Thesis for the Degree of Doctor of Philosophy

AN ENSEMBLE-BASED FEATURE SELECTION METHODOLOGY FOR CASE-BASED LEARNING

Maqbool Ali

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Republic of Korea

August 2018

AN ENSEMBLE-BASED FEATURE SELECTION METHODOLOGY FOR CASE-BASED LEARNING

Maqbool Ali

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Republic of Korea

August 2018

AN ENSEMBLE-BASED FEATURE SELECTION METHODOLOGY FOR CASE-BASED LEARNING

by

Maqbool Ali

Supervised by

Prof. Sungyoung Lee

Submitted to the Department of Computer Science and Engineering and the Faculty of Graduate School of Kyung Hee University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Dissertation Committee:

Prof. Seung-Kyu Lee

Prof. Sung-Ho Bae

Prof. Chang-Ho Jihn

Prof. Seok-Won Lee

Prof. Sungyoung Lee

I dedicated my thesis to my beloved parents and family, who play a pivotal role by encouraging and supporting in all hard times to keep me in the position to complete my PhD degree.

Abstract

Case-based learning (CBL) approach has been receiving a lot of attention in medical education, as an alternative to the traditional learning environment. This student-centric teaching methodology, exposes the medical students to real-world scenarios, where they can then utilize strong clinical reasoning skills in a non-obtrusive and scalable way, along with any existing theoretical knowledge, and experience, to resolve a large number of ever-evolving, complex problems. However, this activity takes its toll on the medical students, who then tend to choose computer-based cases as opposed to lectures for their learning. In order to support the learning outcomes of students, a plethora of web-based learning systems have been developed; however, these systems do not provide computer-based as well as experiential knowledge-based support for CBL practice. Medical literature contains a lot of useful knowledge in textual form, which can be used as a very beneficial source for the computer- based CBL practice. Therefore, designing and developing an efficient, automated case-based learning approach, which utilizes the strength of both humans (experiential knowledge) and computers (domain knowledge) is a major problem.

In order, to solve this problem, the text mining domain provides the basic framework for constructing domain knowledge, which includes text preprocessing, text transformation, feature selection, term extraction, relation extraction, and model construction tasks. Amongst these tasks, feature selection is considered to be one of the most critical problems, whereby from a large set of features, only the appropriate features have to be selected. Feature selection techniques, which solve this problem, are generally split into three categories: filter-based, wrapper-based, and hybrid approaches.

In the filter-based feature selection technique, features are first ranked and then filtered, based on a threshold value. While a lot of methods are available for performing the feature ranking task,

e.g. Information Gain, Chi Square, Gain Ratio, and Symmetrical Uncertainty; there is no one comprehensive solution, since each method suffers from some limitations.

Similarly, a feature ranking task is also important as it requires an optimal cut-off value to select important features from a list of candidate features.

Keeping in view all above-mentioned facts and to support of students' learning systems, this research, provides contribution, in the following areas:

(1) Feature Ranking; where we propose, an innovative unified features scoring (UFS) algorithm to evaluate the feature-set in a comprehensive manner to generate a final ranked list of features, which ranks the features with-out using any learning algorithm, has low computational cost, and does not suffer from any individual statistical biases.

(2) Feature Selection; where we propose, an innovative threshold value selection (TVS) algorithm to define a cut-off point for removing irrelevant features, irrespective of the characteristics of the dataset and selecting a subset of features that are deemed important for the domain knowledge construction, and;

(3) CBL Platform; where we designed and developed, an interactive case-based learning system (iCBLS) to integrate experiential knowledge and domain knowledge. The iCBLS enables medical teachers to create real-world CBL cases for their students with the support of their experiential knowledge and computer-generated trends, review the students' solutions, and give feedback and opinions to their students. The developed system also facilitates medical students in preparation by providing a machine-generated domain knowledge support, which can be utilized before attending the actual CBL class.

Throughout this thesis, we perform both quantitative and qualitative evaluation of our proposed (1) methodology on benchmark datasets, and (2) CBL approach. The extensive experimental results show that our approach provides competitive accuracy and achieved (1) on average, around 7% increase in f-measure as compared to the baseline approach, (2) on average, around 5% increase in predictive accuracy as compared to state- of-the-art methods, and (3) a high success rate of 70% for students' interaction, 76.4% for group learning, 72.8% for solo learning, and 74.6% for improved clinical skills.

Acknowledgement

Alhumdolillah, By grace of ALLAH Almighty, who is the most beneficent and merciful. Who gave me the strength, courage, patience during my doctoral study and showering HIS blessings upon me and my family.

I am highly grateful to my advisors Prof. Sungyoung Lee from Kyung Hee University (KHU), South Korea, and Prof. Byeong Ho Kang from University of Tasmania (UTAS), Australia for their boundless moral and technical supervision, guidance, and courage in coping with the difficult challenges throughout the education period of my doctoral studies. They trained me in multi-directions to face the challenges of practical life in a professional manner. Their lively natures, clear assistance, and direction enabled me able to complete my thesis. They have refined the key ingredients for high quality research, namely my skills of creativity, thinking, and technical understanding. Moreover, I would like to acknowledge their valuable guidance and support to refine the problem statement as well as that to streamline my research direction.

I appreciate my dissertation evaluation committee for their valuable observations and insight recommendations during the dissertation defense. These comments enhanced the presentation and contents of the dissertation.

I am also grateful to my KHU fellows who have supported me with their precious time, technical expertise, and brotherly encouragement during my study period. They were always ready to direct me in tough situations during my studies. I would like to thank Dr. Wajahat Ali Khan, Dr. Thien Huynh The, Dr. Bilal Amin, Jamil Hussain, Taqdir Ali, Shujaat Hussain, Hafiz Syed Muhammad Bilal, Muhammad Asif Razzaq, Syed Imran Ali, Usman Akhtar, Musarrat Hussain, Ubaid Ur Rehman, Fahad Ahmed Satti, Muhammad Sadiq, and Anees Ul Hassan. They supported me in successfully performing various personal and academic tasks that presented hurdles me dur-

ing my stay at South Korea. I would like to thank Mrs. Kim for being available and helping me in many regards.

I also appreciated to all of my current and former UCLab fellows for their kind support to my personal and academic life at Korea. I am highly gratified to the following excellent researchers - Dr. Rehman Ali, Dr. Maqbool Hussain, Dr. Muhammad Afzal, Tae Ho Hur, Dohyeong Kim, Jaehun Bang, Hua-Cam Hao, Asim Abbas, Muhammad Zaki Ansaar, Dr. Kifayat Ullah, Dr. Waqas Nawaz, Dr. Ahsan Raza Kazmi, Saeed Ullah, Waseem ul Hassan, Dildar Hussain, Abdul Sattar, Zain Abbas, Ahmad Jan, and Aftab Alam. This journey would have been quite difficult without their support. They helped my personal and academic life to boost myself. I also appreciate all my Korean and international friends who worked collaboratively with me and inculcated team work skills in me.

I would like to thank Dr. Soyeon Caren Han, Matthew Jee Yun Kang, Mrs. Kerrin McKeown, David Herbert, and Leandro Disiuta for their kind support during my stay at UTAS. I would like to thank Mrs. Kerrin McKeown also for reviewing and editing the English in this thesis.

I am thankful to my MS advisor Dr. Ali Mustafa Qamar in SEECS, NUST, Islamabad for his guidance and continuous support during my study in Korea. Last but not the least, I would like to express my sincere gratitude to my close friends especially Dr. Kashif Sattar, Dr. Aftab Ahmad, and Muhammad Junaid.

Maqbool Ali

August, 2018

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The main focus of this dissertation is on investigating the dynamics of case-based learning (CBL), leading to a proposal for an interactive medical learning approach to prepare medical students using real-world CBL case(s) for better clinical practice outside the class. For interactive and effective learning purposes, this dissertation includes a methodology to construct the domain knowledge (i.e. structured declarative knowledge) from unstructured text to facilitate and provide domain knowledge to medical students for solving the real-world clinical case(s) during CBL practice. For the domain knowledge construction, the feature selection task is considered to be one of the most critical problems in a text mining domain. This thesis proposed an efficient and comprehensive feature selection methodology for selecting appropriate features from a larger set of features. The opening chapter will contain the main motivations for this process in Section 1.1, the problem statement along with research questions in Section 1.2, key contributions of this research in Section 1.3, and finally, the summary of dissertation is outlined in Section 1.4.

1.1 Motivation

Medical education is an active area of research and has seen tremendous revolutionary measures in the past few decades. The main purpose of these educational programs is to: (1) develop educational leaders, (2) change the learners' knowledge, skills, or attitudes, and (3) improve the educational structures [20]. Various teaching methodologies have been introduced in professional health education [21], with active learning gaining a lot of attention around the world [22]. In active learning, instructions are given to students to actively engage them [23]. Case-Based Learning (CBL) is one of the active learning approaches, which provides favorable circumstances to students in order to explore, question, discuss and share their experiential knowledge for improving their

practical intelligence [22]. CBL is not a new term, from its introduction in the medical domain since 1912 [24]. It has proceeded in many forms, from simple hands-on, in-class exercises to semester long projects and/or case studies [25], CBL, has maintained its focus around clinical, communal, and scientific problems.

In terms of student-centric pedagogy, CBL is being widely used in various health-care training environments around the world [26–33]. In particular, this approach has been met with general acceptance in the fields of medicine, dentistry, pharmacology, occupational and physical therapy, nursing, allied health fields, and child development. Similarly, it is being used in clinical as well as non-clinical courses such as nursing courses, adult health, mental health, pediatric, and obstetrical nursing courses, pathophysiology, statistics, law, school affairs, physics education, and research [22, 34, 35]. In addition, this approach has been utilized in various departments such as medical education, information technology, and quality improvement [24], and has also been practiced in rural as well as underserved areas [24]. These findings validate the effectiveness and universal nature of CBL, which is especially useful for the curricula of medical and health professions [24].

In CBL practice, the clinical case is a key component in learning activities, which includes basic, social, and clinical studies of the patient [36]. In the medical domain, this component provides the foundation to understand the particulars of a disease. Recent trends have emphasized the use of real-life clinical case(s) for providing this much needed practice for the medical students [37–39]. These cases enable the students to use their experiential knowledge to interpret them easily [22]. In medical area, CBL facilitates students in learning the diagnosis and management of clinical cases [24], and prepares the participants to practice primary care and critical situations [40]. The CBL approach promotes learning outcomes and builds confidence in students, enabling them to practice real-life decisions [30,41]. According to Thistlethwaite [36], "CBL promotes learning through the application of knowledge to clinical cases by students, enhancing the relevance of their learning and promoting their understanding of concepts". CBL is also known to be an effective learning approach for small groups of medical students at undergraduate, graduate, postgraduate education levels as well as for professional development [24, 36, 37, 42, 43].

Besides the benefits of CBL approach, there are also a few shortcomings of this approach. For

example, in professional education for health and social care domains, students feel that classroom CBL activities require a significant amount of time [44]. Sometimes, students feel uncomfortable while participating in group learning activities and they prefer to work alone [45]. Normally, formal learning activities are performed without a real patient case [36], where interactions are often unplanned and rely on the goodwill of patients. In specialized literature, medical education programs are considered to be complex due to their diverse interactions amongst participants and environments [20]. Discussion-based learning in a small group, like CBL, is considered to be a complex system [46]. In small-groups, multiple medical students are interacting and exchanging information with each other, where each student is also a complex system [47]. In health care professional education, students have to tackle uncertain situations due to the interplay of a number of problems [48]. In such situations, each student has his/her own judgment, opinion, and feedback and will consider this integral as well as appropriate for that situation. In such situations, an experiential knowledge (EK) is thought-out as a resource [48] which can facilitate and provide lived knowledge to students. According to Willoughby [49], "Experiential knowledge is a knowledge of particular things gained by perception and experience". Experiential knowledge enables individuals to capture practical experience for problem solving. It is considered as a valuable resource to enhance an individual's participation and user empowerment [48].

For problem-based learning, humans and computers can play a key role in the medical domain. However, both have their own strengths and weaknesses [50, 51]. For example, In terms of their strengths, (1) Human judgment is considered as credible, (2) Humans have common sense and can determine new rules, off the shelf, (3) Humans can easily identify trends or abnormality in visualization data. However, Humans also suffer from severe weaknesses whereby they (1) cannot often accomplish complex computational decisions, (2) cannot perform fast reasoning computations, and (3) get easily tired and bored. These human weaknesses can be mitigated by using a computer, which can perform complex computation decisions relatively faster and will not suffer from tiredness or boredom.

Being a human, students are easily tired or bored, and tend to choose computer-based cases [36, 52] and opt for web-based cases as compared to lectures for their learning [53, 54]. Additionally, more attention is given to online/web-based learning environments [36]. In order to support

the learning outcomes of students, a plethora of web-based learning systems have been developed [55–64]. A review of the literature shows that these systems either do not support computer-based interactive case authoring as well as its formulation, or without the support of acquiring real-world CBL cases or do not provide feedback to students. Currently, much less attention is given to the development mechanisms of real-world clinical cases using experiential knowledge and no support of domain knowledge while formulating the case. Case formulation means identification of a medical chart's components (demographics, chief complaint, medical history, habits, family history, medicines, allergies, diagnosis, treatment, and recommendations) from a given clinical case and then writing personal observations for each component.

There exists plenty of textual data in the medical domain, which can be useful for medical education, especially for CBL purposes. This data is available in a variety of formats and with different semantics. This overwhelming data provides various opportunities to gain useful knowledge that reflects the depth of information that plays an important role in decision-making. Declarative knowledge (also called factual knowledge) is a type of knowledge expressed in the form of unstructured text, which can play an important role in health's education, decision support, and wellness applications after structured transformation. According to the Simply Philosophy study [65], "Factual knowledge is a justified affirmation of something". It combines the concepts to make an affirmation of something. For example, "Blood_disease" and "is a symptom" make an affirmation "Blood_disease is a symptom". The produced affirmation is either true or false; however, in declarative knowledge it is always true. Handling unstructured contents is the foundation to construct the domain knowledge (structured declarative knowledge) required for interactive learning, to prepare medical students for their clinical practice before and outside the class.

Text mining is the process of deriving high-quality information from an unstructured text. It involves the application of techniques from areas like information retrieval, natural language processing, information extraction, and data mining [66]. In the text mining domain, normally text preprocessing, text transformation, feature selection, term extraction, relation extraction, and model construction tasks are involved to construct domain knowledge from textual data. For constructing reliable domain knowledge, the feature selection (FS) task is considered one of the most critical problems for selecting appropriate features from a larger set of features [67–69].

Feature selection performs a key role in the (so-called) process of 'Knowledge Discovery' [69]. Traditionally, this task is performed manually by a human expert; thereby making it more expensive and time-consuming, as opposed to an automatic FS which has become necessary for the fast-paced digital world of today [13]. Feature selection techniques are generally split into three categories: filters, wrappers, and hybrid, where each technique has capabilities and limitations [12–14]. Popular evaluation methods used for these techniques are *information-theoretic measures*, *co-relational measures*, *consistency measures*, *distance-based measures* and *classification/predictive accuracy*. A good feature selection algorithm can effectively filter out unimportant features [70]. In this regard, a significant amount of research has focused on proposing improved feature selection algorithms [71–75]; consequently, most of these algorithms use one or more of the aforementioned methods for performing feature selection. However, there is a lack of a comprehensive framework, which can select features from a given feature set.

1.2 Problem Statement

For an automated CBL, a structured knowledge construction from textual data is a challenging task [76]. In the text mining domain, normally text preprocessing, text transformation, feature selection, term extraction, relation extraction, and model construction tasks are involved, where the feature selection task is considered to be one of the most critical problems for selecting appropriate features from a larger set of features [67–69]. To design an effective CBL approach for better clinical competency, three major research questions must be answered:

1. How to rank the features without using any learning algorithm, high computational cost, and individual statistical biases of state-of-the-art feature ranking methods? In this case, the filter-based feature ranking approach is more suitable than the other two approaches (wrapper, hybrid). Filter-based methods evaluate a feature's relevance without using any learning algorithm [12, 67]. Filter-based feature ranking methods are further split into two subcategories: univariate and multivariate. Univariate filter methods are simple and have high performance characteristics as compared to other approaches [77]. Even though the univariate filter-based methods are considered to be much faster and less computationally

expensive than wrapper methods [12, 15]; each method has its capabilities as well as its limitations. For example, Information Gain (IG) is a widely acceptable measure for ranking the features [78]; however, IG is biased towards choosing features with a large number of values [17]. Similarly, Chi Square (CS) determines the association between a feature and its target concept/class; however, CS is sensitive to sample size [17]. In addition, Gain Ratio and Symmetrical Uncertainty enhances the information gain; however, both are biased towards features with fewer values [79]. Therefore, designing an efficient feature ranking approach and overcoming the aforementioned limitations is our first target.

2. How to find a minimum threshold value for retaining important features irrespective of the characteristics of the dataset? In this case, for defining cut off points for removing irrelevant features, a separated validation set and artificially generated features approaches are used [72]; however, it is not clear how to find the threshold for the features' ranking [18, 80]. Finding an optimal cut-off value to select important features from different datasets is problematic [80]. Therefore, designing an empirical method to specify a minimum threshold value for retaining important features and overcoming the aforementioned limitations is our second target.

3. How to fill the gaps between human-based and computer-based learning to innovate the CBL approach for better clinical proficiency? Both humans and computers have their own strengths and weaknesses [50, 51]. In the medical area, human (domain expert) judgment is considered as more credible than a computer; however, a human cannot perform fast reasoning computations to work for extended periods and will get tired and feel bored. A computer has the advantage over a human of being able to perform fast reasoning computation without feeling bored. Being a human, students feel that classroom CBL activities require a significant amount of time; they get tired [44], and tend to choose computer-based cases [36, 52]. Similarly, students opt for web-based cases as compared to lectures for their learning [53, 54]. Additionally, more attention is given to online/web-based learning environments [36]. In order to support the learning outcomes of students, a plethora of web-based learning systems have been developed [55–64]. A review of the literature shows that these systems either do not support computer-based interactive case authoring as well as its

formulation, or without the support of acquiring real-world CBL cases, or do not provide feedback to students. Currently, much less attention is given to fill the gaps between human-based and computer-based learning. Therefore, designing and developing an interactive and effective case-based learning approach to utilize the strength of both human (experiential knowledge) and computer (domain knowledge) and overcoming the aforementioned limitations is our third target.

1.3 Key Contributions

We summarize the main contributions of this thesis as below:

1.3.1 Novel feature ranking algorithm

For evaluating the feature-set in a comprehensive manner to generate a final ranked list of features, a *unified features scoring* (UFS) algorithm is introduced, which ranks the features without using any learning algorithm, without high computational cost, and without any of the individual statistical biases of state-of-the-art feature ranking methods.

1.3.2 Novel threshold value selection algorithm

For defining the cut-off point for removing irrelevant features, a *threshold value selection* (TVS) algorithm is introduced, which selects a subset of features that are deemed important for the domain knowledge construction. TVS finds a minimum threshold value for retaining important features irrespective of the characteristics of the dataset.

1.3.3 Improved feature selection

Proof-of-concept for the UFS and TVS techniques, after performing extensive experimentation which achieved (1) on average, a 7% increase in f-measure as compared to the baseline approach, and (2) on average, a 5% increase in predictive accuracy as compared to state-of-the-art methods.

1.3.4 Reliable domain knowledge construction

For interactive and effective learning purposes, this research includes a methodology to construct the domain knowledge (i.e. structured declarative knowledge) from unstructured text, to facilitate and provide computer-based domain knowledge to medical students for solving real-world clinical cases during CBL practice. With the evolution of knowledge stored in a database, the proposed system can hold better clinical competence and can provide intensive learning in the future. For effective transformation, controlled natural language is used, which constructs syntactically correct and unambiguous computer-processable texts.

1.3.5 Semi-automatic real-world clinical case creation technique

In professional education for health and social care domains, the clinical case is a key component in learning activities and provides a foundation to understand the nature of a disease. To innovate the case-based learning approach for better clinical proficiency, a semi-automatic technique for real-world clinical case creation is introduced. The proposed technique facilitates health care professionals (medical teachers) who are interconnected in common practice, to produce experiential knowledge for the purpose of developing clinical knowledge. This knowledge includes scientific knowledge and realistic experiences to provide responses in risky and uncertain situations.

