the system is 71% and 80% respectively. The study found that there is a shortage of physicians and medical experts who can diagnose and treat obesity. That means that a single physician or medical doctor will not be able to treat a large number of people in a short amount of time. The proposed system in this study could help to alleviate the shortage of experts who can diagnose and treat obese people. The system helps to decrease the amount of work that physicians have to do by allowing them to delegate the same patient cases to other experts.

Keywords
Case-based reasoning (CBR), knowledge-based system (KBS), obesity, cases, artificial intelligence

1. Introduction

In the medical field, a physician's major responsibilities are diagnosis, categorization, and treatment. Because the medical domain, such as the psycho-physiological domain, is multifaceted

ACI'23: Workshop on Advances in Computational Intelligence at ICAIDS 2023, December 29-30, 2023, Hyderabad, India
*Corresponding author.
†These authors contributed equally.
anupam2007@gmail.com (A. Singh)
0000-0002-6264-9783 (A. O. Salau)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

and complicated, it frequently needs the development of a system that employs a number of artificial intelligence techniques. Obesity [1] is one of the psycho-physiological domains in the medical realm. Obesity has long been recognized as a public health issue, with huge rises in incidence in both emerging and industrialized countries over the last 30 years dubbed a ”obesity epidemic.” Obesity is a major public health issue that affects industrialized and developing countries alike [2].Obesity is a global health problem and affects more billions of people of all ages and sex [1].

According to WHO, obesity and overweight are both defined by abnormal or excessive fat buildup that might have harmful health repercussions [3, 4]. The symptoms of obesity are classified as case problems like short-term problems, long-term problems, and Psychological problems [5]. The short-term problem symptom is the day-to-day life activity happens like breathlessness, snoring, increased sweating, difficulty sleeping, and inability to manage sudden physical activity. The long-term problem symptom is the symptoms that you may not know but has seriously harmed your health, such as hypertension, diabetes, infertility, and the like [6, 7]. Psychological problem symptom is a problem to do with mental health like having low self-esteem, poor self-image (not liking how you look), low confidence levels, feeling isolated from the society. The major important factor in influencing our weight is lifestyle choices [8]. For example, bad food choices such as processed or fast food that is high in fat, a lack of fruits and vegetables, excessive alcohol use, low energy expenditure, and heredity. In addition to unhealthy food choices, the most common cause of obesity is physical inactivity, overeating, medications, psychological factors, diseases like hypothyroidism, diabetes, pituitary cancer, and social issues [5] There is no easy cure and no country has yet achieved major reductions in obesity rates [3]. Excess body weight has considerable impacts on individuals, the health care system, and society. Only knowing the symptoms is not the treatment of obesity. The diagnosis should be refined in order to select the most appropriate and effective treatment. A professional doctor can test obesity treatments with all epidemics and related disorders [3]. As a result, diagnosing and treating obese patients takes time and is difficult to follow up on. To overcome such issues, a knowledge base system must be developed and designed [1]. Knowledge-based systems (KBS) employ intelligent reasoning to solve a problem that would normally necessitate a significant amount of human work, time, and skill. It provides dependable solutions for daily decisions and processes, as well as a substantial amount of data. Knowledge-based systems (KBSs) can reduce the burden of physician and medical errors during the diagnosis and treatment of patients. KBS are developed by duplicating human intelligence to assist health physicians in the decision-making process without asking the specialist doctors. The system does not replace specialists, but it does assist physicians in diagnosing and treating patients. As a result, KBS appear to be a promising solution for avoiding time wastage and help to minimize medical failure of a medical system [**?**].

According to the WHO, the global burden of deaths of persons having overweight and obesity is recorded to be 2.8 million deaths per year and 35.8 million disability-adjusted life [**?**]. Despite the sustained high prevalence of under nutrition, overweight and obesity are becoming an emergent public health hazard in developing countries [1, 8**?** , 9, 10]. In Africa, the body mass index (BMI) has risen over time in all regions, following the global trend. Few remarkable research to diagnose the human diseases using soft computing approaches were published [11, 12, 13, 14, 15].

The remaining section of this paper is organized as follows. In section 2, related literature work is provided, while the methodology is discussed in section 3. In section 4, the model/architecture of the system is presented. In section 5 and 6, the system evaluation and conclusion are presented respectively.

2. Related literature work

International and national research on knowledge-based systems (KBSs) is being conducted to develop alternatives in the field of medical diagnosis [8]. Ndukwe et al. [1] developed an improved rule-based expert system for the diagnosis of Obesity. The study was conducted using rule sets developed from getting information from Obesity experts to build a system. This developed RB system cannot learn and does not support complex domains. This makes the system manual work. The key limitation of the RB system is inference efficiency problems i.e. when there was a large rule base. Also, it is not possible to conclude rules when there are missing values in the input data. So, the case-based reasoning technique is the alternative solution to develop a KBS. The existing developed rule-based expert system is done for diagnosis only but does not support the treatment of obesity [1]. It is only applied for diagnosis purposes with a few cases and features. In addition to this, the previous studies of diagnosis and treatment of obesity are not learning KBS as most are developed with a rule-based approach. Therefore, by considering this gap the authors proposed to design a KBS to diagnose and treat obesity. KBS can help alleviate the shortage of physicians and medical doctors who help to identify and treat obesity. This aids a single physician or medical doctor can provide care to a large number of patients in a short amount of time. The solution to this study's dilemma could help to alleviate the shortage of experts who can diagnose and treat obese people. It will help reduce the workload on physicians by allowing them carry out the same patient cases with various experts.

The remaining section of this paper is organized as follows. In section 2, related literature work is provided, while the methodology is discussed in section 3. In section 4, the model/architecture of the system is presented. In section 5 and 6, the system evaluation and conclusion are presented respectively.

3. Methodology of the study

3.1. Data collection

The relevant information was gathered from both primary and secondary sources in order to complete this study. Particularly, 243 obesity cases were collected from University of Gondar referral hospital.

3.2. Sampling method and sample size

For knowledge acquisition purposes, the purposive sampling method is used to select the domain experts from the University of Gondar referral hospital.