1.3.6 An interactive and effective automated CBL system development

For an interactive as well as an effective case-based learning (CBL) approach, an *interactive case-based learning system* (iCBLS) is designed and developed, which utilizes the strength of both human (experiential knowledge) and computer (domain knowledge). The iCBLS enables medical teachers to create real-world CBL cases for their students with the support of their experiential knowledge and computer-generated trends, review students' solutions, and give feedback and opinions to their students. It also facilitates medical students to do CBL rehearsal with a machine-generated domain knowledge support before attending an actual CBL class.

1.4 Thesis Organization

The dissertation aims to investigate an efficient feature selection methodology to construct reliable domain knowledge for case-based learning. Figure 1.1 shows the dissertation overview, and summarizes the structure and flow of the dissertation.

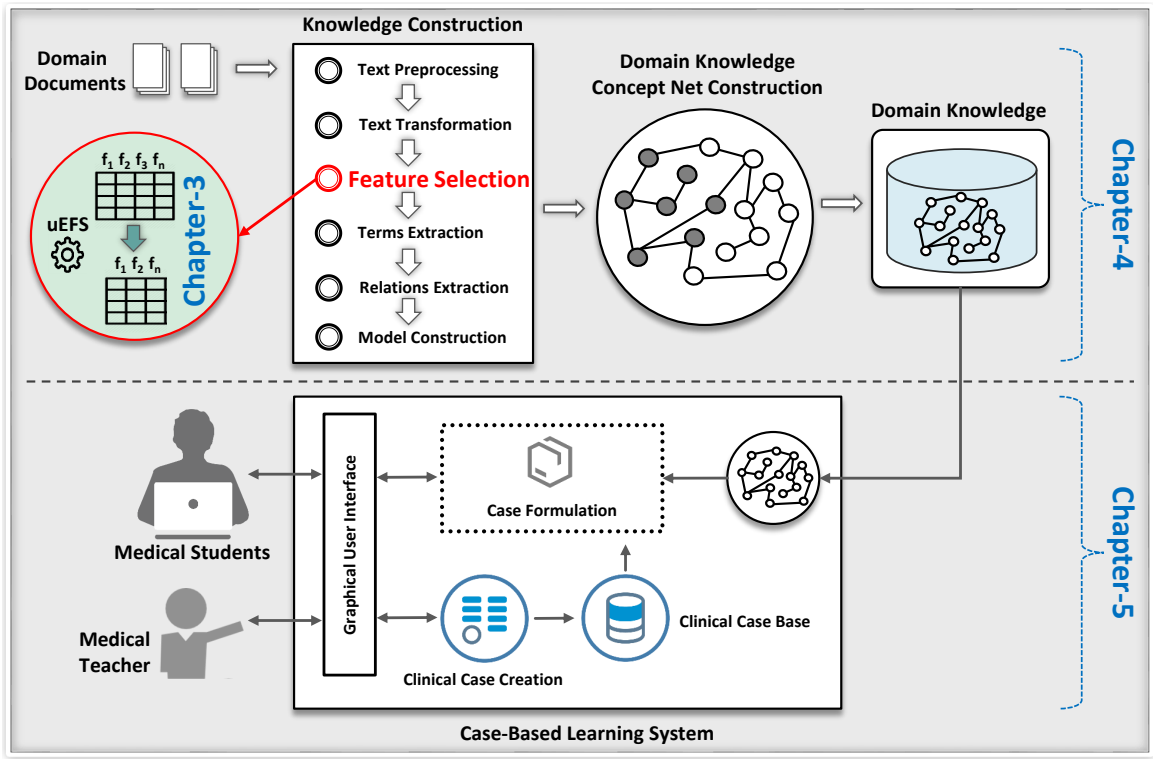


Figure 1.1: Idea diagram of the proposed research studies with chapters mapping.

This dissertation is organized into chapters as following.

- **Chapter 1: Introduction**. Chapter 1 provides the introduction of the research work for feature selection to construct domain knowledge for an interactive and effective case-based learning. It focuses on the problems in areas, the goals to achieve these problems, the objectives achieved in this research work, and finally the dissertation overview.

- **Chapter 2: Related work**. Chapter 2 reviews previous research for feature selection methodologies to filter out irrelevant features. This research focuses on presenting a comprehensive and flexible feature selection methodology based on an ensemble of univariate filter

measures for constructing a reliable domain knowledge to innovate the case-based learning approach. Therefore, we present an overview of different methodological studies of feature selection as well as case-based learning approaches. Various research directions related to (1) feature selections like features ranking and ensemble approaches, (2) technologies used for the domain knowledge construction, and (3) case-based learning methodologies and related web-based learning systems are discussed in each subsection. Finally, we summarize the related works that utilize feature selection, knowledge construction, and case-based learning methodologies.

- **Chapter 3: Univariate ensemble-based feature selection**. In this chapter, we present *univariate ensemble-based feature selection* (uEFS) methodology to select informative features from a given dataset. For the uEFS methodology, we first propose a *unified features scoring* (UFS) algorithm to generate a final ranked list of features after a comprehensive evaluation of a feature set. For defining a cut-off point to remove irrelevant features, we then propose a *threshold value selection* (TVS) algorithm to select a subset of features, which are deemed important for the domain knowledge construction. To evaluate the proposed uEFS methodology, we have performed two studies. Finally, for each study, we present the experiment setup, and then provide the corresponding experimental results for each study under different settings.

- **Chapter 4: Domain knowledge construction**. This chapter describes a methodology to construct the machine-generated domain knowledge (i.e. structured declarative knowledge) from an unstructured text. The proposed methodology constructs an ontology from unstructured textual resources in a systematic and automatic way using artificial intelligence techniques with minimum intervention of a knowledge engineer.

- **Chapter 5: Case-based learning**. This chapter presents an interactive and effective case-based learning approach for medical education, which utilizes the strength of both human (experiential knowledge) and computer (domain knowledge). In this chapter, we introduce (1) a semi-automatic technique for real-world clinical case creation, (2) case formulation technique with domain knowledge support, and (3) an IoT-based platform for supporting

flipped case-based learning. To automate the proposed CBL approach, we design and develop an *interactive case-based learning system* (iCBLS). To evaluate the proposed approach, we have performed two studies. Finally, for each study, we present the evaluation setup and then provide the corresponding evaluation results for each study under different settings.

- **Chapter 6: Conclusion and future directions**. This chapter concludes the thesis and provides future directions in this research area. It also describes the potential applications of the proposed methodology.

Chapter 2

Related Work

This chapter describes various existing studies related to each aspect of this research work. This research focuses on presenting a comprehensive and flexible feature selection methodology based on an ensemble of univariate filter measures for constructing a reliable domain knowledge, to innovate the case-based learning approach. Therefore, this section is split into three subsections to present an overview of different methodological studies of feature selection, domain knowledge construction, and case-based learning approaches. Various research directions related to (1) feature selections such as features ranking and ensemble approaches, (2) technologies used for domain knowledge construction, and (3) case-based learning methodologies and related web-based learning systems, are discussed in each subsection. Finally, we summarize the related works that utilize feature selection, knowledge construction, and case-based learning methodologies.

2.1 Overview of feature selection

This study includes a univariate ensemble-based feature selection (uEFS) methodology for selecting salient features from a dataset. This methodology is based on an empirical study of different univariate filter-based feature selection measures such as including information gain, gain ratio etc. The following are some relevant feature selection (FS) studies from a methodological point of view, which contain:

- basic concepts and procedures of feature selection
- state-of-the-art feature selection approaches, and
- research surveys, comparative studies, and frameworks in the domain of FS

FS is an approach that chooses a subset of features from a given list of original features and filters the irrelevant features to speed up the processing of a machine learning algorithm for improving mining performance (predictive accuracy, result comprehensibility). Feature selection is an active area of research and has undergone significant revolution in the past few decades. Various research disciplines such as pattern recognition, machine learning, data mining, and text mining have applied FS techniques to many fields such as text categorization, image retrieval, customer relationship management, and intrusion detection [1]. The FS task is considered to be one of the most critical problems for selecting appropriate features from a larger set of features [67]. This approach becomes expensive and intractable (NP-hard), when the number of features N increases. It performs a key role in the so-called process of 'Knowledge Discovery' [69]. The FS task can also be performed manually by a human expert; however, in this case it is considered as an expensive and time-consuming task. In such cases, an automatic FS is necessary [13].

A review of applied FS methods for microarray datasets was performed by Bolón et al. [81]. Microarray data classification is a difficult task due to its high dimension and small sample sizes. Therefore, feature selection is considered the de-facto standard in this area [81]. Normally, a FS approach consists of four basic steps, namely, 'subset generation', 'subset evaluation', 'stopping criterion', and 'result validation' [82] as shown in Figure 2.1, which are described as follows.

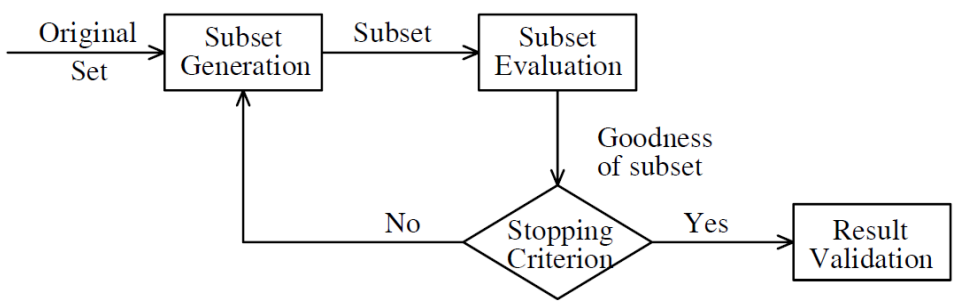


Figure 2.1: Basic steps of feature selection [1].

- *Subset generation* is a searching process, which is based on a specific approach to evaluate a candidate subset. For this process, two basic criterion are defined. The first one is to decide the starting point of the search and the second one is about the search strategy. For the first criteria, the search can be started either from an empty set, or from a full set, or from both

ends, or at random. Similarly, for the second criteria, the search strategy can be sequential, complete, or a random search.

- *Subset evaluation* is the second step for the feature selection procedure. In this step, each candidate subset, which is generated from the previous step, is compared against the previous best subset based on a certain evaluation criterion. In the case of better results, the new subset replaces the previous one, as it is considered the best subset. The goodness of a subset is evaluated either by an independent criterion (without involvement of mining algorithm such as filter method) or by a dependent criterion (reliant on mining algorithm such as wrapper and hybrid methods). For independent criteria, information-theoretic measures, correlational or dependency-based measures, consistency-based measures, and distance-based measures are widely used in literature [1]. Most of the feature selection algorithms use one or more of the aforementioned measures for performing feature selection.

- *Stopping criterion* is the third step, in which the procedure of feature selection is stopped due to some stopping criteria. Following are some definitions of stopping criteria, which are: (1) when the search is complete, (2) when a specific number(limit) is reached, (3) when addition or deletion of features are not improving the result, and lastly, (4) when the error rate is reduced for the given task [1].

- *Result validation* is the final step, where the selected subset is validated either by beforehand knowledge or by observing the change of mining performance using synthetic or real-world data sets [1].

A research taxonomy of feature selection approaches is shown in Figure 2.2; the components represented with bold text and highlighted background are covered in this study. This figure shows an abstract view of taxonomy for feature ranking methods.

Feature selection approaches are generally split into three categories: filter, wrapper, and hybrid as shown in Figure 2.2, where each approach has capabilities and limitations as shown in Table 2.1.

Liu and Yu [1] proposed a categorizing framework to build an integrated system for automatic feature selection. This framework was based on a unifying platform and laid the important foun-

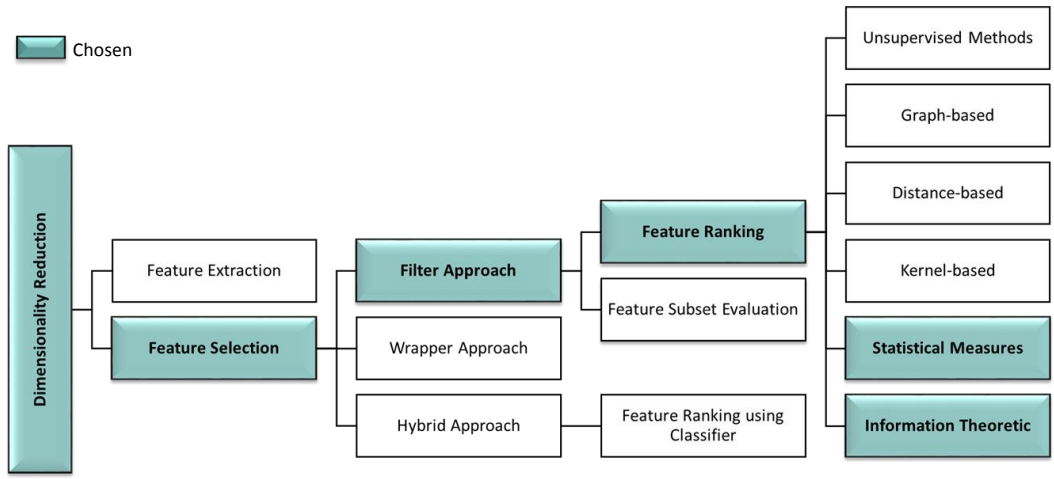


Figure 2.2: Research Taxonomy - Dimensionality reduction and different feature selection approaches [2, 3].

Table 2.1: Feature selection approaches [1, 12–16].

	Filter approach	**Wrapper approach**	**Hybrid approach**
Capabilities	+ Performs simple and fast computation + Not dependent on the classification algorithm + Generally have less computational costs than wrapper and hybrid methods + Better suited to high dimensional datasets	+ Conducts a subset search with an optimal algorithm + Better classification accuracy	+ Requires less computation than wrapper method
Limitations	– Decreases classification performance	– Higher risk of over fitting – High computational cost	– Specific to a learning machine
Examples	Information Gain, Chi-Squared, ReliefF etc.	Sequential Forward or Backward Selection, Genetic Algorithm etc.	Information Gain + Genetic Algorithm etc.

dation for methodologically integrating different feature selection methods based on their shared characteristics. Chen et al. [83] performed a survey on FS algorithms for an intrusion detection system. Experiments were performed for different FS methods i.e. filter, wrapper, and hybrid.

Since this study was not focused on comprehensible classifiers it did not study the effects of FS algorithms on the comprehensibility of a classifier. In addition to this, no unifying methodology was proposed which could categorize existing FS methods based on their common characteristics or their effects on classifiers.

With respect to ensemble feature selection studies, Rokach et al. [73] investigated an ensemble approach that could enhance feature selection; however, the researchers only considered non-ranking filters. Similarly, Jong et al. [74] proposed an ensemble feature ranking methodology that integrated various feature rankings from the same and artificial datasets to improve the stability of feature ranking. In addition, Slavkov et al. [75] conducted a study on various aggregation approaches of the feature rankings of public neuroblastoma microarrays using multiple ranking algorithms and datasets. They showed that aggregating feature rankings produced favorable outcomes compared to the use of a single feature ranking method. Prati [72] also proposed a general framework for the use of ensemble feature ranking to improve the quality of feature rankings, and was able to obtain better results than others. Belanche and Gonzalez [71] performed a thorough study of feature selection algorithms in synthetic problems to evaluate their performance. In this study, a scoring measure was devised to score the output of the feature selection methods, a solution that was considered to be optimal. In addition to this, a comprehensive survey of FS methods was also performed.

In the current study, we have used ensemble-based filter approach. A generalized filter approach is described in Algorithm-1.

This algorithm takes a list of N features $(f_1, f_2, ..., f_n)$ from a given data set D as input and then sequentially passes through mandatory steps to produce best subset S_{best}. S_0 is a subset from which it starts the searching process. It can be either an empty set or a full set, or any random set. δ is a stopping criteria to stop the feature selection process as mentioned earlier. Initially, S_0 is assumed as the best subset and represented by S_{best}. Similarly, evaluate S_0 using an independent measure M and store the result in γ_{best}. Now based on stopping criteria δ, generate subset S from a given data set D. After subset generation, evaluate that subset S against measure M and store the result in γ. After comparing γ with γ_{best}, if γ has a better result, consider γ and S as γ_{best} and S_{best}. In each iteration, values are compared with the previous best one. This process is repeated

Algorithm 1: A generalized filter algorithm [1]

Input : $D - (f_1, f_2, ..., f_n)$ *// a training data set with N features*
S_0 *// a subset from which to start the search*
δ *// a stopping criteria*
Output: S_{best} *// an optimal subset*

initialization;
$S_{best} \leftarrow S_0$
$\gamma_{best} \leftarrow evaluate(S_0, D, M)$;
while $(\delta\ is\ not\ reached)$ **do**
  $S \leftarrow generate(D)$;
  $\gamma \leftarrow evaluate(S,\ D,\ M)$;
  if $(\gamma\ is\ better\ than\ \gamma_{best})$ **then**
    $\gamma_{best} \leftarrow \gamma$;
    $S_{best} \leftarrow S$;
  end
end
return $\boldsymbol{S_{best}}$

until predefined δ stopping criteria is reached. Finally, the algorithm provides best subset S_{best} as an output.

For ensemble-based feature selection studies, various combinations of univariate filter methods are used in the literature, including (i) IG, GR, CS, and SU [3,67], (ii) IG, CS, and SU [84], and (iii) IG, GR, SU, CS, and OneR [72]. In literature, a hybrid approach by combining filter and wrapper methods is also presented that can eliminate unwanted features by using a ranking technique [85]. A similar concept to an EFS approach is also mentioned in [69,86]. For ensemble feature ranking, two aggregate functions called arithmetic mean and median were used to rank features [3]. Authors obtained the ranking by arranging the features from the lowest to the highest. They assigned rank 1 to a feature with the lowest feature index and rank M to a feature with the highest feature index [3]. Similarly, authors aggregated several feature rankings to demonstrate the robustness of ensemble feature ranking that surges with the ensemble size [74]. Onan and Korukoğlu [77] presented an ensemble-based feature selection approach, where different ranking lists obtained from various FS methods were aggregated. Authors used the genetic algorithm (GA) for producing an aggregate ranked list, which is a relatively more expensive technique than a weighted aggregate technique. They performed experiments involving binary class problems; it is not clear how would the proposed method would deal with more complex datasets. Popular filter methods used for the ensemble-based feature selection approach are information gain, gain ratio, chi square, symmetric

uncertainty, OneR, and ReliefF. Most of the feature selection methodologies use three or more of the aforementioned methods for performing feature selection [3, 17, 18, 67, 72, 84]. Finally, feature ranking approach is used in this study as it is considered an attractive approach due to its simplicity, scalability, and good empirical success [3, 87].

A good feature selection algorithm can effectively filter out unimportant features [70]. A feature selection algorithm assesses the usefulness of the features present in the dataset, based on some evaluation metrics. For this study, information-theoretic measures (information gain, gain ratio, and symmetric uncertainty) and co-relational or dependency-based or statistical measures (chi-squared and significance) are utilized. Statistical measures provide good performance in various domains [79] and information-theoretic measures such as entropy are good measures to quantify the uncertainty of features and provide good performance in various domains [2, 79], each of these measures is defined as follows:

Information Gain is an information theoretic as well as symmetric measure, which is computed by following equation [78]:

$$InformationGain(A) = Info(D) - Info_A(D) \tag{2.1}$$

Where *InformationGain(A)* is the information gain of an independent feature *A*. *Info(D)* is the entropy of the entire dataset. $Info_A(D)$ is the conditional entropy of feature *A* over *D*.

Gain Ratio utilizes the split information value that is given as follows [78]:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * \log_2 \frac{|D_j|}{|D|} \tag{2.2}$$

Where *SplitInfo* represents the structure of partitions. Finally, Gain Ratio is defined as follows [78]:

$$GainRatio(A) = InformationGain(A) \ / \ SplitInfo(A) \tag{2.3}$$

Chi-Squared helps to measure the independence of feature from its class. It is defined as follows [78]:

$$CHI(t, c_i) = \frac{N * (AD - BE)^2}{(A + E) * (B + D) * (A + B) * (E + D)} \tag{2.4}$$

$$CHI_{max}(t) = \max_i(CHI(t, c_i)) \tag{2.5}$$

Where *A*, *B*, *E*, *D* represent the frequencies of occurrence of both *t* and C_i, *t* without C_i, C_i without *t*, and neither C_i nor *t* respectively. While N represents the total number of features. The zero value of CHI will represent that both C_i and *t* are independent.

Symmetric Uncertainty is an information theoretic measure to assess the rating of constructed solutions. It is a symmetric measure and is expressed by the following equation [88]:

$$SU(A, B) = \frac{2 * IG(A|B)}{H(A) + H(B)} \tag{2.6}$$

Where $IG(A|B)$ represents the information gain computed by an independent feature *A* and the class-attribute *B*. While *H(A)* and *H(B)* represent the entropies of the features *A* and *B*.