4. Knowledge acquisition (KA)

Every knowledge engineer should consider the two basic important steps in knowledge engineering, which are important through the growth of KBS. The first one is knowledge elicitation from domain experts and various relevant documents. The second step is represented the collected knowledge that we get from the domain experts with the appropriate knowledge representation method [8]. KA is the method of gaining useful information (applicable data) from domain experts as well as from other sources such as books, research papers, manual/guidelines, and transfer to the knowledge base (case base. KA is the most important step in KBS development, but at the same time, it is the most difficult that requires great care, patience, and attention. The first and most time-consuming phase in the creation of KBS is KA. There are several phases in the KA process. Some of these are as follows: choosing a problem for the computer to answer, interviewing an expert, questionnaires, observation, record reviews, codifying the information in some representation language, and improving the knowledge base by testing and expanding its capabilities [**?**]. During KA, we are interested with how information is collected and where it occurs, since this impacts the system's utility [9]. For this study, primary and secondary sources are required. The knowledge acquisition process is applied to get primary sources (tacit knowledge) from the domain experts using a semi-structured interview.

4.0.1. Knowledge acquisition from domain expert

In this step, knowledge is gathered from domain experts. Domain experts are professionals and an experienced people who can solve a problem in a specific domain area. Obtaining knowledge from domain experts involves understanding how they perform a specific task and describes what general knowledge they have about the domain area. Interviews are one type of knowledge elicitation strategy that includes asking domain experts how they accomplish their duties and achieve success. In the heads of experts (tacit knowledge), the most important knowledge lies. Tacit knowledge has a life with limits (when the domain expert dies, the knowledge also dies with the expert). Therefore, to be used and understood by non-experts, the enormous amount of tacit information in expert minds should be codified and digitized. To collect the relevant knowledge the authors applied a semi-structured interview technique. Six domain experts from the domain area were selected and the purposive sampling technique was used. To minimize the constraints of KA tasks, experts were chosen to optimize the acquisition process based on their educational credentials, years of experience, and immediate job positions in the domain field.

4.0.2. Knowledge acquisition from the relevant document

To create KBS as a secondary data source, explicit knowledge from the appropriate document is gathered. Relevant obesity-related documents were evaluated in order to elicit explicit information for this study. Books, published articles, Ethiopia Standard Treatment Guidelines for General Hospital and manuals, and other overweight and obesity treatment guidelines are among the documents.

4.0.3. Knowledge acquisition from obesity cases

After collecting knowledge from domain experts and appropriate papers, the acquired knowledge from obese cases is used. The domain specialists not only provide their knowledge, but also assist in the acquisition of knowledge from the obesity record files. These instances have been discovered in the leprosy ward. Age, sex, family history, BMI, and waist circumference were among the factors acquired by the writers that were not included in the text. These variables were acquired with the help of a domain expert from obesity cases at the University of Gondar referral hospital. These factors are crucial in determining whether or not a patient is obese. All of these obesity-related factors were gathered from obese patients at the University of Gondar referral hospital. All of these elements were gathered with the help of domain specialists. The researcher tried to call the obese patient directly from the hospital, but due to the coronavirus, no one came to the hospital.

5. Knowledge modeling

The knowledge modeling is performed after gaining the knowledge (data) from domain experts and related document. It is a critical step in the KA process to know the problem and to plan the representation steps of the knowledge. In the knowledge acquisition phases, knowledge engineer collects both tacit and explicit knowledge. The knowledge engineer will attempt to understand both the tacit and the explicit parts of the knowledge and then represent the knowledge with diagrams. The knowledge engineer must then create an abstract model of everything that was addressed throughout the knowledge acquisition step. Different knowledge modeling strategies exist, such as hierarchical and decision tree structures. The decision tree structure was chosen for this study to simulate how obesity diagnosis is carried out.

From the Fig. 1, the oval symbol represent the idea of continuing for the next diagnosis. That is the patients has no psychological symptoms, then the patients test by other criteria when the patients encounter them.

The symbol has no other meaning, only used as a purpose with the variable X. As shown in Fig. 1, the diagnosis and treatment of obesity are mostly determined by BMI and waist circumference. After checking the short-term and long-term symptoms of obesity, the patients will be checked by measuring their BMI and waist circumference.

5.1. Design the model of CBR for obesity

The next step is to encode the knowledge into computer system after gathering cases and expertise from domain experts as well as numerous relevant papers. The case structure will be built when the knowledge engineer obtains the data through interviews and document analysis. To create the case structure, first decide whether you want simple or compound attributes. After selecting an attribute type, assign a weight value to each important attribute and select the appropriate local similarity for each. Following the creation of the case structure, the knowledge engineer should configure the connectors by selecting from a variety of connector types such as SQL database, plain text, or other connectors. When a new problem meets the prototype, the system looks for each characteristic from the entered cases and stored solved cases in the case base to find a retrieved case for the new case. The system computes the local similarity since JCOLIBRI provides distinct local similarity metrics for each data type. It identifies the global similarity of the new case with previously solved instances by multiplying the weight of each attribute with the local similarity calculation result after computing the local similarities. The prototype system ranks relevant retrieved examples based on their global similarities after computing global similarities between the current case and previous cases. When we present a new problem, we fetch the pertinent cases from the case database. If possible, reuse/adaption is the solution to the preceding instance. Revise is the application of knowledge that frequently leads to a revision of that knowledge or cases based on the expert's experience. If the solution is successful, the solution is retained. From Fig. 1, general knowledge refers to the type of knowledge such as, vocabulary knowledge, adaptation knowledge, and retrieval knowledge.

When the new case (query) from the user comes, the first search from the case base. If the search case is exactly matched with the stored cases, use the solved cases directly. If the retrieved case is a partial match from the case base, use the adapt solution. If the retrieved case is new (not similar to the case base), by using the adaptation process, determine the solution for the new case. The adaptation (reuse and revision) means a modification of solutions of former similar cases to fit for a current one. But, using the suggested solutions directed may have a risk in diagnosis and treatment condition. Therefore, the expert adapted by modifying the variations between the old case and the current cases.

In this situation, cases might be maintained based on the result of the proposed solution for storing purposes. Case variations are updated in addition to adaption if the retrieved case differs from the new case. Finally, in the case-based reasoning system's retention phase, the experience gained from the freshly solved case is saved for future use. As a consequence, the newly solved obesity case is saved in the case database for future reference.