Significance of an attribute A_i is denoted by $\sigma(A_i)$, which is computed by the following equation:

$$\sigma(A_i) = \frac{AE(A_i) + CE(A_i)}{2} \tag{2.7}$$

Where $AE(A_i)$ represents the cumulative effect of all possible attribute-to-class association of an attribute A_i, which is computed as follows:

$$AE(A_i) = \left(1/k \sum_{r=1,2,\ldots,k} {\vartheta_i}^r \right) - 1.0 \tag{2.8}$$

Where *k* represents the different values of the attribute A_i.

Similarly, $CE(A_i)$ captures the effect of change of an attribute value by changing of a class decision and represents the association between the attribute A_i and various class decisions, which is computed as follows:

$$CE + (A_i) = (1/m) * \left(\sum_{j=1,2,..,m} {A_i}^j \right) - 1.0 \tag{2.9}$$

Where *m* represents the number of classes, while $+(A_i)$ depicts the class-to-attribute association of the attribute A_i.

In order to identify an appropriate cut-off value studies for the threshold, Sadeghi and Beigy [2] proposed a heterogeneous ensemble-based methodology for feature ranking. Authors used the genetic algorithm to determine the threshold value; however, a θ value is required to start the process. Moreover, the user is given an additional task of defining the notion of relevancy and redundancy of a feature. The proposed wrapper-based method is tightly coupled with the performance evaluation of a single classifier i.e. SVM; hence losing the generality of the method. Osanaiye et al. [18] combined the output of various filter methods; however, a fixed threshold value i.e. 1/3 of a feature set, is defined a priori irrespective of the characteristics of the dataset. Sarkar et al. [17] proposed a technique that aggregates the consensus properties of Information gain, Chi-Square, and Symmetric Uncertainty feature selection methods to develop an optimal solution; however, this technique is not comprehensive enough to provide a final subset of features. Hence, a domain expert would still need to make an educated guess regarding the final subset. To define cut-off points to remove irrelevant features, a separated validation set and artificially generated features approaches are used [72], however, it is not clear how to find the threshold for the features' ranking [18, 80]. Finding an optimal cut-off value to select important features from different datasets is problematic [80].

2.2 Overview of domain knowledge construction

This section describes the important aspects of the data science (DS) process. It deals with: (1) DS background and the *Cross Industry Standard Process for Data Mining* (CRISP-DM) methodology, (2) methodological studies of knowledge construction approaches, and (3) controlled natural languages background and methodological studies for domain model construction.

The term DS was used in the early 1960s to cover six processes [89]–problem identification, data collection, data preprocessing, data analysis, data modeling, and product evaluation, in order to extract knowledge for decision-making. Text mining (TM) is a multidisciplinary research area, which derives high-quality information from textual data. TM includes information retrieval, natural language processing, data mining (DM), machine learning, and others [66]. Data mining is generally considered a sub-step of the DS process [89]. CRISP-DM, published in the year 2000, is a widely-used systematic methodology for developing DM/DS projects. It is considered to be the de facto standard [19] for executing a DM project systematically. Gupta [90] discussed software development and CRISP-DM, two different approaches to the data mining process. In the software development approach, the data mining process includes six steps: 'requirement analysis,' 'data selection and collection,' 'cleaning and preparing data,' 'data mining exploration and validation,' 'implementation, evaluation, and monitoring,' and 'results visualization.'

According to Abacha and Zweigenbaum [6], "the medical knowledge is growing significantly every year. According to some studies, the volume of this knowledge doubles every five years, or even every two years". Since most of the information available in digital format is unstructured [91], the information extraction problem has attracted wide interest in several research communities [92]. Rajni and Taneja [93] proposed a framework, called U-STRUCT that converts textual documents into an intermediate structured form; however, a knowledge engineer is required to convert that intermediate form into fully structured form. Similarly, Friedman et al. [94] developed an approach, which maps the textual data into UMLS codes for translating them into a structured form (XML format); however, their approach does not support lexical ambiguity and requires a knowledge engineer as well as domain knowledge for structured translation. Leao et al. [95] proposed an ontology learning methodology using OntoUML. They converted unstructured text into structured form by utilizing WordNet lexicon to study word-sense disambiguation.

Reuss et al. [96] proposed and implemented a semi-automatic methodology to extract knowledge from unstructured as well as semi-structured data. The proposed methodology does not support lexical ambiguity.

For knowledge construction, keyword extraction is a vital technique for textual data as well as information retrieval, automatic indexing, text summarization, text mining, text clustering, text categorization, topic detection, and question-answering [4, 5, 97]. Loh et al. [98] noted that concept extraction is a low cost process that helps to build a vocabulary for constructing/discovering domain knowledge. Haggag [4] described that both qualitative and quantitative techniques can be used for keywords extraction task. Qualitative techniques are considered reliable, while quantitative techniques are preferable due to handling multiple text processing tasks. According to Chen and Lin [99], machine learning approaches can be used for keyword extraction; however, as this approach is used in specific domains and for moving to other domains, re-learning is required to build that domain model. Zhu et al. [100] utilized supervised methods for extracting the term relations; however, they required human help to tag the data for learning an extractor. Wenchao et al. [101] presented a keyword extraction approach using a thesaurus; however, the man-made thesaurus are unable to follow the abrupt changes in textual information. In the literature various methodologies are used, which are represented in Figure 2.3.

Similarly, various technologies are used that help to construct the domain knowledge from textual data. Each method/technique/tool involved in knowledge construction process has advantages and disadvantages, which are illustrated in Tables 2.2, and 2.3.

Kuhn [8] described how controlled natural language (CNL) is similar to natural language and humans can easily understand it. CNL is a restricted language, which can be processed and interpreted by computers. This language preserves its essential properties, while restricting its syntax, semantic, and lexicon [111]. CNL was proposed to build knowledge bases (ontologies). Multiple CNLs have been developed to build semantic web ontologies such as *Attempto Controlled English* (ACE), *Sydney OWL Syntax* (SOS), *Controlled Language for Ontology Editing* (CLOnE), and Rabbit. In the literature, various categories of controlled natural languages are used, which are represented in Figure 2.4.

CNLs have been successfully used in various commercial applications such as machine trans-

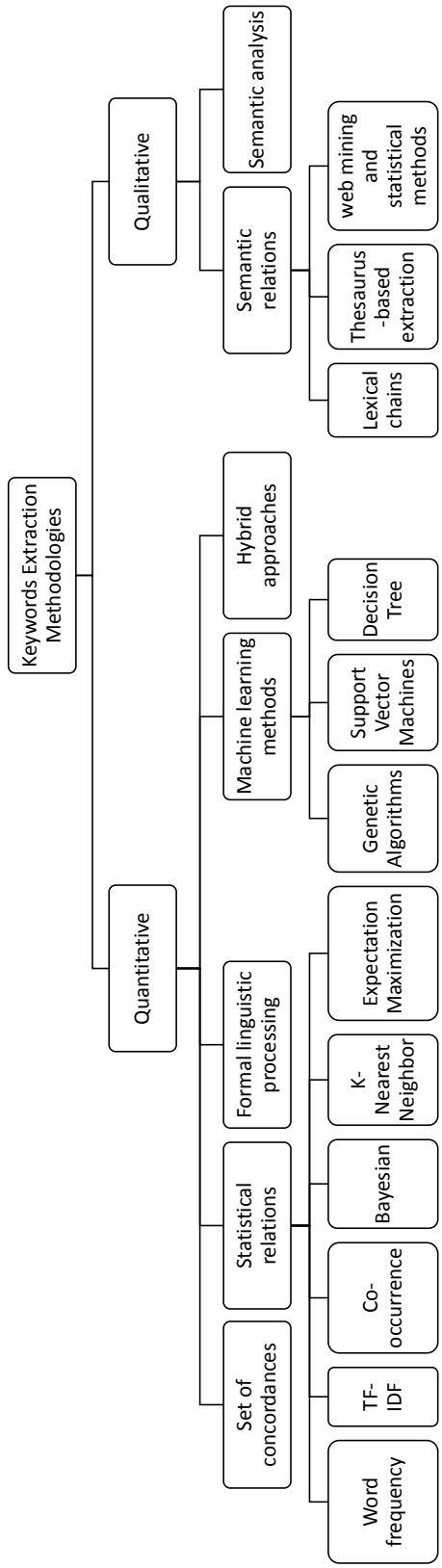


Figure 2.3: Keyword extraction methodologies [4–7]

Table 2.2: Advantages and disadvantages of technologies used for domain knowledge construction.

Reference	Method / Technique / Tool	Advantages	Disadvantages
[102]	Corpus dependent approach for keyword extraction	• Provides better performance	• Requires documents and fixed keywords to develop a prediction model for single domain
[4,5,7,99, 103–105]	Statistical approaches for keyword extraction	• Considered as simplest models, • Used for Complex terms extraction	• Filter out important infrequent keyword, • Results have low precision, • Require hand-annotated data sets for learning, • Restricted by word-related features and become more complex by adding more features.
[103, 106–108]	Word Frequency Analysis - Term Frequency Inverse Domain Frequency (TF-IDF)	• Determines good candidate keywords, • Most commonly used due to having simplicity and effectiveness characteristics	• Does not always discover meaningful relationship between words, • Term frequency Ignores the contents' semantics
[91]	Co-occurrence strategy	• Constructs rules in fast manner, • Much simpler strategy to seek the relevant terms without syntactic or semantic consideration	• Relatively low precision
[6]	MetaMap for medical entity recognition	• Maps medical text to UMLS concepts for identifying the precise concepts	• Recognizes some common words as medical terms, • Considers multiple concepts and their semantic types for the same term, • Needs a disambiguation step to obtain required concept.
[101]	Naïve Bayes technique	• Simple technique to produce good results	• Requires manually assigned keywords for model training.

Table 2.3: Advantages and disadvantages of technologies used for domain knowledge construction (cont.)

Reference	**Method / Technique / Tool**	**Advantages**	**Disadvantages**
[6]	Domain-dependent relation extraction methods	• Extracts relations using meta-thesaurus and semantic network of UMLS	• Dependent on domain knowledge
[4, 5, 109]	WordNet	• Enables to calculate the similarity between noun as well as verb pairs, • Provides semantic features within words	• Supports limited vocabulary and not covers all domains
[98]	Word-sense disambiguation	• Natural language processing techniques help to solve the ambiguity problems	• Requires complex algorithms and time to analyze the text, • Requires knowledge models and rules
[110]	Restricted vocabulary or thesaurus	• Produce consistent results	• Construction and maintenance of thesaurus
[106, 109]	Lexical chains	• Widely used in text summarization, • Locate terms and their sequence in quick and accurate manner	• Not fully explored in keyword extraction problems, • Is an exhaustive method

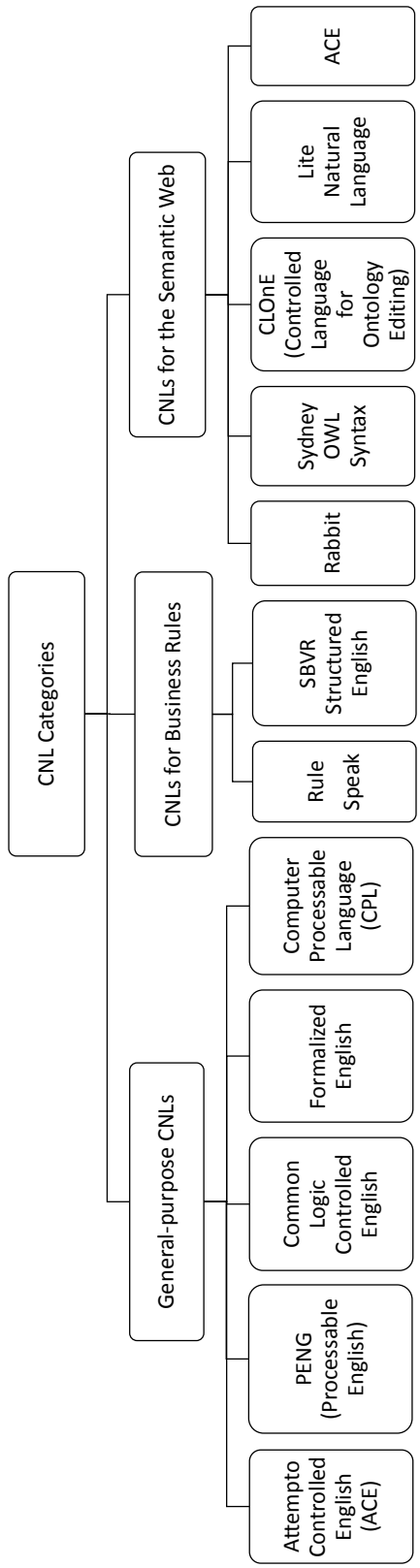


Figure 2.4: Controlled natural languages categories [8]

lation, information management, mobile communication, and so on [112]. Shiffman et al. [113] translated a complete set of guideline recommendations into computer-interpretable statements using controlled natural language. Similarly, in GuideLines Into Decision Support (GLIDES) project, BRIDGE-Wiz used controlled natural language to formalize a process for writing implementable recommendations to improve guideline quality [114].

For computer processability, the CNL is written in formal logic. The basic purpose of defining CNL is to design computer-processable text for improving machine translation. Similarly, Safwat and Davis [115] noted that controlled natural languages (CNLs) facilitate non-expert users to develop ontologies of varying sizes in an easy-to-use manner. Williams et al. [116] described how CNLs are knowledge representation languages, which help non-expert users to translate their knowledge into a computer interpretable form without involvement of a knowledge engineer. Schwitter [117] worked on communication among humans with different native languages and used CNL to represent the formal notations. He concluded that CNL can improve human communication. In addition, Miyabe and Uozaki [112] described various features of CNL, namely that they:

- Enhance readability
- Improve the terms dis-ambiguity
- Are easy to understand
- Reduce misunderstanding
- Minimize the role of knowledge engineer
- Reduce the human translation cost, and
- Improve re-usability of knowledge

Kuhn [8] designed a CNL, called *Attempto Controlled English* (ACE), which is considered one of the most mature CNLs. ACE was developed in early 1995 and has been under development for more than 20 years. This language is most widely used in the academic domain. Its vocabulary is not fixed and varies based on the particular problem domain. ACE also covers all four

design principles, as compared to other CNLs, which do not satisfy all principles. In addition, it is acknowledged to be an unambiguous language. Similarly, Denaux [118] also described some features of the ACE language; he noted that ACE can be used for ontology construction without knowing the knowledge of web ontology language (OWL). It supports all kinds of ontology expressiveness. In addition, it is easy to use for all domain experts.

One of the key problem of CNL is the writability problem, i.e. how to write statements that satisfy the restrictions of the language. Power et al. [119] defined that, "The domain expert can define a knowledge base only after training in the controlled language; and even after training, the author may have to try several formulations before finding one that the system will accept." and similarly, Schwitter et al. [120] stated that, "It is well known that writing documents in a controlled natural language can be a slow and painful process, since it is hard to write documents that have to comply with the rules of a controlled language." It is very difficult to write a syntactically correct statement without any external support. In order to resolve the writability problem of CNLs, Kuhn [8] has mentioned three approaches, namely *Error messages*, *Predictive editors*, and *Language generation*. He also designed the predictive editor and described how the predictive editor is showing the most promise to resolve the writability problem. Schwitter [121] also mentioned that a predictive interface of an editor can help to write correct CNL sentences for building a knowledge base [121].

For evaluating the CNLs, ontographs are considered a simple and powerful approach [122, 123]. Kuhn described how ontographs are intuitive, represent the logic forms in simple manner, and help to understand the core logic [122, 123].

2.3 Overview of case-based learning

This section demonstrates pedagogical concepts, methodologies applied in *case-based learning* (CBL), and related web-based learning systems in medical education. It is further classified into: (1) a background subsection, which describes the basics of CBL with respect to background, features, its comparisons with *problem-based learning* (PBL), and role of experiential knowledge in CBL; (2) an evolutionary technologies subsection, which explains that how IoT technology was used in the medical domain, and how CBL with flip environment was applied in medical

education; and (3) a review subsection, which overviews the existing web-based learning systems, and compares these with well-established CBL systems.

2.3.1 Background for case-based learning

CBL is one of the successful approaches in student-based pedagogy. Jones et al. [124] described that CBL arose from research that indicated that learners who commenced by tackling problems before attempting to understand underlying principles had equal or greater success that learners using a traditional approach. CBL is described as active learning that is focused around a clinical, community or scientific problem. Learning starts with a problem, query or question that the learner then attempts to solve. The learner attempts to solve a specific problem while acquiring knowledge on how to solve similar problems.

CBL was introduced by pedagogy experts to improve knowledge exploration, emphasize critical thinking, achieve better collaboration, and increase opportunities for receiving feedback [125]. Research literature provides multiple features of CBL, such as: (i) it assists students to examine fact-based data, employ analytical tools, articulate their concerns, and draw conclusions for relating to new situations [27, 126], (ii) it offers an opportunity to realize theory in practice [27], and (iii) it develops students' clinical skills in independent and group learning, as well as in communication and critical thinking, to acquire meaningful knowledge for improving students' attitudes towards medical education [26–33]. Because of these features, there are several researchers who have applied CBL in medical education. Fish et al. [34] states Samford University received a grant to apply CBL in undergraduate education. CBL was integrated into the some of the nursing courses. This was successful and as a result CBL was implemented across the entire curriculum. CBL was effectively used in adult health, mental health, pediatric and obstetrical nursing courses. CBL was also used effectively in non-clinical courses such as pathophysiology, statistics and research. Moreover, students studying medicine at the University of Missouri who graduated from 1993 through to 1996 went through a traditional curriculum, whereas students graduating from 1996 through to 2006 went through a CBL curriculum [35]. As part of both curriculums students must pass a 'step 1' test in their third year of study before progressing on to their fourth year. They must complete a 'step 2' test in order to graduate. Since the introduction of the CBL curriculum,

these scores have risen significantly and have remained significantly higher.

CBL is a teaching methodology that utilizes PBL principles. Scavarda et al. [127] and Thistlethwaite et al. [36] described CBL as more structured than PBL as it uses authentic cases for clinical practice. Similarly, Grauer et al. [128] noted that CBL methods require less time and are more efficient in providing large amounts of material compared to PBL. Moreover, Umbrin [129] differentiated PBL from CBL and defined the steps for learning in both PBL as well as CBL. In PBL, the steps are: Problem $\rightarrow$ Explore problem $\rightarrow$ Self-learning $\rightarrow$ Group discussion, while in CBL, the steps are: Prior reading $\rightarrow$ Problem $\rightarrow$ Seeking out extra information $\rightarrow$ Interview with a knowledge expert. Furthermore, the researcher of [129] mentioned that in PBL, students improved their problem solving skills; while in CBL, students learned clinical skills. In addition, in PBL, the role of a facilitator is passive as opposed to CBL, where a facilitator's role is active. Finally, the researcher of [129] concluded that CBL is a preferred methodology over PBL.

In specialized literature, medical education programs are considered to be complex due to their diverse interactions amongst participants and environments [20]. Discussion-based learning in a small-group, like CBL, is considered to be a complex system [46]. In small-groups, multiple medical students are interacting and exchanging information with each other, where each student is also a complex system [47]. In health care professional education, students have to tackle uncertain situations resulting from the accumulation of multiple problems [48]. In such situations, everyone has his/her own judgment, opinion, and feedback and will consider these integral as well as appropriate to the situation. Baillergeau and Duyvendak [48] relate this situation with bricolage, and investigated the ways to correlate the non-expert knowledge with other types of knowledge (expert knowledge). In such situations, experiential knowledge (EK) is considered a valuable resource [48, 130], which can facilitate and provide lived knowledge to students for enhancing individual's participation and user empowerment [48].

According to Willoughby [49], "Experiential knowledge is a knowledge of particular things gained by perception and experience". Similarly, Baillergeau and Duyvendak [48] noted that "Experiential knowledge is a type of knowledge that has the potential to enhance the understanding of the nature, causes and most effective responses to social problems". EK either recalled from experiences, or learned, or acquired [131] is mostly utilized for problem solving. Teachers, gen-

eral practitioners, and social workers are the leading experts that provide experiential knowledge. These experts provide competent interventions utilizing their practical knowledge that is built up using experiential or lay knowledge. Experiential knowledge can be domain-specific as well as holistic and is mostly described in the form of statements [131]. The idea of experiential expertise was introduced in early 1980s [132]. Willoughby [49] observed that the brain has remarkable capacity for accumulating information and facts. She mentioned that an older brain has accumulated and stored vastly more information than a younger brain. So an older person has a well of information and experience to draw on. Therefore, age and experience are advantages in fields like coaching, journalism, law, and management. According to Storkerson [131], "The term *experience* refers to the interactions that humans have with their environments". Similarly, Baillergeau and Duyvendak [48] stated that "Practical knowledge is a key element in clinical knowledge and clinicians build this up through face-to-face observations, screening and evaluation of persons". Experiential knowing is an endless practice of perception and decision making, which is an important aspect for analyzing experiential knowledge [131]. Prior [133] explained the nature of experiential knowledge and considered it as a resource for individual deed. In health research, lay knowledge is widely used to deal with health issues; however, this knowledge is not considered as reliable as experiential knowledge, which helps to improve the quality of interactions. Baillergeau and Duyvendak [48] used a number of cases to analyze the role of experiential knowledge in uncertain situations of mental health and youth-related policy areas. They also analyzed the growth in identification of experiential expertise and highlighted important dimensions of experiential knowledge as a resource for action.

2.3.2 Evolutionary technologies for case-based learning

In this study, we have proposed *IoT-based Flip Learning Platform* (IoTFLiP) for medical education, especially, case-based learning; where IoT infrastructure is exploited to support flipped case-based learning in the cloud environment with state of the art security and privacy measures for potential personalized medical data. In order to propose the IoTFLiP, we conducted a literature review in IoT and flip learning research domains. This section covers (1) how IoT technology was used in the medical domain, and (2) how CBL with flip environment was applied in medical

education.

IoT is no longer new to human and it has gained much attention in recent years [134]. According to the Gartner study[1], 26 billion devices could be communicating with one another by 2020 with an estimated global economic value-add of $ 1.9 trillion. It has changed the concept of the virtual world for communication, information exchange, availability, and ease of use. The concepts of device-to-device connectivity is described by IoTivity. In healthcare, IoTivity has been exploited from wellness applications [135] for treatment and patient care, such as using sensors for monitoring and real-time status detection [136]. Apart from the wellness applications of IoT, it has been used for medical treatment, identification of diseases, complications, and prevention. IoTivity has been exploited to overcome the challenges of existing healthcare, hospital information and management systems [137, 138]. IoT offers great promise in healthcare fields especially in reducing the cost of care [139]. Due to its low cost and with reduced sensing device sizes, IoT can play an important role in boosting the learning capability of medical students by providing real-world CBL cases. In current practices, multiple IoT platforms exist with particular features. As health is the primary concern for society and has strong impact on all stakeholders, IoT in healthcare domains not only improves healthcare in society but is also beneficial for macroeconomic conditions[2].

Aazam et al. [140] presented a resource management and pricing model for IoT through fog computing. The authors emphasized the usefulness and importance of customers' history while determining the amount of resources required for each type of service. However, they did not discuss how their resource management can be mapped to flipped learning. This is also the case with another study the same authors presented in [141], where smart gateway architecture is discussed. The authors proposed that several type of services require smart and real-time decision making, which can be performed by a middleware gateway. Our proposed work integrates the features of [140, 141] and builds on those works, providing an architecture of how IoT resources and infrastructure can be used for medical education. In addition to that, various other platforms

[1]Gartner says the Internet of things installed base will grow to 26 billion units by 2020, http://www.gartner.com/newsroom/id/2636073

[2]Transforming economic growth with the industrial Internet Of things, http://www.forbes.com/sites/valleyvoices/2015/01/21/transforming-economic-growth-with-the-industrial-internet-of-things/

and systems have been applied to acquire real-time data through IoT devices such as *Masimo Radical-7®*, *Freescale Home Health Hub reference platform*, *Remote Patient Monitoring* [139], *IoT-enabled mobile e-learning platform* [142], *Remote Monitoring and Management Platform of Healthcare Information* (RMMP-HI) [143]. They have been proposed or implemented in specific domains for particular applications without flip learning, as well as CBL, for the purpose of medical education.

With the flipped learning environment, the effectiveness of CBL is surprisingly improved. The flipped classroom is a pedagogical framework in which the traditional lecture and assignment elements of a course are flipped or reversed [144]. Students can learn necessary knowledge before the class session, while in-class time is devoted to exercises and discussion by applying the knowledge. In comparing flip learning in CBL with traditional learning practices, Gilboy et al. [145] showed that students preferred flip learning over traditional pedagogical approaches. Similarly, according to Street et al. [146], "The flipped classroom could be a useful and successful educational approach in medical curricula". With the technologies available today, students learn more through active interactions as compared to passively watching the teacher do everything. Lack of such features is one of the main motivations of our proposed flip-based learning for medical education.

2.3.3 Review of existing web-based learning systems

In order to support the learning outcomes of students, a plethora of web-based learning systems have been developed [55–64]. A review of the literature shows that learning systems, *Design A Case* (DAC) [56] and *Extension for Community Healthcare Outcomes* (ECHO) [57] are well established CBL projects. The ECHO platform was developed for case-based learning in which primary and specialty care providers work together to provide care for patients using video conferencing and sharing electronic records. Similarly, the DAC provided an online educational tool, which is designed to supplement traditional teaching and allows for the development of health related virtual cases for medical students. Both ECHO and DAC projects support postgraduate medical students; however, they do not provide domain knowledge support for CBL practice, while ECHO does not support interactive case authoring and formulation.

Ali et al. [55] developed an online CBL tool, called *interactive case-based flip learning tool* (ICBFLT), which formulates the CBL case summaries (e.g., further history, examination, and investigations) of virtual patient through intervention of student as well as medical experts' knowledge. This tool also provides learning services to medical students before attending an actual class. Boubouka [63] designed a case-based learning environment, called *CASes for Teaching and LEarning* (CASTLE) for supporting teaching as well as learning through cases. In CASTLE, a teacher can author the cases for their students and monitor the elaboration of scenarios interpreted by their students. In conclusion, ICBFLT and CASTLE lack the support of acquiring real-world patient cases and do not provide domain knowledge support for CBL practice. For medical training purposes, Dilullo et al. [60] created online predefined case-based tutorials to provide clinical exposure to medical students without the support of acquiring real-world patient cases and without providing feedback to students.

Cheng et al. [59] adopted a web-based prototype system called *Health Information Network Teaching-case System* (HINTS) in practical training of medical students for clinical medicine. They also explained the development mechanism of teaching cases but with no support of providing feedback to students. Shyu et al. [58] established a platform, called *Virtual Medical School* (VMS) for problem-based learning. They utilized their online authoring tools to capture the patient cases from the *Hospital Information System* database. Suebnukarn and Haddawy [61] developed a problem-based learning system, called *Collaborative Medical Tutor* (COMET) for medical students to provide intelligent tutoring during problem solving tasks. The COMET generates tutorial hints to guide medical students in problem solving. Both VMS and COMET have been used for problem-based learning; however, they lacked tutor feedback and domain knowledge support. Sharples et al. [62] described a case-based training system called *MR Tutor* for learning purposes. This system provided computer-assisted training in radiology; it also provided feedback to users without considering tutors' feedback for solved clinical cases. Chen et al. [64] developed a web-based learning system that followed the development of real clinical situations; however their system also lacked the support of feedback and domain knowledge.

2.4 Summary of literature

2.4.1 Feature selection literature

The feature selection (FS) task is considered as one of the most critical problems for selecting appropriate features from a larger set of features [67]. Feature selection performs a key role in the (so-called) process of 'Knowledge Discovery' [69]. Traditionally, this task is performed manually by a human expert, thereby making it more expensive and time-consuming, as opposed to an automatic FS, which has become necessary for the fast paced digital world of today [13].

Feature selection approaches are generally split into three categories: filters, wrappers, and hybrid, where each approach has capabilities and limitations [12–14]. The filter approach [12,15]: (i) is generally much faster and have less computational cost than the wrapper approach, (ii) is better suited to high dimensional datasets, and (iii) provides better generalization. Both evaluate feature relevance without using any learning algorithm [12,67]. The feature selection task requires two basic steps, ranking and filtering. Here the former step requires ranking of all features, while the later involves filtering out of irrelevant features based on some threshold value.

The ranking approach is considered an attractive approach due to its simplicity, scalability, and good empirical success [3, 87]; however, each feature ranking method has its own statistical biases and reveals different relative scales. For example, *information gain* (IG) is biased towards choosing features with a large number of values [17]. Similarly, *chi square* (CHI) is sensitive to sample size [17]. The ensemble feature selection (EFS) approach, has been examined recently by some researchers [69, 86], gives an improved estimation of ranks [3, 69, 147, 148]. The EFS, contains an intuitive concept of ensemble learning and obtains a ranked list of features by incorporating the outcomes of different feature ranking techniques [3, 67]. Popular filter methods used in the ensemble-based feature selection approach are information gain, gain ratio, chi square, symmetric uncertainty, OneR, and ReliefF. Most of the feature selection methodologies use three or more of the aforementioned methods for performing feature selection [3, 17, 18, 67, 72, 84]. In the literature, most of the ensemble-based feature ranking studies are wrapper-based or hybrid-based [2, 3, 69, 72–75, 77, 85, 86], which are relatively more expensive approaches than the filter-based approach. The feature ranking task is important as it requires an optimal cut-off value to

select important features from a list of candidate features. Finding an optimal cut-off value to select important features from different datasets is problematic [80]. With respect to identifying an appropriate cut-off value for the threshold, some studies have been performed [2, 17, 18, 72, 80], which are either wrapper-based to determine the threshold value or domain expert needed to make an educated guess regarding the final subset; or a starting value is required to initiate the process or a fixed threshold value is defined; or a separated validation set and artificially generated features approaches are required, or it is not clear how to find the threshold value.

Taking into consideration the aforementioned discussion, a significant amount of research [2, 17, 18, 71–75, 77, 83] has focused on proposing improved feature selection methodologies; however, not so much consideration is given to how to select features from a given feature set in a comprehensive manner. The availability of a comprehensive feature ranking and filtering approach, which alleviates existing limitations and provides an efficient mechanism for achieving optimal results, is a major problem. State-of-the-art feature selection methodologies have either used relatively more expensive techniques to select the features or required an educated guess to specify a minimum threshold value for retaining important features.

2.4.2 Domain knowledge construction literature

Knowledge is the wisdom of information that plays an important role in decision making [149]. There exists an enormous amount of textual data in a medical domain, which can be useful for medical education, especially for CBL purposes. This overwhelming data provides various opportunities to obtain useful knowledge that reflects the wisdom of information. Declarative knowledge (also called factual knowledge) is a type of knowledge expressed in the form of unstructured text, which can play an important role in health education, decision support, and wellness applications after structured transformation. According to the Simply Philosophy study [65], "Factual knowledge is a justified affirmation of something". It combines the concepts to make an affirmation of something. For example, "Blood_disease" and "is a symptom" make an affirmation "Blood_disease is a symptom". The produced affirmation is either true or false; however, in declarative knowledge it is always true. Handling unstructured content is the foundation to construct the domain knowledge (structured declarative knowledge) required for interactive learning to prepare

medical students for their clinical practice before and outside the class. One way to represent declarative knowledge is ontology, which has been considered as a common way to represent a real-world machine interpretable knowledge and is not constructed systematically [150].

According to Abacha and Zweigenbaum [6], "the medical knowledge is growing significantly every year. According to some studies, the volume of this knowledge doubles every five years, or even every two years". Since most of the information available in digital format is unstructured [91] the information extraction problem has attracted wide interest in several research communities [92]. Text mining (TM) is a multidisciplinary research area, which derives high-quality information from textual data. TM involves the application of techniques from areas such as information retrieval, natural language processing, information extraction, and data mining [66]. In text mining domain, normally text preprocessing, text transformation, feature selection, term extraction, relation extraction, and model construction tasks are involved to construct domain knowledge from textual data. For reliable knowledge construction, keywords as well as their relations are the key elements for knowledge representation, which are mostly extracted from given data using machine learning approaches and a thesaurus [99–101].

In the literature, most of the systems/methodologies [93–95] require a knowledge engineer to translate unstructured text into fully structured form and most of the systems have been proposed or implemented in narrow domains for particular applications using natural language processing techniques and without support of controlled natural language [94, 151, 152]. Regarding structured knowledge construction, some studies do not support lexical ambiguity [93, 96]. We have responded to these deficiencies by including a methodology to construct the domain knowledge (i.e. structured declarative knowledge) from unstructured text. For effective transformation, controlled natural language is used, which constructs syntactically correct and unambiguous computer-processable texts [8].

2.4.3 Case-based learning literature

Case-based learning (CBL) is an active learning approach, which focuses around clinical, community and scientific problems. CBL is a teaching methodology that utilizes problem-based learning (PBL) principles and is preferred over PBL methodology [36, 127, 128]. In CBL, the role of

the facilitator is active and authentic cases for clinical practice are used [36, 129]. The CBL approach is one of the successful approaches in student-based pedagogy and it is widely applied in medical education [26–33]. CBL has been used in clinical as well as non-clinical courses such as nursing courses, adult health, mental health, pediatric, and obstetrical nursing courses, pathophysiology, statistics and research [34, 35]. In professional education for health and social care domains, the clinical case is a key component in learning activities, which includes basic, social, and clinical studies of the patient. Normally, formal learning activities are performed without a real patient case, where interactions are often unplanned and rely on the goodwill of patients [36]. Furthermore, students also feel that classroom CBL activities require a significant amount of time [44]. Sometimes, students feel uncomfortable while participating in group learning activities and they prefer to work alone [45]. In specialized literature, medical education programs are considered to be complex due to their diverse interactions amongst participants and environments [20]. Discussion-based learning in a small-group, like CBL, is considered to be a complex system [46]. In health care professional education, students have to tackle the uncertain situations due to the accumulation of a diverse range of problems [48]. In such situations, everyone has his/her own judgment, opinion, and feedback and will consider this integral as well as appropriate for the situation. In such situations, an experiential knowledge is thought-out as a valuable resource [48, 130], which can facilitate and provide lived knowledge to students for enhancing individual's participation and user empowerment [48].

In the medical area, human (domain expert) judgment is considered as more credible than a computer; however, a human cannot perform fast reasoning computation to work for long periods and they fatigue, as well as feel bored. A computer has the advantage over a human of being able to perform fast reasoning computations, while not experiencing boredom. Being human, students feel that classroom CBL activity requires a significant amount of time and they report tiredness [44]. Medical students tend to choose computer-based cases [36, 52] and opt for web-based cases as compared to lectures for their learning [53,54]. Additionally, more attention is given to online/web-based learning environments [36] while real-life clinical case(s) are increasingly emphasized in medical students' practice [36, 153, 154].

In order to support the learning outcomes of students, a plethora of web-based learning sys-

tems have been developed [55–64]. A review of the literature shows that these systems either do not support computer-based interactive case authoring as well as its formulation, or without the support of acquiring real-world CBL cases or do not provide feedback to students. Currently, very less attention is given to fill the gaps between human-based and computer-based learning. In addition, very little attention is given to the development mechanisms of real-world clinical cases using experiential knowledge and no support of domain knowledge while formulating the case.

Recent trends show that increasing attention is being paid to flipped learning approaches for boosting learning capabilities [145, 155]. As defined by Kopp [156], "Flipped learning is a technique in which an instructor delivers online instructions to students before and outside the class and guides them interactively to clarify problems. While in class, the instructor imparts knowledge in an efficient manner". Currently, CBL is typically performed without exploiting the advantages of the flipped learning methodology, which has significant evidence supporting it over traditional learning methods [55, 145, 146, 157].

Chapter 3

Univariate Ensemble-based Feature Selection

This chapter covers the solutions of the first two research questions/challenges mentioned in the problem statement section of chapter 1 and explains the proposed *Univariate Ensemble-based Feature Selection* (uEFS) methodology, which includes two innovative *Unified Features Scoring* (UFS) and *Threshold Value Selection* (TVS) algorithms to select informative features from a given data for constructing a reliable domain knowledge. The uEFS methodology is evaluated using standard textual as well as non-textual benchmark datasets and achieved (1) on average, a 7% increase in F-measure as compared to the baseline approach, and (2) on average, a 5% increase in predictive accuracy as compared to state-of-the-art methods.

3.1 Introduction

In the domain of data mining and machine learning, one of the most critical problems is the Feature Selection (FS) task, which pertains to the complexity of appropriate feature selection from a larger set of features [67]. Feature selection performs a key role in the (so-called) process of 'Knowledge Discovery' [69]. Traditionally, this task is performed manually by a human expert, thereby making it more expensive and time-consuming, as opposed to an automatic FS which has become necessary for the fast paced digital world of today [13]. Feature selection techniques are generally split into three categories: filters, wrappers, and hybrid, where each technique has capabilities and limitations [12–14]. Popular evaluation methods used for these techniques are *information-theoretic measures*, *co-relational measures*, *consistency measures*, *distance-based measures* and *classification/predictive accuracy*. A good feature selection algorithm can effectively filter out unimportant features [70]. In this regard a significant amount of research has focused on proposing improved feature selection algorithms [71–75]; consequently most of these algorithms use one

or more of the aforementioned methods for performing feature selection. However, there is a lack of a comprehensive framework, which can select features from a given feature set.

This chapter introduces an efficient and comprehensive feature selection methodology, called *Univariate Ensemble-based Feature Selection* (uEFS), which includes two innovative *Unified Features Scoring* (UFS) and *Threshold Value Selection* (TVS) algorithms to select informative features from a given dataset. The uEFS is a consensus methodology for appropriate features' selection in order to generate a useful feature subset for the domain knowledge construction task.

The main intention of the UFS algorithm is to evaluate the feature-set in a comprehensive manner, which is based on different filter-based feature selection measures. In this algorithm, univariate filter measures are employed to assess the usefulness of a selected feature subset in a multi-dimensional manner. The UFS algorithm generates a final ranked list of features after a comprehensive evaluation of a feature set without (a) using any learning algorithm, (b) high computational cost, and (c) without any individual statistical biases of state-of-the-art feature ranking methods. The current version of the UFS has been plugged into a recently developed tool, called *data-driven knowledge acquisition tool* (DDKAT) [158] to assist the domain expert in selecting informative features for the data preparation phase of *cross-industry standard process for data mining* (CRISP-DM). The DDKAT supports an end-to-end knowledge engineering process for generating production rules from a dataset and covers all major phases of the CRISP-DM [158]. The current version of the UFS code and its documentation is open-source and can be downloaded from GitHub [159, 160].

Research shows that the ranking of variables, or ensemble features' selection does not suggest any cut-off point to select only important features [80]. For defining cut off points for removing irrelevant features, a separated validation set and artificially generated features approaches are used [72]; however, it is not clear how to find the threshold for the features' ranking [80]. Finding the optimal value of this threshold for different datasets is problematic. In this regard, an algorithm called *threshold value selection* (TVS), is proposed for feature selection that is empirically based on the data-sets considered in this study. The TVS provides an empirical algorithm to specify a minimum threshold value for retaining important features irrespective of the characteristics of the dataset. It selects a subset of features that are deemed important for the domain knowledge

construction.

The motivation behind the uEFS is to design and develop an efficient feature selection methodology for evaluating a feature subset through different angles and produce a useful reduced feature set for constructing a reliable domain knowledge. In order to accomplish this aim, this study is undertaken with the following objectives: (1) To design a comprehensive and flexible features ranking methodology to compute the ranks without (a) using any learning algorithm, (b) high computational cost, and (c) without any individual statistical biases of state-of-the-art feature ranking methods (see Section 3.2.2), and (2) To identify an appropriate cut-off value for the threshold to select a subset of features irrespective of the characteristics of the dataset with reasonable predictive accuracy (see Section 3.2.3).

The key contributions of this research are as to:

1. Present a flexible approach, called UFS for incorporating state-of-the-art univariate filter measures for feature ranking.

2. Propose an efficient approach, called TVS for selecting a cut-off value for the threshold in order to select a subset of features.

3. Provide proof-of-concept for the aforementioned techniques, after performing extensive experimentation which achieved (1) on average, a 7% increase in f-measure as compared to the baseline approach, and (2) on average, a 5% increase in predictive accuracy as compared to state-of-the-art methods.

This chapter is organized as follows: Section 5.2 covers the methodology of the proposed uEFS approach; the experimental results of the TVS algorithm is discussed in Section 3.3. Section 3.4 provides the details of the uEFS evaluations performed along with results, while Section 5.7 concludes the chapter with a summary of the research findings.