5.2. CBRS for obesity diagnosis and treatment

To develop the CBRS for obesity diagnosis and treatment, several steps are performed with the JCOLIBRI tool. Some of these are gathering cases and knowledge of the context, modeling an appropriate case representation, computing similarity measures, implementation of retrieval features, and creating user interfaces. To do this, an interdependent task in JCOLIBRI must be configured first. These are configuring all CBR tasks, methods, connectors, case structure, and building the case base.

5.2.1. Managing case structure

Managing case structure is the fundamental task in a CBR system in JCOLIBRI. In this task, the description and solution attribute, cases are defined clearly. Those acquired cases are saved in a plain text file format. JCOLIBRI generates codes automatically and stores them in XML file format when creating a case structure. The important attributes are declared with a higher weight value.

On the right side of the case structure window, as illustrated in Fig. 2, the data type, weight, and local similarity of the selected attribute may be specified. The right side of the window contains the attribute's property values as cases, whilst the left side of the window contains the

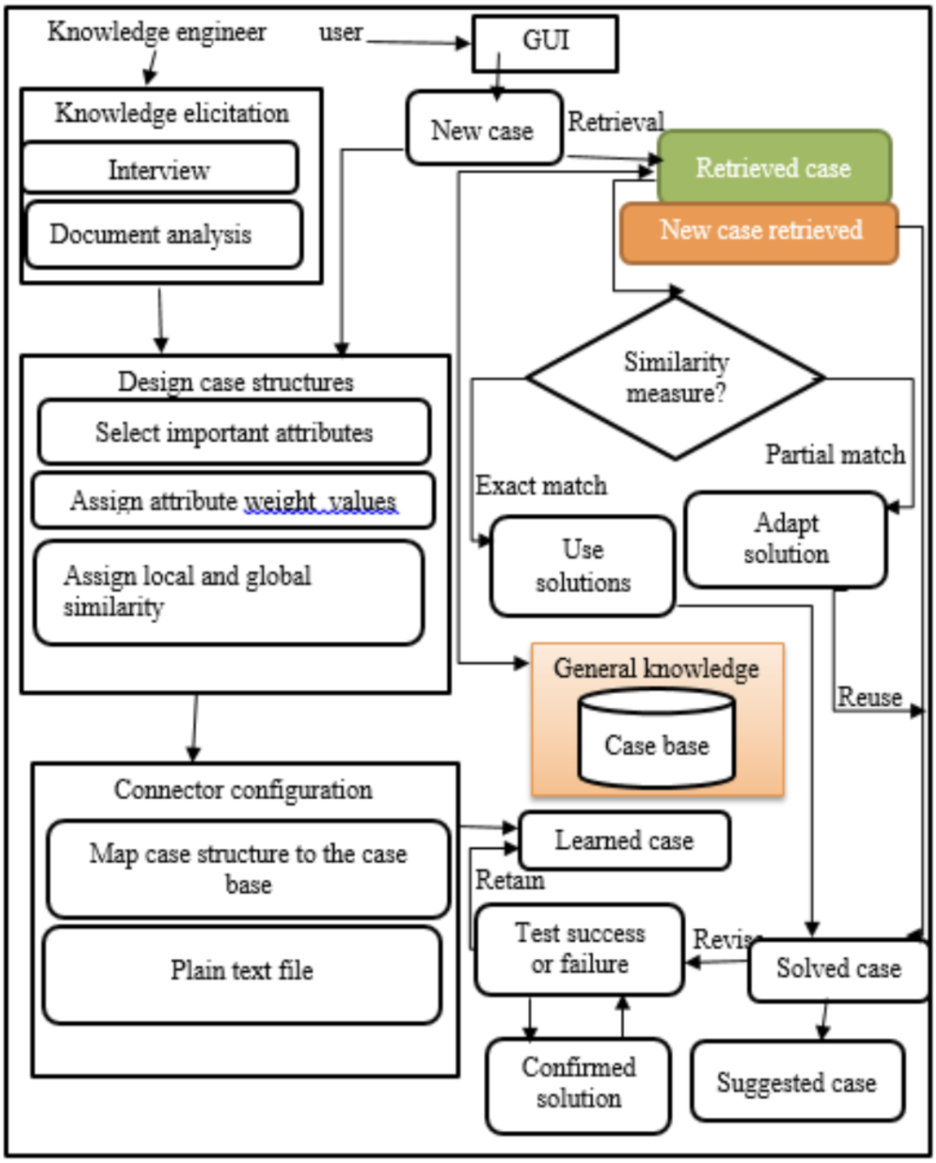

Figure 1: Architecture of CBRS for the diagnosis and treatment of obesity.

attribute structure as a tree.

5.2.2. Managing connectors in JCOLIBRI

As the name indicates that, the connector can be defined as a link between two or more things together. As shown below in Fig. 3, the connector connects the case structure and the knowledge-base or case base (datasets). The use of connectors is to make JCOLIBRI flexible against the physical storage, therefore users of the system choose the most appropriate one for

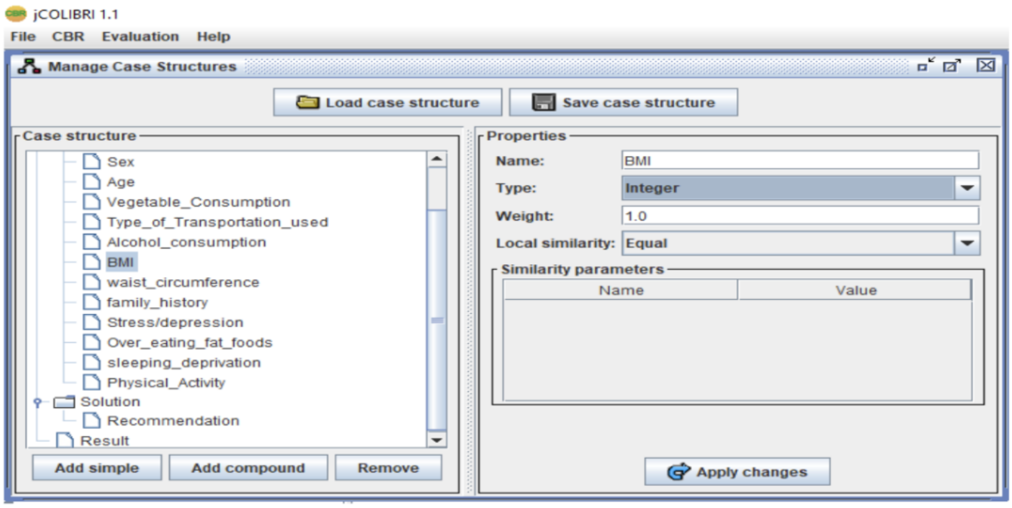

Figure 2: Snapshot from definition of the case and managing case structure.

the system when they need it. JCOLIBRI holds different connectors with different file formats like XML, plain text file, relational database file, and CSV files. For the implementation of the Obesity diagnosis and treatment model, the researcher used plain text connector. Because the plain text connector is easy to configured the case structure and other JCOLIBRI tasks.