3.2 Materials and methods

This section firstly explains the process of uEFS methodology. Secondly, the UFS algorithm is explained. Thirdly, the TVS algorithm is presented and, lastly, the statistical measures used for

evaluating the performance of the proposed uEFS methodology are explained.

3.2.1 Univariate ensemble-based features selection (uEFS) methodology

In the feature selection process, normally two steps are required [80]. In the first step, normally features are ranked, whereas in the second step, a cut-off point is defined to select important features and to filter out the irrelevant features. In this regard, the proposed UFS algorithm [158] covers the first step of feature selection, while the TVS algorithm covers the second step.

Figure 3.1 shows the functional details of the proposed uEFS methodology, which consists of three major components, called the *Unified Features Scoring*, *Threshold Value Selection*, and *Select Features*. The *Unified Features Scoring* component evaluates the feature-set in a comprehensive manner and generates a final ranked list of features. For example, feature f_2 has the highest priority, then feature f_4 and so on as shown in Figure 3.1. Similarly, the *Threshold Value Selection* component defines a cut-off point for selecting important features. Finally, the *Select Features* component filters out the irrelevant features from the final-ranked list of features based on a cut-off point, and selects a subset of features which are deemed important for the classifier construction. For example, $f_2, f_4, f_1, ..., f_{n-45}$ are the list of features that were selected by the proposed uEFS methodology as shown in Figure 3.1.

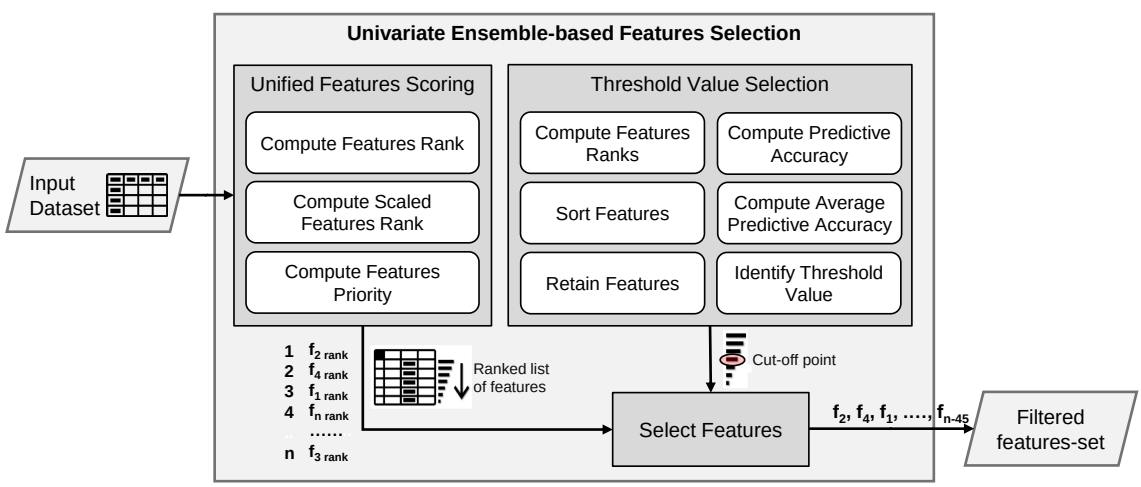


Figure 3.1: uEFS - Univariate ensemble-based features selection methodology.

3.2.2 Unified features scoring (UFS)

Unified Features Scoring, called UFS is an innovative feature ranking algorithm that attempts to unify different feature selection measures. The intention of the UFS algorithm is to evaluate the feature-set in a comprehensive manner, which is based on different filter-based feature selection measures. In this algorithm, univariate filter measures are employed to assess the usefulness of a selected feature subset in a multi-dimensional manner. It uses an intuitive approach to ensemble learning and produces a final ranked list by combining the results of various feature ranking techniques [3, 67].

The following is a rationale for the approaches used in UFS. The feature selection methods are generally split into three categories: *filters*, *wrappers*, and *hybrid* [12–14]. The UFS focuses on filter-based methods, which evaluates feature's relevance in order to assess its usefulness without using any learning algorithm [12, 67]. The filter methods [12, 15]: (i) are generally much faster and have less computational costs than wrapper methods, (ii) are better suited to high dimensional datasets, and (iii) provide better generalization. They evaluate feature's relevance without using any learning algorithm [12, 67]. Filter-based feature selection methods are further split into two subcategories: univariate and multivariate. UFS focuses on univariate filter measures due to simplicity and high performance characteristics [77]. The UFS algorithm uses the *ensemble feature selection* (EFS) approach, which has been examined recently by some researchers [69, 86]. The EFS, an intuitive concept of ensemble learning obtains a final ranked list by combining the outcomes of various feature ranking techniques [3, 67]. Generally, the purpose of the EFS approach is to reduce the risk of selecting an irrelevant feature, yield more robust feature subsets, give an improved estimation to the most favorable subset of features, and finally to improve classification performance [3, 69, 147, 148]. As mentioned in [3], fewer studies have focused on the EFS approach to enrich feature selection itself. Although ensemble-based methodologies have additional computational costs, these costs are affordable due to offering an advisable framework [161]. As mentioned in [3], there are three types of filters' approaches: *ranking*, *subset evaluation*, and a *new feature selection framework* that decouples the redundancy analysis from relevance analysis. The UFS uses a *ranking* approach as it is considered an attractive approach due to its simplicity, scalability, and good empirical success [3, 87]. Feature ranking measures the relevancy of the

features (i.e. independent attributes) by their correlations to the class (i.e. dependent attribute) and ranks independent attributes according to their degrees of relevance [67]. These values may reveal different relative scales. To avoid the impact of multiple relative scales, the UFS rescales the values to the same range (i.e. between 0 and 1) using min-max normalization (MMN) to make it scale insensitive. The MMN is defined as follows:

$$MMN = \frac{value - min}{max - min} \tag{3.1}$$

For rescaling, the UFS assigns rank 1 to a feature with the highest feature index, as opposed to [3], which assigned rank 0 to a feature with the highest feature index. After features rescaling, the UFS uses an ordered-based ranking aggregation approach as it is easy to implement, scale insensitive, and elegant as well as being an effective technique [72]. The ordered-based ranking aggregation method combines the base rankings and considers only the ranks for ordering the attributes [72]. Finally, the UFS applies an arithmetic mean as an aggregate function to compute relative feature weights and their ranking priorities.

UFS is explained through Algorithm 2. This algorithm takes a data set (i.e., D) as input and sequentially passes this through mandatory steps of the algorithm to compute ranks (scores) of the features. UFS is based on n univariate filter-based measures. The key rationale for n filter measures is to evaluate a feature through different considerations.

In Algorithm 2, the first step was to compute the number of features from a given dataset. In the second step, each feature in a data set was ranked using n number of univariate filter-based measures as shown in line-4 to line-7 of Algorithm 2. After that, Algorithm 3 was used to scale (normalize) all computed ranks using the first filter measure. This process was replicated for other $(n - 1)$ measures as well as shown in line-9 to line-12. Once each feature is evaluated and scaled according to different filter measures then different ranks of feature were combined as shown in line-18 of Algorithm 2. Later, the comprehensive score of each feature was assessed as shown in line-25 of Algorithm 2. Moreover, the attribute weight was also calculated based on features individual score and combined scores of all the features present in the data set. Finally, attribute priority was computed based on contributions of a feature in terms of its individual measure score

Algorithm 2: Unified Features Scoring (D)

Input : D: Input data set (data)
Output: $FR-$ Features Ranks

$noOfAttrs \leftarrow numAttributes(data)$ // compute the number of attributes ;
/* Consider n attribute evaluation measures, also called univariate filter measures ($AttrEv_1$, $AttrEv_2$, $AttrEv_3$, ..., *and* $AttrEv_n$) */;
/* Compute the ranks using each selected measure */;
$CR_1[] \leftarrow computeRanks(data, AttrEv_1)$ //where CR represents computed ranks;
$CR_2[] \leftarrow computeRanks(data, AttrEv_2)$;
$CR_3[] \leftarrow computeRanks(data, AttrEv_3)$;
$CR_n[] \leftarrow computeRanks(data, AttrEv_n)$;
/* Compute the scaled ranks of each computed ranks using Algorithm 3 */;
$scaledRanks_1[] \leftarrow scaleRanks(CR_1)$ // invoke Algorithm 3 ;
$scaledRanks_2[] \leftarrow scaleRanks(CR_2)$ // invoke Algorithm 3 ;
$scaledRanks_3[] \leftarrow scaleRanks(CR_3)$ // invoke Algorithm 3 ;
$scaledRanks_n[] \leftarrow scaleRanks(CR_n)$ //invoke Algorithm 3;
/* Compute the combined sum of all computed ranks */;
$combinedranksSum \leftarrow 0$;
$combinedRanks[]$;
for $\forall\ noOfAttrs \in D$ **do**
  /* For each attribute, compute the combined rank by adding all computed scaled ranks */;
  $combinedRanks_i \leftarrow \sum_{j=1}^{n} scaledRanks_{ji}$ //where n represents the number of filter measures;
  $combinedranksSum = combinedranksSum + combinedRanks_i$;
end
/* Rank the list in ascending order */;
$sortedRanks[] \leftarrow sort(combinedRanks)$;
/* Compute the score, weight, and priority of each attribute */;
for $\forall\ noOfAttrs \in D$ **do**
  $attrScores_i \leftarrow combinedRanks_i/n$ //where n represents number of filter measures;
  $attrWeights_i \leftarrow combinedRanks_i/combinedranksSum$;
  $attrPriorities_i \leftarrow attributesScores_i * attributesWeights_i$;
  /* Assign an index (Rank ID) on ascending order to each attribute based on its priority value */;
  $FR[] \leftarrow assignRank(attrPriorities_i)$;
end
return $FR : features\ ranks$

Algorithm 3: Scaling the Computed Ranks (CR)

Input : CR: Input computed ranks (ranks)
Output: $SR-$ Scaled Ranks

```
1  smallest ← ranks_0 ;
2  largest ← ranks_0 ;
3  for ∀ noOfAttrs ∈ CR do
4      if rank_i > largest then
5          largest ← rank_i
6      else
7          if rank_i < smallest then
8              smallest ← rank_i
9          end
10     end
11 end
12 min ← smallest ;
13 max ← largest ;
14 SR[] ← (ranks − min)/(max − min) ;
15 return SR : scaled ranks
```

(line-25) and its relative weightage (line-26) in a data set. This priority value of a feature was further utilized for ranking and feature subset selection.

For the proof of concept, five univariate filter-based measures, namely information gain, gain ratio, symmetric uncertainty, chi-square and significance [3, 67, 72, 84, 158] were used to explain the process of the proposed unified features scoring algorithm. With each of these filter measures, the features are evaluated under various considerations. The rationale for choosing each is as follows:

- *Information gain*, one of the popular feature selection measures, measures how much information a feature provides about the target class [78].
- *Gain ratio* is a disparity measure that enhances the information gain result [78].
- *Symmetrical uncertainty* performs well for highly imbalanced feature sets [88].
- *CHI-square* is a statistical measure that determines the association of a feature with its target class [78].
- *Attribute significance* is a probabilistic measure that assesses an attribute's worth. It is a two-way function that computes the attribute's significance, or association with a class at-

tribute [162].

Using above-mentioned five univariate filter-based measures, the process of the UFS is depicted in Figure 3.2.

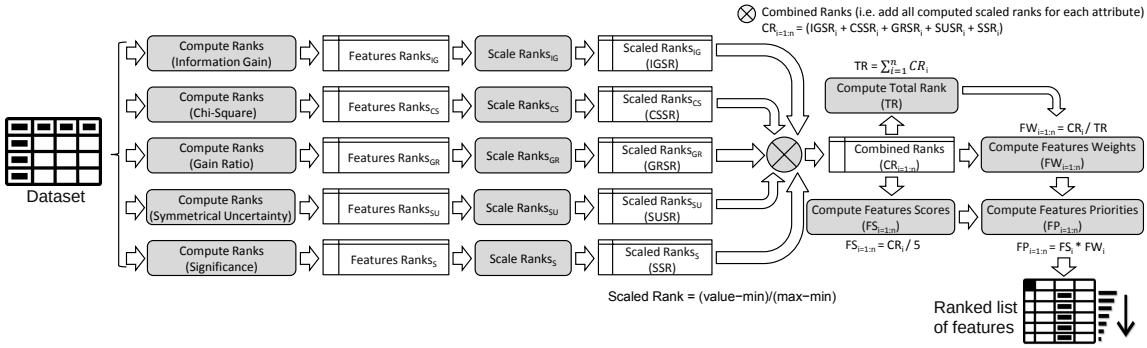


Figure 3.2: UFS - Unified features scoring algorithm.

This process is also explained through a diabetes dataset[1] example, as shown in Figure 3.3.

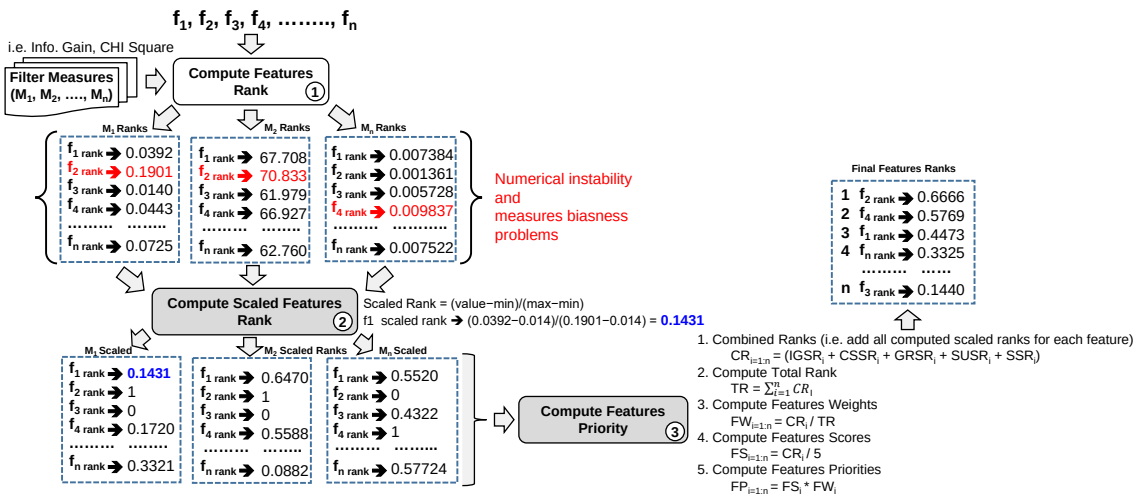


Figure 3.3: Diabetes dataset example for explaining the UFS.

In Figure 3.3, $f_1, f_2, f_3, ..., f_n$ represent the features (such as *preg*, *plas*, *pres*, ..., *age*) of the diabetes dataset, and $M_1, M_2, ..., M_n$ represent the five aforementioned univariate filter-based measures. Ranks are computed using each filter measure. For example, using M_1 (information gain), the computed ranks of each feature are:

[1]https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/

1, rank of @attribute *preg* = 0.0392

2, rank of @attribute *plas* = 0.1901

3, rank of @attribute *pres* = 0.014

4, rank of @attribute *skin* = 0.0443

.....

8, rank of @attribute *age* = 0.0725

After calculating the information gain of each feature, min-max normalization is applied to each attribute. For example, the attribute *preg* is normalized to 0.1431. This process is then replicated for the other measures (M_2,M_3,M_4,M_5). The different ranks of the feature are then combined as shown in line-18 of Algorithm 2; once each feature has been evaluated and scaled according to each filter measure, a comprehensive score of the individual feature is calculated, as shown in Figure 3.3 and in line-25 of Algorithm 2. The attribute weight is also calculated based on the feature's individual score and the combined score of all the features present in the dataset. Finally, attribute priority is computed based on the contribution of a feature in terms of its individual measure score (line-25) and its relative weight (line-26) in a dataset; for example, here f_2 had the highest priority.

3.2.3 Threshold Value Selection (TVS) algorithm

The process of feature selection starts once features are ranked. In order to select a subset of features a threshold value is required. This threshold value specifies those attributes which are deemed important for domain knowledge construction. Those attributes which score less than the minimum threshold value can be discarded without significantly affecting the reliability of knowledge. Hence, specifying the value of a threshold is an important task.

Research shows that finding an optimal cut-off value to select important features from different datasets is problematic [80] and also it is not clear how to find the threshold for the features' ranking [18, 80]. Moreover existing methodologies [17, 18] required an educated guess to specify a minimum threshold value for retaining important features.

Keeping in view these facts, a *threshold value selection* (TVS) algorithm is introduced, which provides an empirical approach for specifying a minimum threshold value. The proposed algo-

rithm is implemented in Java language using WEKA API. TVS is explained through Algorithm 4. This algorithm takes n data sets (i.e., D) as input and sequentially passes these through mandatory steps of the algorithm to find the cut-off value from a predictive accuracy graph.

Algorithm 4: Threshold value selection (TVS) Algorithm

```
   Input  : D − (d_1, d_2, ..., d_n)        // set of n datasets with varying complexities
            C − (c_1, c_2, ..., c_m)          // set of m machine learning classifiers
   Output: V − cut − off value
1  initialization;
2  for d_i ← in D do
3  |  d_i ← computeFeatureRank(d_i)              // rank each feature ;
4  |  d_i ← sortByRankASC(d_i)                   // sort features by rank in ASC ;
5  end
6  P ← 100;
7  for d_i ← in D do
8  |  while P ≥ 5 do
9  |  |  k ← sizeOf(d_i) ∗ (p/100)               // compute partition size ;
10 |  |  Acc ← newSet()                          // initialize empty set ;
11 |  |  for c_i ← in C do
12 |  |  |  P_acc ← predictiveAccuracy(c_i, topKFeatures(d_i, k)) ;
13 |  |  |  Acc.add(P_acc)                       // add accuracy to set ;
14 |  |  end
15 |  |  AVG_acc ← computeAVG(Acc)               // compute average accuracy ;
16 |  |  G ← Plot(AVG_acc, k)                    // plot the average point ;
17 |  |  P ← P − 5                               // decrease the partition size by 5 ;
18 |  end
19 end
20 C ← getCutOffValue(G);
```

In Algorithm 4, first consider the n number of benchmark datasets having varying complexities. After that for each dataset, compute the feature ranks using ranker search mechanism and then sort them in an ascending order as shown in line-3 and line-4 of Algorithm 4. Then partition each dataset into different chunks (filtered dataset) from 100% to 5% features retained. Once filtered datasets are created then consider m number of classifiers from various classifiers category/family having varying characteristics (where $m << n$) and feed each filtered dataset to these classifiers as shown in line-6 and line-11 of Algorithm 4. Following this, record predictive accuracies of these classifiers to each chunk of dataset partitioning using 10-fold cross validation approach (line-12). Later compute the average predictive accuracy of all classifiers as well as datasets against each chunk of dataset partitioning (line-15). Finally, plot all computed average

predictive accuracies against each chunk of dataset partitioning (line-16) and identify the cut-off value from the plotted graph (line-20).

For the proof of concept, eight datasets of varying complexities, were used, to explain the process of the proposed threshold selection algorithm. The process of threshold value selection is depicted in Figure 3.4.

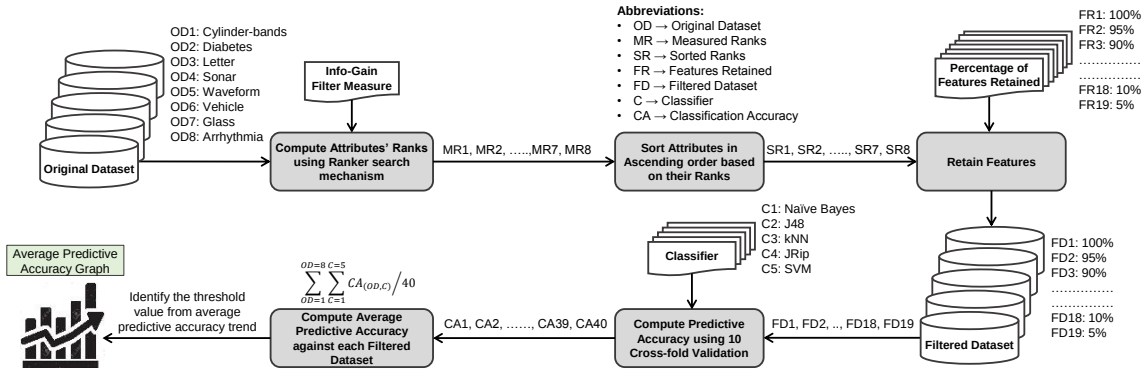


Figure 3.4: TVS - Threshold value selection algorithm.