The plain-text file case base connector is used to keep the case in the case base. The connector maps the case structure to its column from a plain text file stored in.txt format, which is then saved as an XML file, much like the case structure. The connectors link the case structure stored with the XML extension to the dataset or cases saved in plain text (.txt) format. The most fundamental tasks in connection management are specifying the appropriate case structure and file path. The case structure path is used to access and match case structure characteristics, whereas the case base.txt file is defined by the file path. A comma (,) is used as the connector's delimiter to separate the values of each attribute in the case.

5.2.3. Building case base for obesity cases

In the CBR system, cases play a great role as the source of the dataset. The researcher collects Obesity patient cases from the University of Gondar Referral Hospital. The gathered instances are used to create an Obesity diagnostic CBR system that assists physicians and other health workers by providing decision assistance. All of the obtained cases are saved in attribute-value representation format as a plain text file. We represent cases by the attribute-values representation technique because it supports the nearest neighbor retrieval algorithm and represents cases in a simple manner.

To implement the prototype of the CBR system for Obesity diagnosis and treatment, JCOLIBRI software is used. COLIBRI tool is selected because of the following essential reasons: JCOLIBRI represents cases in a very simple way, and also has different functionalities to reflect the information gained and organized. It provides a simpler process of development, based on the reuse

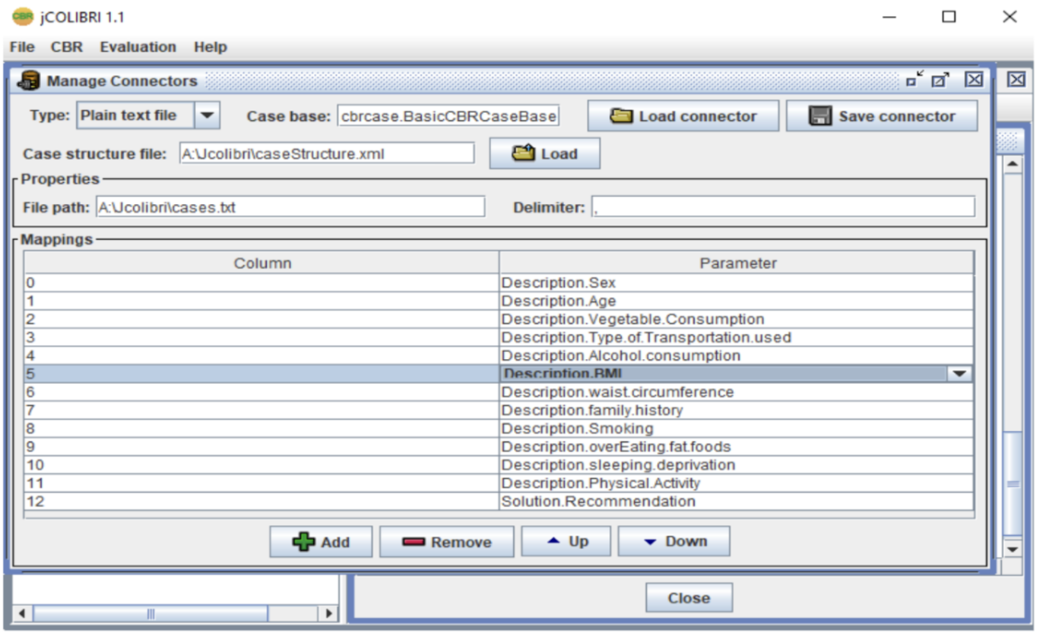

Figure 3: Snapshot of the configuration of connector in JCOLIBRI

of past designs and implementations. It supports different data types of case structure, which describes every simple event. The tool supports various knowledge representing techniques. It supports an effective facility interface with external programs and systems.

JCOLIBRI software mainly includes managing case structure, managing connectors, managing tasks, and methods of subtasks to solve the problem. The basic thing to describe those managing steps in the CBR system, first to describe the representation of cases in the case base.

- **The Case representation**
  The case can be defined as an abstract representation of a past problem and its solution. Cases in CBR have two parts. Problem description, which is the state of the world while the case is happening and what problem needed solving at the time. The second part is the Solution, which stated or derived a solution to the problem. Those collections of cases are represented in different representation techniques. Depending on the system, cases may be represented as simple plain attribute-value, textual cases, or complex hierarchical (object-oriented) structures where attributes are connected among them [10]. For this study, the researcher selected attribute-value pair representation.
- **Attribute selection** The Cases in CBR are objects of the class described by attributes. The attributes has a great role in its value and the weights of the attribute govern the importance of the attribute in the case structure. Attributes with weight zero are not included in the case base. To select case values from a case-based, the appropriate attributes or parameters must be selected. Different possible attributes are used to represent the obesity cases in the medical domain. For this work, 13 attributes are selected accordingly as presented in Table 1. The attributes are selected by analyzing

Table 1
Description of the selected case attribute in JCOLIBR.

Descriptive attributes/Features of obesity			
Attribute name	Data type	Weight	Local similarity
Sex	String	0.05	Equal
Age	Integer	0.05	Equal
Over_eating_fat_foods	Boolean	0.08	Equal
Vegetables_Consumption_Frequency	String	0.06	Max String
Stress	Boolean	0.06	Equal
Sleeping_deprivation	Boolean	0.06	Equal
Physical_Activity	Boolean	0.06	Equal
Alcohol_consumption	String	0.06	Max string
Type_of_transportation_used	Boolean	0.06	Equal
BMI	Integer	0.2	Interval
Family history	Boolean	0.06	Equal
Waist circumference	Integer	0.2	Interval

the previous case. The important attributes are used to extract a solution for the new problem from the CBR system. Since all attributes are not equally significant, assigning the appropriate weight values is necessary. The assignments of weights to each attribute indicate that attributes having higher weights are the most important and smaller weights are, the less important ones to diagnosis and treatment of obesity patients. The weight of each attribute has been assigned its value by domain experts at the time of attribute selection.

5.3. Implementing the Prototype of CBR Application

Once all the steps required in JCOLIBRI are defined and configured, the CBR application using the real Obesity patient's data is implemented and tested as shown in Fig. 4. The Configuration and the total CBR task could be saved for future use and modification. After configuring the case structure, connector, and selected tasks in each CBR cycle, the remaining task is filling the query interface to store cases in the case base as shown in Fig. 5.

After filling the values in the window interface, the system will retrieve the given query. The retrieved task starts with a problem description and ends when a best-matching previous case has been found. This data from the Battery stats helps to enhance energy estimations. The information gathered is connected to the active components' present condition, the frequency of CPU and method call invocation, and the files loaded by the power profile. The power profile may be used to calculate the current consumption for each component, as well as the approximate battery drain for each component. This will specify how many milliAmperes of current are necessary for the CPU to complete a single cycle at a set frequency for each repetition of execution for each application. These details may be obtained from any smart phone's power profile [**?**].

Retain in CBR comes after the revision step in the CBR system. When we save the revision

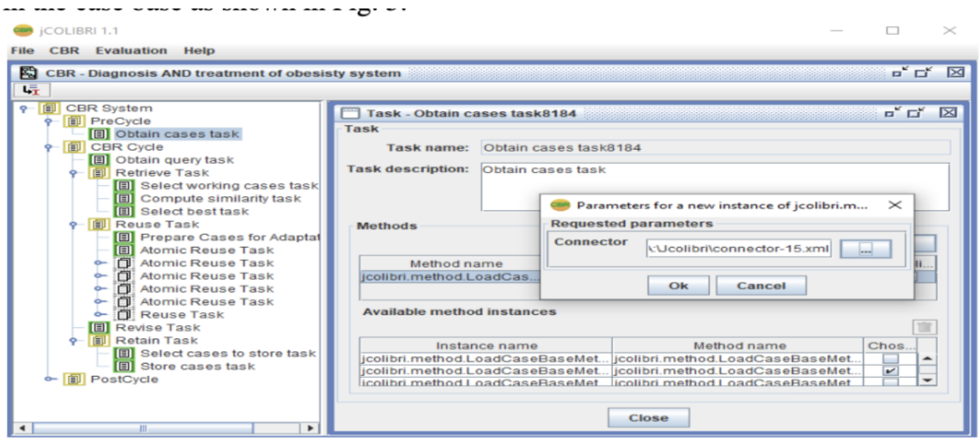

Figure 4: Snapshot of each CBR System cycle with their subtasks.

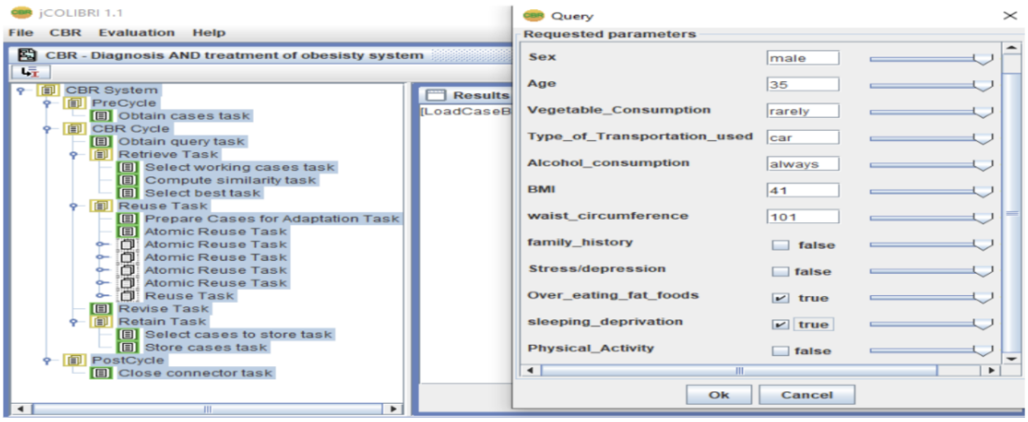

Figure 5: Snapshot of Window for case enter for case base.

cycle, retain window becomes available to store a case in the case base. The retain step in a CBR system is mainly used to store cases in a case base. The retain step also has the advantage to insert the new case in the case base. This step has the chance to add the new cases which are not found in the case base for the new problem that occurred. For example, if there were 34 cases in the case base, we have a chance to add the new case as an additional case for use in the future. This is exemplified in Fig. 7.

6. Testing and evaluation of the prototype

Following the implementation of the CBRS obesity diagnostic and treatment prototype system, each CBR system must be tested and reviewed to ensure that the CBRS prototype's performance is correct and that the prototype system is useable by end-users. Testing and evaluation of

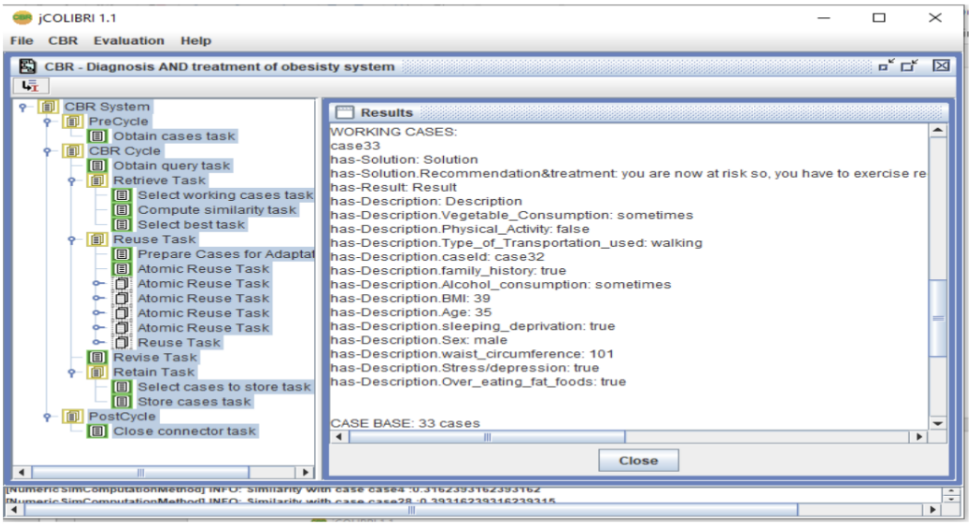

Figure 6: Snapshot for retrieved solution for the new query from the case base.