As depicted in the Figure 3.4, each dataset (*Cylinder-bands*, *Diabetes*, *Letter*, *Sonar*, *Waveform*, *Vehicle*, *Glass*, *Arrhythmia*) was fed to the *Information Gain* filter measure for computing attributes' ranks; then all measured ranks of attributes of each dataset were sorted in ascending order. Afterwards, each dataset was partitioned into different chunks (filtered dataset) from 100% to 5% features retained e.g. in case of 80% chunk, dataset retains nearly 80% highly ranked features while 20% features, which are below the rank, were discarded. Each filtered dataset was fed to 5 well-known classifiers from various classifiers category/family having varying characteristics (Naive Bayes from *Bayes* category, J48 from *Trees* category, kNN from *Lazy* category, JRip from *Rules* category, and SVM from *Functions* category) and then using *10-fold cross validation* approach [72], predictive accuracies of these classifiers were recorded to each chunk of dataset partitioning as illustrated in Tables 3.3, 3.4, and 3.5. Finally, an average predictive accuracy of all classifiers as well as datasets against each chunk of dataset partitioning was computed. The main purpose of this process is to identify an appropriate chunk value, which provides reasonable predictive accuracy and considerably reduces the dataset as well. Through empirical evaluation, it was found that 45% chunk provided a reasonable threshold value of feature subset selection (see

Figure 3.5 in Section 3.3).

3.2.4 State-of-the-art feature selection methods for comparing the performance of the proposed uEFS methodology

In this study, both single feature selection methods, namely information gain (IG), gain ratio (GR), symmetric uncertainty (SU), chi-square (CS), significance (S), one rule (OneR), Relief, ReliefF, and decision rule-based feature selection (DRB-FS); and ensemble-based feature selection methods, namely (GR-χ^2), borda method, and ensemble-based multi-filter feature selection (EMFFS) method, were used as state-of-the-art feature selection methods for comparing the performance of the proposed uEFS methodology [3, 17, 18, 67, 72, 84, 158]. Each of the feature selection methods is defined as follows:

Information Gain (IG) is an information theoretic as well as a symmetric measure, which is one of the popular measures for feature selection. It is calculated based on a feature's contribution in enhancing information about the target class label. An equation for information gain is given as follows [78]:

$$InformationGain(A) = Info(D) - Info_A(D) \tag{3.2}$$

Where *InformationGain(A)* is the information gain of an independent feature or attribute *A*. *Info(D)* is the entropy of the entire dataset. $Info_A(D)$ is the conditional entropy of attribute *A* over *D*.

Gain Ratio (GR) is considered as one of the disparity measures that provides normalized score to enhance the information gain result. This measure utilizes the split information value that is given as follows [78]:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * \log_2 \frac{|D_j|}{|D|} \tag{3.3}$$

Where *SplitInfo* represents the structure of *v* partitions. Finally, Gain Ratio is defined as follows [78]:

$$GainRatio(A) = InformationGain(A) \ / \ SplitInfo(A) \tag{3.4}$$

Chi-Squared (CS) is a statistic measure, which computes the association between the attribute *A* and its class or category C_i. It helps to measure the independence of attribute from its class. It is defined as follows [78]:

$$CHI(A, C_i) = \frac{N * (F_1F_4 - F_2F_3)^2}{(F_1 + F_3) * (F_2 + F_4) * (F_1 + F_2) * (F_3 + F_4)} \tag{3.5}$$

$$CHI_{max}(A) = \max_i(CHI(A, C_i)) \tag{3.6}$$

Where F_1, F_1, F_3, F_4 represent the frequencies of occurrence of both *A* and C_i, *A* without C_i, C_i without *A*, and neither C_i nor *A* respectively. While N represents the total number of attributes. The zero value of CHI will represent that both C_i and *A* are independent.

Symmetric Uncertainty (SU) is an information theoretic measure to assess the rating of constructed solutions. It is a symmetric measure and is expressed by the following equation [88]:

$$SU(A, B) = \frac{2 * IG(A|B)}{H(A) + H(B)} \tag{3.7}$$

Where $IG(A|B)$ represents the information gain computed by an independent attribute *A* and the class-attribute *B*. While *H(A)* and *H(B)* represent the entropies of the attributes *A* and *B*.

Significance (S) is a real-valued two-way function used to assess the worth of an attribute with respect to a class attribute [162]. The significance of an attribute A_i is denoted by $\sigma(A_i)$, which is computed by the following equation:

$$\sigma(A_i) = \frac{AE(A_i) + CE(A_i)}{2} \tag{3.8}$$

Where $AE(A_i)$ represents the cumulative effect of all possible attribute-to-class association of an

attribute A_i, which is computed as follows:

$$AE(A_i) = \left(1/k \sum_{r=1,2,...,k} {\vartheta_i}^r \right) - 1.0 \tag{3.9}$$

Where *k* represents the different values of the attribute A_i.

Similarly, $CE(A_i)$ captures the effect of change of an attribute value by changing of a class decision and represents the association between the attribute A_i and various class decisions, which is computed as follows:

$$CE + (A_i) = (1/m) * \left(\sum_{j=1,2,...,m} {A_i}^j \right) - 1.0 \tag{3.10}$$

Where *m* represents the number of classes, while $+(A_i)$ depicts the class-to-attribute association of the attribute A_i.

One Rule (OneR) is the rule-based method to generate a set of rules, which test one particular attribute. The details of this method can be found in [163].

Relief [11] and *ReliefF* [164] are the distance-based methods to estimate the weightage of a feature. The original Relief method deals with discrete and continuous attributes; it does not handle incomplete data and is limited to two-class problems. The ReliefF is an extension of the Relief method, which covers the limitations of the Relief method. The details of these methods can be found in [11, 164].

Decision Rule-Based Feature Selection (DRB-FS) is a statistical measure to eliminate all irrelevant and redundant features. It allows one to integrate domain-specific definitions of feature relevance, which are based on high, medium and low correlation that is measured using Pearson's correlation coefficient, which is computed as follows [2, 9]:

$$r_{XY} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{(n-1) S_X S_Y} \tag{3.11}$$

Where $\bar{x}$ and $\bar{y}$ represent the sample means, while S_X and S_Y are the sample standard deviations for the features X and Y respectively. Here, n represents the sample size.

GR-χ^2 is an ensemble ranking method, which simply adds the computed ranks of the gain ratio and chi-squared methods [2].

Borda method is a position-based ensemble scoring mechanism, which aggregates ranking results of features from multiple feature selection techniques [17]. The final rank of a feature is computed as follows:

$$score_{final} = \sum_{i=1}^{n} score_{pos(i,j)} \tag{3.12}$$

Where n represents the total number of feature selection techniques, while $pos(i, j)$ is the j^{th} position of a feature ranked by the i^{th} feature selection technique.

Ensemble-based multi-filter feature selection (EMFFS) is an ensemble feature selection method, which combines the output of four filter methods, namely information gain, gain ratio, chi-squared, and reliefF to obtain an optimum selection [18].

3.2.5 Statistical measures for evaluating the performance of the proposed uEFS methodology

In this study, precision, recall, f-measure, and the percentage of correct classification were used as evaluation criteria for feature selection accuracy [2, 17, 18, 72, 77, 165]; second for processing speed; and a non-exhaustive *k-fold cross-validation* technique (i.e. rotation estimation) for predictive accuracy to measure and assess the performance of machine learning methods or schemes [18, 72, 77, 166–168]. Furthermore, a 10-fold cross-validation (i.e. k = 10) technique was selected for computing predictive accuracy [72, 166].

In order to compute the statistical measures (precision, recall, f-measure, and percentage of correct classification), the following four measures are required:

- *True Positives* (TP) represents the correctly predicted positive values (actual class = yes, predicted class = yes)
- *True Negatives* (TN) represents the correctly predicted negative values (actual class = no, predicted class = no)

- *False Positives* (FP) represents contradictions between actual and predicted classes (actual class = no, predicted class = yes)

- *False Negatives* (FN) represents contradicts between actual and predicted classes (actual class = yes, predicted class = no)

Joshi [169] defined these measures as follows:

Accuracy is a ratio of correctly predicted observation to the total observations, which is computed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.13}$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations, which is computed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.14}$$

Recall (Sensitivity) is the ratio of correctly predicted positive observations to the all observations in actual class - yes, which is computed as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.15}$$

F-measure is the weighted average of Precision and Recall, which is computed as follows:

$$F - measure = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \tag{3.16}$$

3.3 Experimental results of the TVS algorithm

This section first describes the characteristics of classifiers used in explaining the process of the proposed threshold selection algorithm, and then demonstrates the results of the proposed TVS algorithm. The purpose is to interpret as well as comment on the results obtained from experimentation.

In order to explain the process of the proposed threshold selection algorithm, five well-known classifiers from various classifiers category/family as shown in Tables 3.1 and 3.2, including *Naive Bayes*, *J48*, *kNN*, *JRiP*, and *SVM* of varying characteristics were considered. Tables 3.1 and 3.2 show the characteristics of each classifier.

Tables 3.3, 3.4, and 3.5 record predictive accuracies of eight datasets (*Cylinder-bands*, *Diabetes*, *Letter*, *Sonar*, *Waveform*, *Vehicle*, *Glass*, *Arrhythmia*) against five classifiers (*Naive Bayes*, *J48*, *kNN*, *JRip*, *SVM*) with varying threshold values from 100 to 5. In these tables, predictive accuracies are recorded in percentages, which were determined by the *10-fold cross validation* technique; whereas each threshold value represents the percentage of features retained. After recording the predictive accuracies, an average predictive accuracy of all classifiers as well as datasets against each threshold value was computed, which is shown in Figure 3.5. This figure depicts the summarized effects of different threshold values on the predictive accuracy of the datasets present in the Tables 3.3, 3.4, and 3.5.

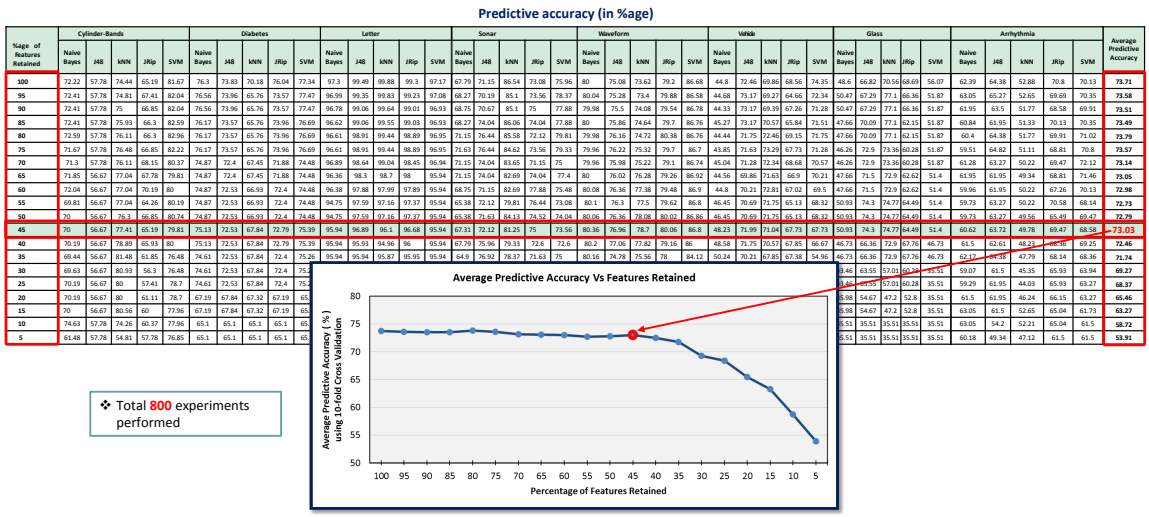


Figure 3.5: An average predictive accuracy graph using the 10-fold cross validation technique for threshold value identification.

Furthermore, predictive accuracies using training examples of these eight datasets were also recorded against the same five classifiers with varying threshold values from 100 to 5. After recording the predictive accuracies, again an average predictive accuracy of all classifiers as well as datasets against each threshold value was computed, which is shown in Figure 3.6.

Table 3.1: Selected classifiers characteristics.

Classifier	Parameters	Description	Classifiers category
Naïve Bayes	useKernelEstimator = False	- Use a kernel estimator for numeric attributes rather than a normal distribution.	Bayes
	useSupervisedDiscretization = False	- Use supervised discretization to convert numeric attributes to nominal ones.	
J48	binarySplits = False	- Whether to use binary splits on nominal attributes when building the trees.	Trees
	confidenceFactor (C) = 0.25	- The confidence factor used for pruning (smaller values incur more pruning).	
	minNumObj (M) = 2	- The minimum number of instances per leaf.	
	subtreeRaising = True	- Whether to consider the subtree raising operation when pruning.	
	unpruned = False	- Whether pruning is performed.	
	useMDLcorrection = True	- Whether MDL correction is used when finding splits on numeric attributes.	
kNN	KNN (K) = 1	- The number of neighbors to use.	Lazy
	distanceWeighting = No distance weighting	- Gets the distance weighting method used.	
	searchAlgorithm = LinearNNSearch	- The nearest neighbour search algorithm to use.	
	distanceFunction = EuclideanDistance	- Implementing Euclidean distance (or similarity) function.	
	attributeIndices (R) = first-last	- Specify range of attributes to act on.	
	windowSize (W) = 0	- Gets the maximum number of instances allowed in the training pool. A value of 0 signifies no limit to the number of training instances.	

Table 3.2: Selected classifiers characteristics. (cont.)

Classifier	Parameters	Description	Classifiers category
JRip	checkErrorRate = True	- Whether check for error rate >= 1/2 is included in stopping criterion.	Rules
	folds (F) = 3	- Determines the amount of data used for pruning. One fold is used for pruning, the rest for growing the rules.	
	minNo (N) = 2.0	- The minimum total weight of the instances in a rule.	
	optimizations (O) = 2	- The number of optimization runs.	
	seed (S) = 1	- The seed used for randomizing the data.	
	usePruning = True	- Whether pruning is performed.	
SVM	c (C) = 1.0	- The complexity parameter C.	Functions
	toleranceParameter (L) = 0.001	- The tolerance parameter (shouldn't be changed).	
	Epsilon (P) = 1.0E-12	- The epsilon for round-off error (shouldn't be changed).	
	filterType (N) = 0 (Normalize training data)	- Determines how/if the data will be transformed.	
	numFolds (V) = -1	- The number of folds for cross-validation used to generate training data for logistic models (-1 means use training data).	
	randomSeed (W) = 1	- Random number seed for the cross-validation.	
	kernel (K) = PolyKernel	- The kernel to use.	
	cacheSize (C) = 250007	- The size of the cache (a prime number).	
	Exponent (E) = 1.0	- The exponent value.	

Table 3.3: Predictive accuracy (in %age) of classifiers using benchmark datasets.

%age of Features Retained	Naive Bayes	J48	kNN	JRip	SVM	Naive Bayes	J48	kNN	JRip	SVM	Naive Bayes	J48	kNN	JRip	SVM
	Cylinder-Bands					Diabetes					Letter				
100	72.22	57.78	74.44	65.19	81.67	76.3	73.83	70.18	76.04	77.34	97.3	99.49	99.88	99.3	97.17
95	72.41	57.78	74.81	67.41	82.04	76.56	73.96	65.76	73.57	77.47	96.99	99.35	99.83	99.23	97.08
90	72.41	57.78	75	66.85	82.04	76.56	73.96	65.76	73.57	77.47	96.78	99.06	99.64	99.01	96.93
85	72.41	57.78	75.93	66.3	82.59	76.17	73.57	65.76	73.96	76.69	96.62	99.06	99.55	99.03	96.93
80	72.59	57.78	76.11	66.3	82.96	76.17	73.57	65.76	73.96	76.69	96.61	98.91	99.44	98.89	96.95
75	71.67	57.78	76.48	66.85	82.22	76.17	73.57	65.76	73.96	76.69	96.61	98.91	99.44	98.89	96.95
70	71.3	57.78	76.11	68.15	80.37	74.87	72.4	67.45	71.88	74.48	96.89	98.64	99.04	98.45	96.94
65	71.85	56.67	77.04	67.78	79.81	74.87	72.4	67.45	71.88	74.48	96.36	98.3	98.7	98	95.94
60	72.04	56.67	77.04	70.19	80	74.87	72.53	66.93	72.4	74.48	96.38	97.88	97.99	97.89	95.94
55	69.81	56.67	77.04	64.26	80.19	74.87	72.53	66.93	72.4	74.48	94.75	97.59	97.16	97.37	95.94
50	70	56.67	76.3	66.85	80.74	74.87	72.53	66.93	72.4	74.48	94.75	97.59	97.16	97.37	95.94
45	70	56.67	77.41	65.19	79.81	75.13	72.53	67.84	72.79	75.39	95.94	96.89	96.1	96.68	95.94
40	70.19	56.67	78.89	65.93	80	75.13	72.53	67.84	72.79	75.39	95.94	95.93	94.96	96	95.94
35	69.44	56.67	81.48	61.85	76.48	74.61	72.53	67.84	72.4	75.26	95.94	95.94	95.87	95.95	95.94
30	69.63	56.67	80.93	56.3	76.48	74.61	72.53	67.84	72.4	75.26	95.94	95.94	95.92	95.94	95.94
25	70.19	56.67	80	57.41	78.7	74.61	72.53	67.84	72.4	75.26	95.94	95.94	95.92	95.94	95.94
20	70.19	56.67	80	61.11	78.7	67.19	67.84	67.32	67.19	65.1	95.94	95.94	95.99	95.94	95.94
15	70	56.67	80.56	60	77.96	67.19	67.84	67.32	67.19	65.1	95.94	95.94	95.94	95.94	95.94
10	74.63	57.78	74.26	60.37	77.96	65.1	65.1	65.1	65.1	65.1	95.94	95.94	95.94	95.94	95.94
5	61.48	57.78	54.81	57.78	76.85	65.1	65.1	65.1	65.1	65.1	95.94	95.94	95.94	95.94	95.94

Table 3.4: Predictive accuracy (in %age) of classifiers using benchmark datasets.

%age of Features Retained	Naive Bayes	J48	kNN	JRip	SVM	Naive Bayes	J48	kNN	JRip	SVM	Naive Bayes	J48	kNN	JRip	SVM
	Sonar					Waveform					Vehicle				
100	67.79	71.15	86.54	73.08	75.96	80	75.08	73.62	79.2	86.68	44.8	72.46	69.86	68.56	74.35
95	68.27	70.19	85.1	73.56	78.37	80.04	75.28	73.4	79.88	86.58	44.68	73.17	69.27	64.66	72.34
90	68.75	70.67	85.1	75	77.88	79.98	75.5	74.08	79.54	86.78	44.33	73.17	69.39	67.26	71.28
85	68.27	74.04	86.06	74.04	77.88	80	75.86	74.64	79.7	86.76	45.27	73.17	70.57	65.84	71.51
80	71.15	76.44	85.58	72.12	79.81	79.98	76.16	74.72	80.38	86.76	44.44	71.75	72.46	69.15	71.75
75	71.63	76.44	84.62	73.56	79.33	79.96	76.22	75.32	79.7	86.7	43.85	71.63	73.29	67.73	71.28
70	71.15	74.04	83.65	71.15	75	79.96	75.98	75.22	79.1	86.74	45.04	71.28	72.34	68.68	70.57
65	71.15	74.04	82.69	74.04	77.4	80	76.02	76.28	79.26	86.92	44.56	69.86	71.63	66.9	70.21
60	68.75	71.15	82.69	77.88	75.48	80.08	76.36	77.38	79.48	86.9	44.8	70.21	72.81	67.02	69.5
55	65.38	72.12	79.81	76.44	73.08	80.1	76.3	77.5	79.62	86.8	46.45	70.69	71.75	65.13	68.32
50	65.38	71.63	84.13	74.52	74.04	80.06	76.36	78.08	80.02	86.86	46.45	70.69	71.75	65.13	68.32
45	67.31	72.12	81.25	75	73.56	80.36	76.96	78.7	80.06	86.8	48.23	71.99	71.04	67.73	67.73
40	67.79	75.96	79.33	72.6	72.6	80.2	77.06	77.82	79.16	86	48.58	71.75	70.57	67.85	66.67
35	64.9	76.92	78.37	71.63	75	80.16	74.78	75.56	78	84.12	50.24	70.21	67.85	67.38	54.96
30	64.42	71.15	80.29	73.08	72.12	80.12	74.74	73.22	77.2	83.24	46.81	61.7	63.83	60.64	50.47
25	62.98	70.67	73.56	69.23	73.56	75.24	72.92	69.62	74.42	79.86	44.92	61.58	61.58	57.68	47.52
20	63.46	71.63	69.23	71.15	74.52	66.3	64.62	58.28	66.82	70.52	43.85	57.33	53.31	54.49	46.57
15	58.65	69.23	64.9	66.83	69.23	59.14	57.58	51.32	57.42	61.22	41.49	50.12	49.29	42.08	42.55
10	56.73	62.02	57.69	57.69	58.17	51.78	50.42	42.28	48.54	51.78	40.07	43.62	40.9	32.62	30.85
5	55.29	50.48	53.85	54.33	56.73	39.02	38.56	34.44	36.06	38.38	25.65	25.65	25.65	25.65	25.65

Table 3.5: Predictive accuracy (in %age) of classifiers using benchmark datasets.

%age of Features Retained	Naive Bayes	J48	kNN	JRip	SVM	Naive Bayes	J48	kNN	JRip	SVM
	Glass					Arrhythmia				
100	48.6	66.82	70.56	68.69	56.07	62.39	64.38	52.88	70.8	70.13
95	50.47	67.29	77.1	66.36	51.87	63.05	65.27	52.65	69.69	70.35
90	50.47	67.29	77.1	66.36	51.87	61.95	63.5	51.77	68.58	69.91
85	47.66	70.09	77.1	62.15	51.87	60.84	61.95	51.33	70.13	70.35
80	47.66	70.09	77.1	62.15	51.87	60.4	64.38	51.77	69.91	71.02
75	46.26	72.9	73.36	60.28	51.87	59.51	64.82	51.11	68.81	70.8
70	46.26	72.9	73.36	60.28	51.87	61.28	63.27	50.22	69.47	72.12
65	47.66	71.5	72.9	62.62	51.4	61.95	61.95	49.34	68.81	71.46
60	47.66	71.5	72.9	62.62	51.4	59.96	61.95	50.22	67.26	70.13
55	50.93	74.3	74.77	64.49	51.4	59.73	63.27	50.22	70.58	68.14
50	50.93	74.3	74.77	64.49	51.4	59.73	63.27	49.56	65.49	69.47
45	50.93	74.3	74.77	64.49	51.4	60.62	63.72	49.78	69.47	68.58
40	46.73	66.36	72.9	67.76	46.73	61.5	62.61	48.23	68.36	69.25
35	46.73	66.36	72.9	67.76	46.73	62.17	64.38	47.79	68.14	68.36
30	43.46	63.55	57.01	60.28	35.51	59.07	61.5	45.35	65.93	63.94
25	43.46	63.55	57.01	60.28	35.51	59.29	61.95	44.03	65.93	63.27
20	35.98	54.67	47.2	52.8	35.51	61.5	61.95	46.24	66.15	63.27
15	35.98	54.67	47.2	52.8	35.51	63.05	61.5	52.65	65.04	61.73
10	35.51	35.51	35.51	35.51	35.51	63.05	54.2	52.21	65.04	61.5
5	35.51	35.51	35.51	35.51	35.51	60.18	49.34	47.12	61.5	61.5

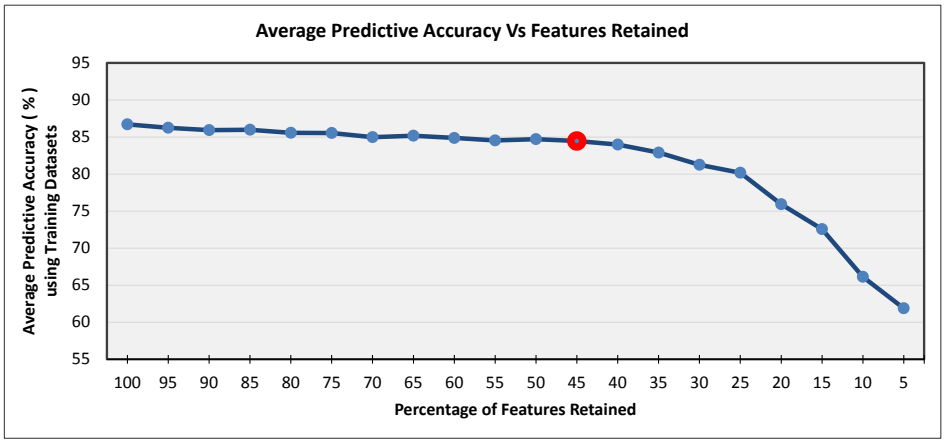


Figure 3.6: An average predictive accuracy graph using training datasets for threshold value identification.

It can be observed from Figures 3.5 and 3.6 that the average predictive accuracy remained consistent from the 100% feature set retained i.e. no feature selection, to 45% features retained. After reducing the dataset from 45% retained features to 5% retained features, the predictive accuracy started to decline as well. Therefore, a threshold value of 45 is selected and top 55% features were selected. This chunked value (i.e. 45%) was utilized in experimentation for evaluating the uEFS methodology, which provided best results. This value can also be used to cut-off the irrelevant data in future dataset as this value is also comparable to the other values in the studies such as 40% [2, 77] and 50% [170].

3.4 Evaluation of the uEFS methodology

The evaluation phase of any methodology has a key role to investigate the worth of any proposed method. This section describes the evaluation setup and compares the proposed feature selection methodology with state-of-the-art feature selection methods. The purpose is to check the impact of the proposed methodology on features' selection suitability in terms of features' ranking on the precision, recall, f-measure, and predictive accuracy performance measure factors.

3.4.1 Experimental setup

For holistic understanding, two studies were performed to evaluate the uEFS methodology by involving non-textual and textual benchmark datasets. In each study, the methodology is compared with the state-of-the-art feature selection methods using precision, recall, f-measure, and predictive accuracy performance measure factors. The motivation behind comparing the results on the textual and non-textual datasets is to check the scalability of the proposed uEFS methodology from low to high dimensional data, where dimension represents the number of attributes or features.

For the *Study-I*, four textual datasets of varying complexity were selected, namely *MiniNewsGroups*[2], *Course-Cotrain*[3], *Trec05p-1*[4], and *SpamAssassin*[5].

These datasets are in textual form and to apply the features ranking algorithms on these datasets, there is need to preprocess the textual data into structured form. In order to perform text preprocessing, the following tasks were performed:

1. Remove HTML tags from web documents, sender as well as receiver information from e-mail documents, urls and etc.
2. Eliminate pictures and e-mail attachments from the documents.
3. Tokenize the documents.
4. Remove the non-informative terms like stop-words from the contents.
5. Perform the term stemming task.
6. Eliminate the low length terms whose length is less than or equal to 2.
7. Finally, generate the feature vectors representing document instances by computing the Term Frequency–Inverse Document Frequency (TF-IDF) weights.

Table 3.6 shows the characteristics of structured form of textual datasets. Our selected datasets are comprised of small to medium size datasets. Both binary and multi-class problems were considered for this study.

[2]http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html
[3]http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-51/www/co-training/data/course-cotrain-data.tar.gz
[4]https://plg.uwaterloo.ca/ gvcormac/treccorpus/
[5]http://csmining.org/index.php/spam-assassin-datasets.html

Table 3.6: Selected textual datasets' characteristics.

Dataset	No. of Features	No. of Documents	No. of Distinct Classes	Description
MiniNewsGroups	27419	1600	4	• Is a 10% subset of 20NewsGroups dataset, • Consider four equal sized categories, namely computer, politics, society and sport
Course-Cotrain	13919	1051	2	• Is a subset of 4Universities dataset, • Consists of web pages, • Consider two categories of pages, namely course and non-course
Trec05p-1	12578	62499	2	• Consists of e-mail documents, • Consider two categories of emails, namely spam and ham
SpamAssassin	9351	3000	2	• Consists of e-mail documents, • Consider two categories of emails, namely spam and ham

For the *Study-II*, eight non-textual benchmark datasets of varying complexity (i.e., small to medium size and binary to multi-class problems) were chosen, namely *Cylinder-bands*, *Diabetes*, *Letter*, *Sonar*, *Waveform*, *Vehicle*, *Glass*, and *Arrhythmia* as shown in the Table 3.7. These datasets were collected from the openML[6] repository.

To select a suitable classifier for assessing the proposed uEFS methodology, initially five well-known classifiers were used: naive Bayes, J48, k nearest neighbors (kNN), JRIP, and support vector machine (SVM) [2, 17, 18, 72, 77, 165, 170, 171]. Using each classifier, predictive accuracy was measured by varying percentage of retained features from 100 to 5 as illustrated in Fig. 3.7. The pictorial results show that of the five classifiers, SVM and kNN tended to perform best on the above-mentioned datasets. Figure 3.7 shows the four datasets, namely *Cylinder-bands*, *Diabetes*, *Waveform*, and *Arrhythmia* on which SVM performed better. Likewise, Fig. 3.7 shows the three datasets, namely *Letter*, *Sonar*, and *Glass* on which kNN performed best. In recent years, the SVM classifier is considered as a dominant tool for dealing with classification problems in a wide range of application [170] and is preferred over other classification methods [171].

[6]http://www.openml.org/

Table 3.7: Selected non-textual datasets' characteristics.

Dataset	No. of Instances	No. of Attributes	No. of Distinct Classes	Description
Cylinder-bands	540	40	2	• Contains the process delay information of engraving printing for decision tree induction.
Diabetes	768	9	2	• Consists of diagnostic measurements, • Consider two prediction categories of patient, namely has diabetes (YES) and not diabetes (NO)
Letter	20000	17	2	• Consists of black-and-white character image features, • Identify English capital alphabet letter (from A to Z).
Sonar	208	61	2	• Contains signals information, • Consider two bounced off categories of signals, namely "bounced off a metal cylinder" and "bounced off a roughly cylindrical rock"
Waveform	5000	41	3	• Contains 3 waves classes, which are produced by integrating 2 of 3 base waves.
Vehicle	846	19	4	• Consists of silhouette features, • Consider/classify four categories of vehicle
Glass	214	10	6	• Consists of oxide content, • Consider/classify six categories of glass
Arrhythmia	452	280	13	• Consists of ECG records, • Consider two prediction categories of cardiac arrhythmia, namely presence of cardiac arrhythmia (YES) and absence of cardiac arrhythmia (NO), • Consider/classify sixteen categories of group

Keeping in view results of the Fig. 3.7 and state-of-the-art classifier considerations, the SVM classifier was chosen to assess the proposed uEFS methodology, as it tends to outperform the F-measures and predictive accuracies for the benchmark datasets [2, 170]. Further, the *SMOreg* function (SVM with Sequential Minimum Optimization) of SVM classifier was used, which is an improved version of the SVM [172]. Table 3.8 shows the characteristics of the selected classifier.

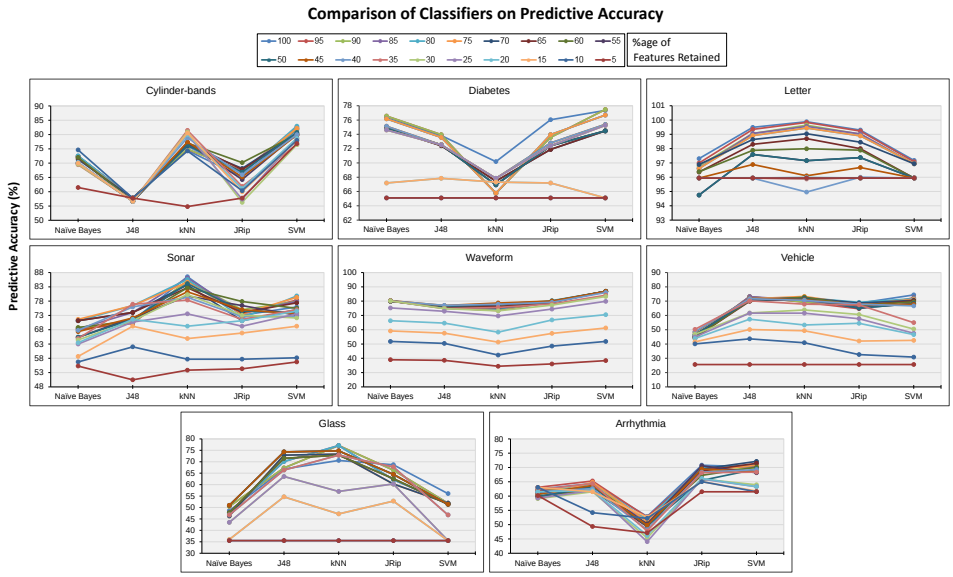


Figure 3.7: Predictive accuracies of classifiers against benchmark datasets with varying percentages of retained features.

Table 3.8: Selected classifier characteristics.

Classifier	Function	Kernel Type	Epsilon	Tolerance	Exponent	Random Seed
SVM	SMO	Polynomial	1.0E-12	0.001	1	1

For comparison purposes, a standard open source implementation of this classifier was utilized as provided by the *Waikato Environment for Knowledge Analysis* (WEKA[7]). Using open source implementation, a method in Java language was written, which computes precision, recall, f-measure, and predictive accuracy of this classifier using the 10-fold cross-validation technique.

Finally, to compare the computational cost, the performance speed of the proposed methodology as well as state-of-the-art methods was measured on a system having the following specifications:

- Processor: Intel (R) Core (TM) i5-2500 CPU @ 3.30GHz 3.30 GHz
- Installed memory (RAM): 16.0 GB
- System type: 64-bit Operating System

[7]http://weka.sourceforge.net/doc.dev/

3.4.2 Experimental execution

For the *Study-I*, a comparison of the proposed uEFS methodology with state-of-the-art feature selection methodologies was performed. The proposed methodology outperforms most of the existing algorithms and individual feature selection measures in terms of f-measure as well as predictive accuracy. It can be observed from Figures 3.8 and 3.9 that the average f-measure and predictive accuracy results of the proposed uEFS methodology on multiple textual benchmarks is higher than existing techniques.

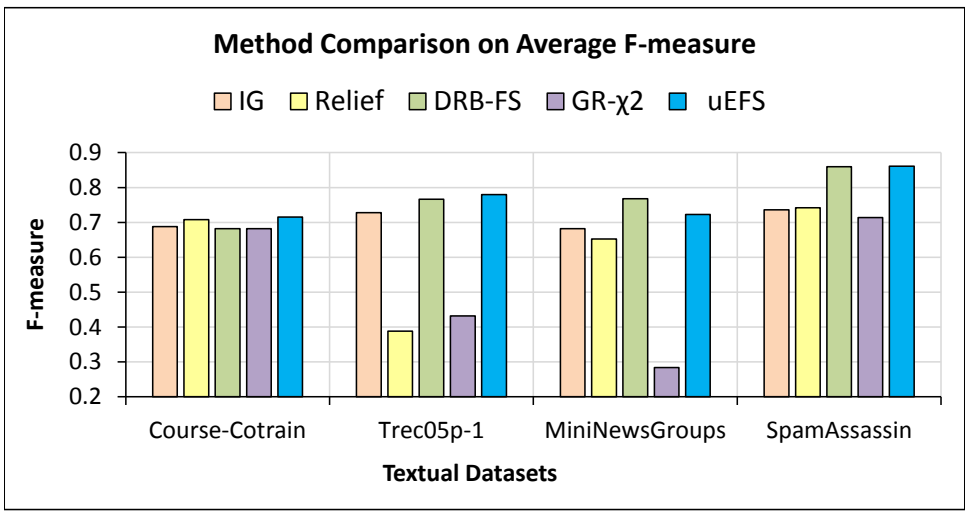


Figure 3.8: Comparisons of F-measure with existing feature selection measures [2, 9–11].

On the other hand, the individual numeric values of precision against each dataset are shown in Table 3.9. On SpamAssassin benchmark; the uEFS outperformed the existing algorithms with the precision of 0.858. Similarly, the uEFS achieved an average of 0.669 precision on Course-Cotrain data, which is close enough to the Relief algorithm with a difference of 0.004, which achieved the highest precision against the existing algorithms. On the other hand, while comparing the average classifier recall, shown in Table 3.10, it was noticed that the proposed uEFS methodology outperforms all of the existing algorithms with the recall of 0.850, 0.864 on Trec05p-1 and SpamAssassin benchmarks respectively.

It can also be observed from the results, shown in Tables 3.9 and 3.10 that in terms of precision and recall, the proposed methodology did not perform better than DRB-FS measure on some

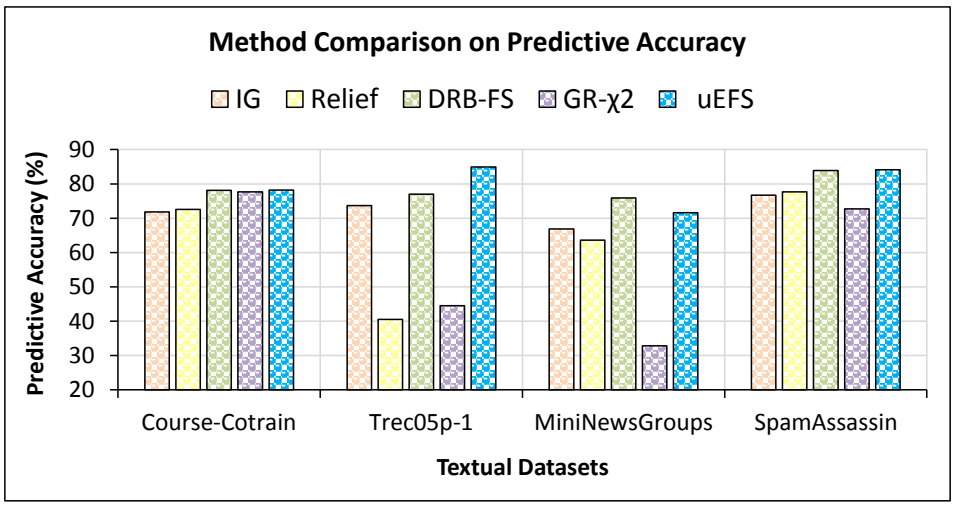


Figure 3.9: Comparisons of predictive accuracy with existing feature selection measures [2,9–11].

datasets due to considering only that measures in the proof of concept purposes, which measures only relevancy and ignoring the feature redundancy factor. As the DRB-FS measure eliminates all irrelevant as well as redundant features and is also based on pre-defined domain-specific definitions of feature relevance [2, 9], therefore there is a chance that the DRB-FS can produce better results as compared to the proposed methodology. However, in terms of f-measure, which is the weighted average of precision and recall, overall the proposed methodology performs better than the DRB-FS measure as shown in Figure 3.8.

Table 3.9: Comparisons of average classifier precision with existing feature selection methods [2, 9–11].

	Feature Selection Algorithms				**Proposed Methodology**
Dataset	**IG**	**Relief**	**DRB-FS**	**GR-χ^2**	**uEFS**
Course-Cotrain	0.668	**0.673**	0.609	0.648	0.669
Trec05p-1	0.836	0.375	**0.839**	0.423	0.721
MiniNewsGroups	0.730	0.708	**0.811**	0.272	0.764
SpamAssassin	0.708	0.710	0.857	0.701	**0.858**

For the *Study-II*, a comparison was made between the proposed uEFS methodology and the five aforementioned univariate filter measures, which were used for the proof of concept. Figure 3.10

Table 3.10: Comparisons of average classifier recall with existing feature selection methods [2, 9–11].

Dataset	Feature Selection Algorithms				Proposed Methodology
	IG	Relief	DRB-FS	GR-χ^2	uEFS
Course-Cotrain	0.717	0.711	**0.780**	0.776	0.768
Trec05p-1	0.731	0.410	0.764	0.451	**0.850**
MiniNewsGroups	0.669	0.636	**0.759**	0.327	0.686
SpamAssassin	0.766	0.778	0.863	0.727	**0.864**

illustrates the difference of the f-measure of the proposed uEFS methodology with each feature selection measure, which is used in the uEFS methodology. It can be deduced from the results, shown in Figure 3.10, that the proposed methodology provides competitive results as compared to state-of-the-art feature selection measures.

For comparison purposes, computed precision and recalls were also used, as recorded in Tables 3.11 and 3.12. The results of these two tables also reveal that the proposed methodology provides competitive results as compared to state-of-the-art feature selection measures. The proposed uEFS methodology yields significant precision and recall on all non-textual benchmarks except the *Glass* dataset, against all existing feature selection measures. On recall comparison, the closest competitors to the uEFS methodology were information gain, gain ratio and symmetrical uncertainty measures, which achieved similar recall of 0.869 on the *Waveform* dataset. While on the other datasets, the existing measures achieved much lower recall as compared to the uEFS. Similarly, on the precision comparison, the chi-squared and symmetrical uncertainty remained the closest competitors to the uEFS on the *Glass* dataset. While on the rest of the datasets, the uEFS outperformed the existing feature selection measures with a significant difference.