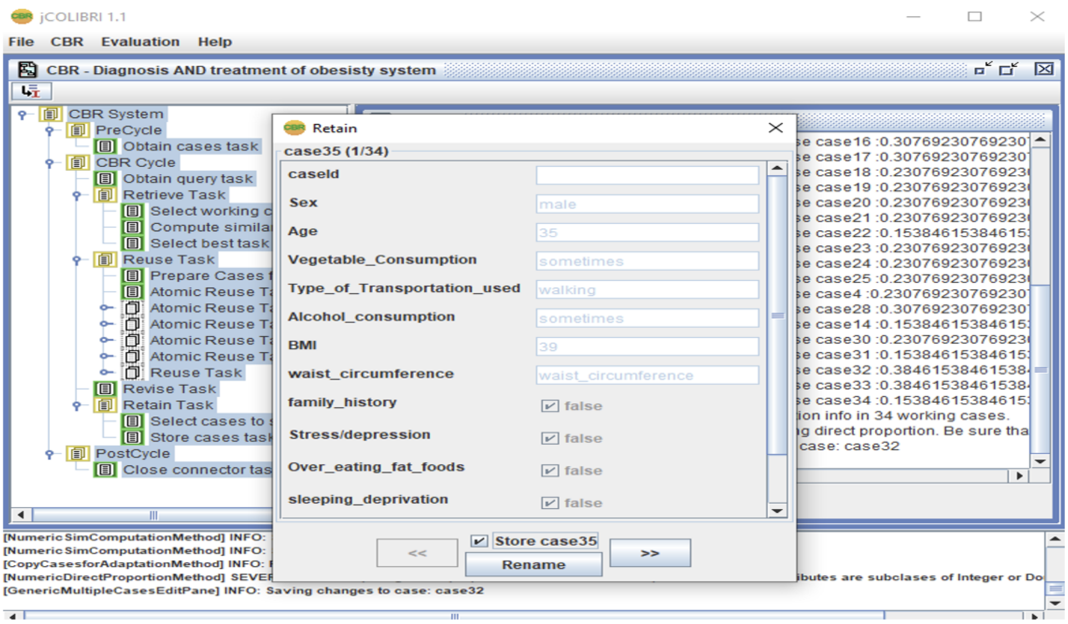

Figure 7: Snapshot for retain cycle of CBR system for obesity cases.

the prototype system responses to the request "how the KBS is diagnosed and treat the obese patient perfectly?". To solve this issue, user acceptance testing and system performance testing are employed.

6.1. User acceptance testing

The user acceptance testing measures the issues of how the system addresses the needs of the user from the user's point of view. To conduct user acceptance testing experts were used. The user acceptance testing is performed in a real situation at the University of Gondar specialized hospital for system validity.

Due to the threats of the coronavirus, doctors are so busy to give time for evaluating the system at the hospital. Therefore, the researcher has selected 2 medical doctor experts and 4 apparent students (5th year medical students) to examine the functions of the prototype by 7 sample questions. For each scale the researcher gives values in number as poor = 1, fair = 2, good = 3, very good = 4, excellent = 5. Based on the given scale, the selected domain experts give value to each attribute through the selected queries. This method of testing allows the researcher to analyze user satisfaction with the prototype system based on interpretations of the user response. The researcher uses visual interaction assessment along with close-ended questionnaires to answer the issue of user acceptance. The method of visual interaction assessment enables the domain expert to comment by interacting with the system and assessing the prototype result.

After assigning the scale value, we can calculate the user acceptance by the following general formula.

$$\mathrm{AV} = \frac{sv1 * nr}{tnr} + \frac{sv2 * nr}{tnr} + \frac{sv3 * nr}{tnr} + \cdots \tag{1}$$

To get the result of user acceptance, average performance is calculated out of 100% can be calculated as:

$$\mathrm{AVP} = \frac{AV * 100}{TS} \tag{2}$$

where AV is average, AVP is the average performances, TS is the total number of scales, i is the individual scale, nr is the number of respondents who participated in each scale value. Tnr is the total number of respondents and sv is the scale value. The AV and AVP are calculated, using the formula given above. For instance, to calculate the average and average performance the first evaluation criteria "Easy to use the system" is as follows: average is equal to scale value multiplied by the number of respondents divided by the total number of the respondent which is (4*2)/6 +(5*4)/6 = 4.6. To calculate average performance, AVP is equal to average multiplied by 100 divided by the total number of the respondent that is AVP = (4.6*100)/6 = 92. The remaining evaluation criteria's AV and AVP are calculated similarly. As indicated in the above Table 1, 33.3% of the respondents rated to evaluate "easy to use the system" as very good and the remaining 66.7% as excellent. The overall average performance for this evaluation criteria is 92%. In the same way, the "system efficient in time" evaluation parameter has the respondents rated scale is 16.7%, 33.3%, and 50% to the rank good, very good, and excellent respectively. The overall average performance for this evaluation criteria is 86%. The user interface interactivity also has the respondent rate as good, very good, and excellent is 16.7%, 50%, and 66.7% respectively. The overall average performance for this evaluation criteria is 84%. Likewise, the accuracy of the system to categorize patients correctly has the respondent rate of 33.3%, good, 33.3% very good, and 33.3%, excellent. The total average performance for this evaluation criteria is 80%. In the same manner, the "effectiveness of the system to time and cost for patient" evaluation criteria have the respondent rated 33.3% very good and 66.7% excellent. The total average performance for this evaluation criteria has been recorded at 96%. Also, the" system applicable to medical domain" evaluation criteria has a response rate of 33.3% very good, and 66.7%. The total average performance for this evaluation criteria is 92%. Lastly, "the speed of the system" has a response rate is 100% and its total average performance is also 100%.

6.2. System performance testing with cases retrieved

Precision and recall are used to evaluate the performance of the prototype system practically. In the CBR system and IR recall and precision are mainly used to measure the retrieval process of the prototype performance [10].

After identifying the valid cases from the knowledge base as shown in Table 2, the succeeding phase is calculating the values of recall and precision. In CBR, there are no standard criteria for the degree of similarity used to identify similar cases from the knowledge base. Different CBR researchers use different case similarity thresholds.

Even while several researchers proposed no fixed threshold/range, the majority of them utilized a number between one and zero-point eight, i.e., [1, 0.8]. This means that the number of instances and the case base's capacity to deliver an acceptable solution between a maximum similarity criterion of 100% and a minimum similarity threshold of 80% is between 1 and 0.8 case similarity-values [9, 10]. The following generic formula is used in the computational equation to compute precision and recall.

$$\text{Precision} = \frac{\text{number of relevant cases retrived}}{\text{total number of relevant retrived cases}} \tag{3}$$

$$\text{Precision} = \frac{\text{number of relevant cases retrived}}{\text{total number of relevant retrived cases}} \tag{4}$$

The system performance in terms of precision and recall are presented in Table 2. Table 2 shows the precision and recall value in each test case. As shown in Table 2, for each test case more than average is recorded in both recall and precision. The average recorded in terms of precision compared with the average recall is marginally lower. That is because of the swap between precision and recall. To calculate the recall and precision, for example, the first test case that is case 33 has 4 best relevant cases which are retrieved from 5 relevant cases as shown in Table 3.

So, to calculate the precision and recall, we follow the above general formula, for instance, to calculate the precision value: precision is equal to the number of relevant cases retrieved divided by the total number of cases retrieved. Based on this formula we got the precision value of 0.66 which is 66% of from 100%. The remaining test cases of their precision values are calculated by this method. Also, the recall of each test case can be calculated as precision, for example, to calculate recall value for test case 33, recall is equal to several relevant cases retrieved divided by several relevant cases in the case base so, we got the recall value of 0.8 which is 80% of from 100%.

Table 2
Description of the selected case attribute in JCOLIBR.

Test cases	Relevant cases in the case base	Relevant retrieved	Irrelevant Retrieve	Relevant not retrieved	Total cases retrieved	Precision	Recall
Case 1	5	5	1	0	6	0.83	1
Case 2	9	7	1	1	8	0.87	0.77
Case 5	10	9	2	1	11	0.81	0.9
Case 7	12	10	1	2	11	0.9	0.83
Case 12	8	6	1	1	7	0.85	0.75
Case 19	6	4	0	2	4	1	0.66
Case 23	9	7	2	1	9	0.77	0.77
Case 21	8	5	3	3	8	0.62	0.62
Case 31	7	6	1	3	7	0.85	0.85
Case 33	5	4	2	1	6	0.66	0.8
Case 40	8	6	1	2	7	0.85	0.75
Case 43	3	2	0	1	2	1	0.66
Case 60	7	7	0	0	7	1	1
Case 77	9	8	2	2	10	0.8	0.88
Average						0.71	0.80

Table 3
Comparison of system performance with previous CBR systems

Disease/Author	Used tool	System performance measurement		
		User acceptance	Precision	Recall
Proposed	JCOLIBRI	90%	71%	80%
Hypertension [13]	Python	38.2%	60%	86.1%
Triage treatment [14]	JCOLIBRI	86.4%	61%	85%
Mental health [15]	JCOLIBRI	41.6%	71%	82%
Aids [5]	JCOLIBRI	36.9%	63%	72%
Obesity [3]	Prolog	Not evaluated		

6.3. Discussion of the results

The performance of the prototype system is evaluated in different testing techniques. The first evaluation technique is user acceptance testing. The system is tested by six selected evaluators. The domain evaluators accepted the system validity with a total average performance result of 90%. This result indicates that the prototype system is more applicable in terms of system ease of use, system efficiency in time, user interface interactivity, system accuracy in categorizing patients correctly, system effectiveness in terms of time and cost for the patient, system applicability to the medical domain, and system speed. The other technique is system performance testing with case retrieval. The testing is performed in terms of recall and precision with fourteen test cases. The average system performance results for precision and recall are 71% and 80% respectively. It is critical to compare the prototype system's aforementioned performance findings with past relevant CBR systems in the medical domain area. This is seen in Table 3. Fig. 8 depicts the link between accuracy and recall.

In Table 3, all authors except [1] attempted to evaluate the performance of the prototype

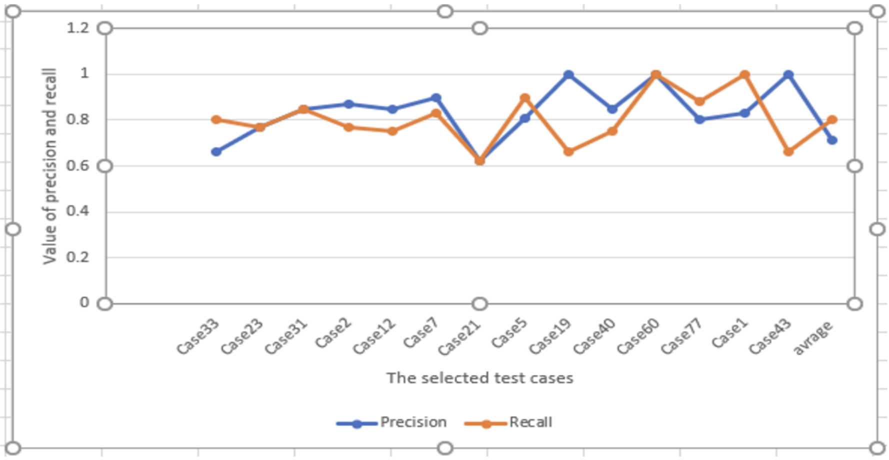

Figure 8: Relationship between precision and recall

system. This needs an improvement to increase the system performance. This work achieved 90% which is better compared to the other works in terms of user acceptance testing. Because in this study no one is evaluated the system as poor and fair, this makes a better improvement to achieve high value. Generally, the value of system performance measurement i.e. precision and recall are not equal as compared to the previous research. Because the test case and total numbers of cases used in the system are different in each previous research. In addition to the test cases and the total number of cases, the precision and recall value depends on the threshold interval taken in each research. When a larger threshold interval similarity is applied, it is feasible to get the most similar instances from the case base. This may result in lower average recall and greater average accuracy of prototype system performance. The greater the recall value, the more important things are gathered from the total number of important items recovered and not retrieved by the system.

7. Conclusion and future work

7.1. Conclusion

Diagnosis and treatment in the medical domain are common tasks. Since this task is very tedious and requires many health physicians to diagnose and treat the patients, the researcher develops a knowledge-based system (KBS) for obese patients. A KBS is a good technology in artificial intelligence to minimize human efforts and to keep the information accurate that occurred from physicians' errors. The system is developed using a CBR approach. CBR is a generic AI paradigm for reasoning from experience, and its technique has been studied in enhancing human decision-making and in designing KBSs in medicine. The main advantage of the CBR system in medicine is the automatic formation of a facility-adapted knowledge base which is a very important aspect in medical decision making.

The primary goal of this research was to create a KBS for the diagnosis and treatment of obesity. A dataset was obtained from the University of Gondar Referral Hospital using a semi-structured interview approach in order to construct a model for the system. The authors use a decision tree to model the collected knowledge. To develop the prototype system, JCOLIBRI software was used. The performance of the prototype was evaluated using user acceptance testing by six domain experts and system performance in terms of precision and recall. The prototype system concerning user acceptance evaluation achieved 90%. Also, the prototype system achieved 71% and 80% in terms of precision and recall, respectively.

7.2. Future work

Even though the results of this study are essential, there are still various problems that future researchers can solve to increase the performance of the system prototype in a real situation. This will help to assist the medical doctors by adapting the previous cases that have been retrieved from the case base. For this study, we recommend the following areas as the direction for future work.

- To achieve high performance, researcher can apply CBR with other AI techniques like rule base, neural network, fuzzy logic, and other techniques.
- In most CBRS, the knowledge acquisition method is done manually, that is either by interview or questionnaire. We hope that the other researchers can apply automatic case elicitation techniques to save time.
- All knowledge-based systems developed with the CBR approach are done in English. To some extent, this may have some impact on the users of the system. So, developing the system in the local language, like in Afan-Oromo and Amharic is will be a better work.
- For case retrieval, most CBR researchers use the two basic techniques independently. These techniques are inductive indexing and the nearest-neighbor algorithm. So, we recommended using the two basic techniques for future CBR applications.
- The developed KBS on CBR approach only supports cases in text format. So, for future work, a KBS that supports all cases in any format like image, graphics, and texts as well can be developed.

References

[1] O. E. Ndukwe, L. N. Onyejegbu, F. E. Onuodu, An improved Rule-Base expert system for diagnosis of obesity, International Journal of Engineering Research & Technology 8 (2019) 986–990.

[2] Y. Yeshaw, S. A. Kebede, A. M. Liyew, G. A. Tesema, C. D. Agegnehu, A. B. Teshale, A. Z. Alem, Determinants of overweight/obesity among reproductive age group women in ethiopia: multilevel analysis of ethiopian demographic and health survey, BMJ Open 10 (2020).

[3] H. E. Short, Adult Obesity Management : Knowledge , Attitudes , and Practices of Alberta Family Physicians by Hilary Erin Short A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science Health Policy Research School of Public He, 2014.

[4] S. E. Cuda, M. Censani, Pediatric obesity algorithm: A practical approach to obesity diagnosis and management, Front. Pediatr. 6 (2019).

[5] S. Mohammed, Amharic based knowledge-based system for diagnosis and treatment of chronic kidney disease using machine learning, Int. J. Adv. Comput. Sci. Appl. 9 (2018).

[6] I. E. Simona, C. Alexandra, J. Gabriela, Obesity treatment strategies, Acta Medica Marisiensis 61 (2015) 361–366.

[7] D. Richard, Overcoming obesity : An initial economic analysis discussion paper, McKinsey Glob. Inst (2014) 1–71.

[8] A. Abebaw, Application of Case-Based Reasoning in Legal Case Management: An Experiment with Ethiopian Labor Law Cases, Ph.D. thesis, Addis Ababa University, 2014.

[9] T. A. Assegie, A. O. Salau, C. O. Omeje, S. L. Braide, Multivariate sample similarity measure for feature selection with a resemblance model, Int. J. Electr. Comput. Eng. (IJECE) 13 (2023) 3359.

[10] A. O. Salau, S. Jain, Feature extraction: A survey of the types, techniques, applications, in: 2019 International Conference on Signal Processing and Communication (ICSC), 2019, pp. 158–164. doi:10.1109/ICSC45622.2019.8938371.

[11] P. Agrawal, V. Madaan, V. Kumar, Fuzzy rule-based medical expert system to identify the disorders of eyes, ent and liver, International Journal of Advanced Intelligence Paradigms 7 (2015) 352–367.

[12] D. Sethi, P. Agrawal, V. Madaan, S. K. Singh, X-gyno: fuzzy method based medical expert system for gynaecology, Indian Journal of Science and Technology (2016).

[13] R. Kaur, V. Madaan, P. Agrawal, Fuzzy expert system to calculate the strength/immunity of a human body, Indian Journal of Science and Technology (2016).

[14] K. Ranjit, M. Vishu, A. Prateek, S. S. Kumar, K. Amandeep, Fuzzy expert system for identifying the physical constituents of a human body, Indian Journal of Science and Technology (2016).

[15] D. Sethi, P. Agrawal, V. Madaan, X-tumour: fuzzy rule based medical expert system to detect tumours in gynecology, International Journal of Control Theory and Applications 9 (2016) 5073–5084.

[16] S. Dagne, Y. A. Gelaw, Z. Abebe, M. M. Wassie, Factors associated with overweight and obesity among adults in northeast ethiopia: a cross-sectional study, Diabetes, metabolic syndrome and obesity: targets and therapy (2019) 391–399.