A comparison was also made between the predictive accuracies of the uEFS methodology and the five aforementioned univariate filter measures. Table 3.13 illustrates the comparison of the predictive accuracy of the uEFS methodology with five feature selection measures that are used in the uEFS methodology. It can be observed from Table 3.13 results that the proposed methodology provides competitive results as compared to existing feature selection measures. Similarly, it can also be observed from the results, shown in Figure 3.10 and Tables 3.11, 3.12, 3.13 that in terms of

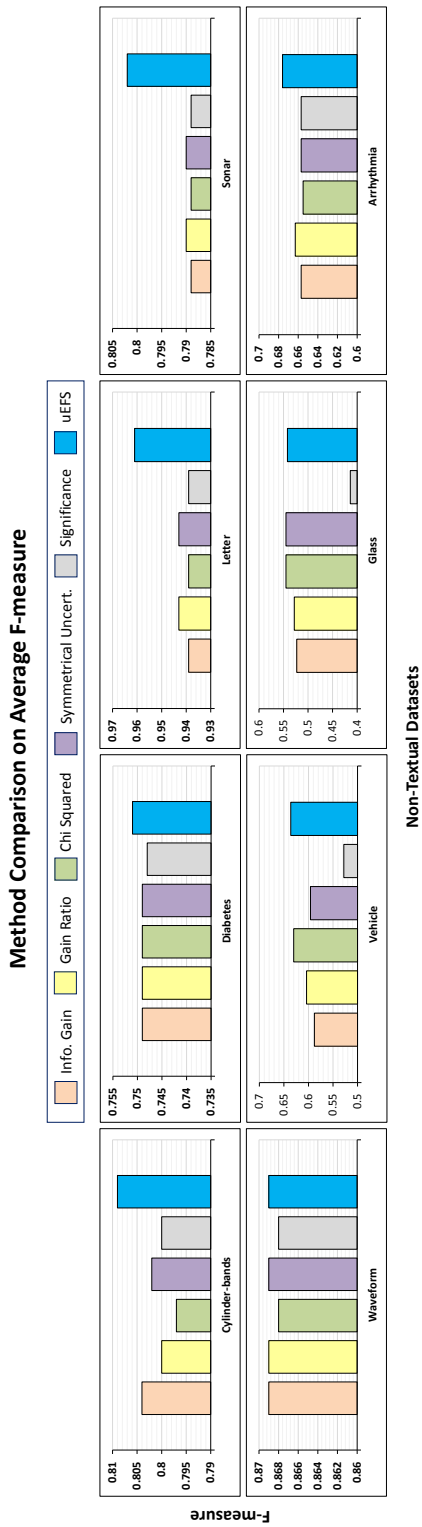


Figure 3.10: Comparisons of F-measure with existing feature selection measures.

Table 3.11: Comparisons of average classifier precision with existing feature selection measures.

Dataset	**Feature Selection Measures**					**Proposed Methodology**
	IG[a]	**GR**[b]	**CS**[c]	**SU**[d]	**S**[e]	**uEFS**
Cylinder-bands	0.805	0.801	0.797	0.803	0.801	**0.811**
Diabetes	0.753	0.753	0.753	0.753	0.738	**0.754**
Letter	0.920	0.962	0.920	0.962	0.920	**0.970**
Sonar	0.789	0.791	0.789	0.791	0.789	**0.803**
Waveform	0.869	0.869	0.868	0.869	0.868	**0.870**
Vehicle	0.586	0.604	**0.642**	0.605	0.534	**0.642**
Glass	0.477	0.484	**0.551**	**0.551**	0.451	0.550
Arrhythmia	0.640	0.647	0.639	0.640	0.639	**0.659**

[a] IG: Information Gain, [b] GR: Gain Ratio, [c] CS: Chi Squared, [d] SU: Symmetrical Uncertainty, [e] S: Significance

Table 3.12: Comparisons of average classifier recall with existing feature selection measures.

Dataset	**Feature Selection Measures**					**Proposed Methodology**
	IG	**GR**	**CS**	**SU**	**S**	**uEFS**
Cylinder-bands	0.806	0.802	0.798	0.804	0.802	**0.811**
Diabetes	0.759	0.759	0.759	0.759	0.758	**0.760**
Letter	0.959	0.961	0.959	0.961	0.959	**0.970**
Sonar	0.788	0.789	0.788	0.789	0.788	**0.803**
Waveform	**0.869**	**0.869**	0.868	**0.869**	0.868	**0.869**
Vehicle	0.617	0.632	0.655	0.631	0.540	**0.658**
Glass	0.579	0.584	**0.589**	**0.589**	0.481	0.584
Arrhythmia	0.719	0.723	0.717	0.719	0.719	**0.728**

f-measure, precision, recall, and predictive accuracy, the proposed methodology did not perform better than existing feature selection measures on the *Glass* dataset due to having small size of data, multiple classes, and imbalanced class characteristics.

The result of *one-sample test* with and without bootstrapping technique is also illustrated in Table 3.13. The purpose of performing this test was to determine whether the values obtained

Table 3.13: Comparisons of predictive accuracy (in %age) of the uEFS with existing feature selection measures using the 10-fold cross validation technique.

Dataset	**Feature Selection Measures**					**Proposed Methodology**	**One-Sample Test p {Sig. (2-tailed)}**	
	IG	**GR**	**CS**	**SU**	**S**	**uEFS**	**Without Bootstrap**	**With Bootstrap**
Cylinder-bands	80.56	80.19	79.81	80.37	80.19	**81.11**	**0.002**	**0.001**
Diabetes	75.91	75.91	75.91	75.91	75.89	**76.04**	**0.000**	**0.002**
Letter	95.94	96.08	95.94	96.08	95.94	**96.97**	**0.000**	**0.001**
Sonar	78.85	78.86	78.85	78.86	78.85	**80.29**	**0.000**	**0.001**
Waveform	86.88	86.88	86.86	86.88	86.86	**86.9**	**0.005**	**0.001**
Vehicle	61.7	63.24	65.48	63.12	54.02	**65.84**	0.093	0.316
Glass	57.94	58.41	**58.88**	**58.88**	48.13	58.41	0.400	0.370
Arrhythmia	71.9	72.35	71.68	71.9	71.9	**72.79**	**0.002**	**0.001**

from the proposed uEFS methodology were significantly different to the values obtained from existing feature selection measures. For performing this test against each dataset, feature selection measures' values were considered as sample data, and the uEFS value as a test value, which is a known or hypothesized population mean. For example, in the case of the *Cylinder-bands* dataset, 81.11 (value generated by the uEFS) was considered a test value, while 80.56, 80.19, 79.81, 80.37, 80.19 (values generated by *Info. Gain*, *Gain Ratio*, *Chi Squared*, *Symmetrical Uncert.*, *Significance*) were used as sample data. The null hypothesis (H_0) and (two-tailed) alternative hypotheses (H_1) of this test will be:

- H_0: 81.11 = $\bar{x}$ ("the mean predictive accuracy of the sample $\bar{x}$ is equal to 81.11")
- H_1: 81.11 $\neq \bar{x}$ ("the mean predictive accuracy of the sample $\bar{x}$ is not equal to 81.11")

In this case, the mean feature selection measures score for *Cylinder-bands* dataset (M = 80.22, SD = 0.28) was lower than the normal uEFS score of 81.11, a statistically significant mean difference of 0.89, 95% CI [0.54 to 1.23], t(4) = -7.141, p = .002. Since $p < .05$, we reject the null hypothesis due to mean predictive accuracy of sample $\bar{x}$ is equal to 81.11 and conclude that the mean predictive accuracy of sample is significantly different from existing methodologies result.

It can be observed from Table 3.13 that most of significance (i.e. p) values are less than 0.05 (i.e. $p < .05$), which shows that the proposed uEFS methodology results are statistically significantly different from the results of existing methodologies.

Table 3.14: Comparisons of predictive accuracy (in %age) of the uEFS with existing feature selection measures using the out-of-sample bootstrapping technique.

Dataset	**Feature Selection Measures**					**Proposed Methodology**	**Bootstrap for One-Sample Test**
	IG	**GR**	**CS**	**SU**	**S**	**uEFS**	**p {Sig. (2-tailed)}**
Cylinder-bands	77.49	77.81	77.34	77.59	77.4	**77.88**	**0.013**
Diabetes	76.27	76.24	76.39	76.27	76.18	**77.74**	**0.000**
Letter	96.56	96.56	96.74	96.7	96.63	**96.8**	**0.011**
Sonar	77.29	76.97	77.27	76.95	76.49	**77.78**	**0.006**
Waveform	86.79	86.68	86.48	86.54	86.31	**86.87**	**0.020**
Vehicle	61.46	62.75	65.28	61.43	54.18	**65.39**	0.077
Glass	51.4	51.3	51.63	52.21	46.45	**53.33**	0.060
Arrhythmia	70.14	70.29	70.21	70.13	**70.45**	70.07	**0.042**

Cross validation and out-of-sample bootstrap sampling techniques are often utilized for approximating the predictive performance of a classification model [173]. The results reported in Table 3.13 for performing a t-test, are computed using *10-fold cross validation* approach [72]. The result of the t-test depends on the independent samples, otherwise, t-tests may yield misleading results. In 10-fold cross-validation, each test set is independent of the others. However, this test still suffers from the problem that the training sets overlap and produced optimistically biased results [173]. This overlap may prevent the t-test from obtaining a good estimate. In order to obtain good estimation of t-test and to remove the biased results, out-of-sample bootstrap sampling was also performed in this study.

Bootstrap is a statistical estimation technique, where a mean is estimated from multiple random samples of data. It provides more robust estimation of a statistical quantity. Out-of-sample bootstrap sampling technique is different from general bootstrap sampling, in which N number of random samples ($B_1, B_2, ..., B_N$) are drawn from original training sample (T), where each drawn sample (B_i, where $i = 1, 2, ..., N$) has the same size as original training sample (T). In out-

of-sample bootstrapping technique, each drawn sample (B_i) is considered as training data, while remaining data ($T - B_i$) is used as a test data. After creating N number of training as well as testing datasets, average performance estimation is computed.

Table 3.14 reports the mean predictive accuracy results of the uEFS and existing feature selection measures, which are computed using out-of-sample bootstrapping technique. For example, in the case of the *Cylinder-bands* dataset (T), 540 random samples or training datasets ($B_1, B_2, ..., B_{540}$) were drawn, which is based on number of instances in a dataset (see Table 3.7). Similarly, 540 test datasets were created. Finally, mean predictive accuracy of existing feature selection measures as well as the proposed methodology is computed. For example, the value 77.49 in Table 3.14, represents the mean predictive accuracy using the IG feature selection measure for the *Cylinder-bands* dataset. It can be observed from Table 3.14 results that the proposed methodology provides competitive results as compared to existing feature selection measures. Table 3.14 also reports the result of *one-sample t-test*. It can be observed from Table 3.14 that most of significance (i.e. p) values are less than 0.05 (i.e. $p < .05$), which shows that the proposed uEFS methodology results are statistically significantly different from the results of existing methodologies.

For evaluating the computation cost of the proposed feature selection methodology, the performance speed was also computed, as shown in Table 3.15. The results show that on average, the proposed methodology takes 0.37 sec more time than state-of-the-art filter measures.

Proposed feature selection methodology is also compared with other well-known feature selection methods (i.e. *OneR* and *ReliefF*) as illustrated in Table 3.16. The results of Table 3.16 also show that the proposed methodology provides competitive results as compared to existing feature selection methods.

For the *Study-II*, a comparison of the proposed uEFS methodology with the two state-of-the-art ensemble methods, namely borda and EMFFS [17, 18] was also performed. A methodological comparison of these two methods with the proposed uEFS methodology is illustrated in Table 3.17. For the proof of concept as well as aforementioned comparisons, five filter measures were used; however, to compare the proposed uEFS methodology with these two state-of-the-art ensemble methods, three [17] and four [18] filter measures defined in each state-of-the-art ensemble method,

Table 3.15: Comparisons of time measure (in seconds) with existing feature selection measures.

Dataset	Feature Selection Measures					Proposed Methodology	ATSM[a]	TD[b]	ATD[c]
	IG	GR	CS	SU	S	uEFS	(sec)	(sec)	(sec)
Cylinder-bands	**4.12**	3.28	3.82	3.79	3.59	**4.53**	3.72	0.81	**0.37**
Diabetes	**0.14**	0.11	0.12	0.12	0.12	**0.17**	0.12	0.05	
Letter	4.60	4.12	**4.63**	4.28	4.60	**4.77**	4.45	0.32	
Sonar	0.06	0.05	**0.08**	0.06	0.06	**0.14**	0.06	0.08	
Waveform	1.11	**1.12**	**1.12**	1.09	**1.12**	**2.09**	1.11	0.98	
Vehicle	**0.33**	0.28	0.30	0.28	0.30	**0.39**	0.3	0.09	
Glass	**0.36**	**0.36**	0.33	0.34	0.33	**0.34**	0.34	0	
Arrhythmia	2.67	2.68	2.54	**2.70**	2.64	**3.31**	2.65	0.66	

[a] ATSM: Average Time of State-of-the-art Measures, [b] TD: Time Difference, [c] ATD: Average Time Difference

Table 3.16: Comparisons of predictive accuracy (in %age) with existing feature selection methods.

Dataset	Feature Selection Methods		Proposed Methodology
	OneR	ReliefF	uEFS
Cylinder-bands	79.63	80.37	**81.11**
Diabetes	75.39	75.52	**76.04**
Letter	**97.14**	96.91	96.97
Sonar	77.88	75.96	**80.29**
Waveform	86.76	**86.90**	**86.90**
Vehicle	64.89	63.83	**65.84**
Glass	49.07	57.01	**58.41**
Arrhythmia	71.02	71.46	**72.79**

were used as mentioned in Table 3.17.

After applying the ensemble-based borda and EMFFS methods, the predictive accuracy and F-measures of the proposed uEFS methodology, using three and four filter measures, were computed, as shown in Tables 3.18 and 3.19. The results of Tables 3.18 and 3.19 show that the proposed methodology provides better results as compared to the two state-of-the-art ensemble methods [17, 18]. It can be observed from the results, shown in Tables 3.18, 3.19 that in terms of predictive

Table 3.17: Comparisons of state-of-the-art ensemble methodologies with the proposed uEFS methodology.

State-of-the-art ensemble methodology–I		**State-of-the-art ensemble methodology–II**	
Borda method [17]	**uEFS methodology**	**EMFFS method [18]**	**uEFS methodology**
1. Consider 3 filter measures (information gain, symmetric uncertainty, chi-squared)	1. Consider 3 filter measures (information gain, symmetric uncertainty, chi-squared)	1. Consider 4 filter measures (information gain, gain ratio, chi-squared, reliefF)	1. Consider 4 filter measures (information gain, gain ratio, chi-squared, reliefF)
2. Compute the ranks using each filter measure	2. Compute the ranks using each filter measure	2. Compute the ranks using each filter measure	2. Compute the ranks using each filter measure
3. Sort the computed ranks in an ascending order	3. Compute the scaled ranks of each computed ranks	3. Sort the computed ranks in an ascending order	3. Compute the scaled ranks of each computed ranks
4. Assign a score to each feature in a list based on its position	4. Compute the combined sum of all computed ranks	4. Select top one-third split of each filter measure's output	4. Compute the combined sum of all computed ranks
5. Compute the sum of all the positional scores from all the lists	5. For each feature, compute the combined rank by adding all computed scaled ranks	5. Define the feature count threshold	5. For each feature, compute the combined rank by adding all computed scaled ranks
6. Sort the computed sum in an ascending order to generate the final ranked feature set	6. Sort the list in an ascending order after computing the score, weight, and priority of each feature	6. Compute the feature occurrence rate among the filter measures	6. Sort the list in an ascending order after computing the score, weight, and priority of each feature
		7. If the feature count is less than the threshold, drop the feature otherwise select the feature	7. Determine the threshold value using the proposed TVS method
			8. Apply the threshold value to drop the irrelevant features and to select the final ranked feature set

accuracy and f-measure, the performance of the proposed methodology is same as state-of-the-art ensemble methods on the *Letter* dataset, while the proposed methodology did not perform better than EMFFS method on the *Arrhythmia* dataset due to having small size of data, multiple classes, and imbalanced class characteristics.

Table 3.18: Comparisons of predictive accuracy and F-measure with Borda method [17].

Dataset	Predictive Accuracy (%)		F-measure	
	Borda method [17]	uEFS (3 filter measures)	Borda method [17]	uEFS (3 filter measures)
Cylinder-bands	57.78	**80.37**	0.423	**0.802**
Diabetes	65.10	**75.91**	0.513	**0.749**
Letter	**95.94**	**95.94**	**0.939**	**0.939**
Sonar	66.83	**78.85**	0.667	**0.789**
Waveform	31.80	**86.88**	0.311	**0.869**
Vehicle	59.22	**63.12**	0.58	**0.596**
Glass	40.19	**58.88**	0.316	**0.545**
Arrhythmia	64.60	**71.90**	0.564	**0.657**

Table 3.19: Comparisons of predictive accuracy and F-measure with EMFFS method [18].

Dataset	Predictive Accuracy (%)		F-measure	
	EMFFS method [18]	uEFS (4 filter measures)	EMFFS method [18]	uEFS (4 filter measures)
Cylinder-bands	80.74	**81.48**	0.805	**0.813**
Diabetes	75.52	**75.91**	0.739	**0.749**
Letter	**95.94**	**95.94**	**0.939**	**0.939**
Sonar	78.37	**80.29**	0.784	**0.803**
Waveform	86.48	**86.90**	0.864	**0.869**
Vehicle	41.73	**63.12**	0.392	**0.596**
Glass	54.67	**58.88**	0.491	**0.545**
Arrhythmia	**73.23**	71.68	**0.672**	0.658

In the proposed uEFS methodology, computed feature ranks are without any given weightages. In order to validate this consideration, the proposed uEFS methodology is compared with and without giving weightage to features. For computing weightage of each attribute, a borda method [17, 165] is used, where a pre-defined score is assigned to each position in a list produced from each univariate filter measure [17]. In this method, a position-based scoring mechanism is

used to compute score of a feature [17], where a final score of each feature is computed by summing all positional scores of that particular feature from all produced lists. After generating a final score list, weightage of each feature is computed using following equation:

$$Weightage = 1 - \frac{(value - min)}{(max - min)} \tag{3.17}$$

Where the $value$ is a final score of feature, while the min and max are minimum and maximum values in a final score list.

This process is explained through a diabetes dataset[8] example, as illustrated in Table 3.20.

Table 3.20: Position-based ranking for computing features weightage.

Univariate Filter-based Measure	**Position**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Information Gain	f_2	f_6	f_8	f_5	f_4	f_1	f_7	f_3
Gain Ratio	f_2	f_6	f_8	f_1	f_5	f_7	f_4	f_3
Chi Squared	f_2	f_8	f_6	f_5	f_4	f_1	f_7	f_3
Symmetrical Uncertainty	f_2	f_6	f_8	f_5	f_1	f_4	f_7	f_3
Significance	f_2	f_6	f_8	f_1	f_5	f_7	f_4	f_3

In Table 3.20, $f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8$ represent the features (such as *preg*, *plas*, *pres*, *skin*, *insu*, *mass*, *pedi*, *age*) of the diabetes dataset. Scaled ranks were computed using each filter measure. For example, using information gain, the computed scaled ranks of each feature were:

1, scaled rank of @attribute *preg* = 0.1431

2, scaled rank of @attribute *plas* = 1.0

3, scaled rank of @attribute *pres* = 0.0

4, scaled rank of @attribute *skin* = 0.1721

5, scaled rank of @attribute *insu* = 0.2584

6, scaled rank of @attribute *mass* = 0.3458

7, scaled rank of @attribute *pedi* = 0.0386

[8]https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/

8, scaled rank of @attribute *age* = 0.3322

After calculating the scaled ranks of each feature using information gain, all features were sorted in a descending order such as f_2, f_6, f_8, f_5, f_4, f_1, f_7, f_3 and then assigned a pre-defined score to each position in a list as shown in first row of Table 3.20; for example, here f_2 (*plas* feature) had the highest priority and is assign score 1. Similarly, f_6 (*mass* feature) had the second highest priority and is assign score 2, and so on. Table 3.20 records all position-based scores of features using each filter measure.

Once each feature has been scored according to each filter measure, a combined position score (final score) of the individual feature is calculated, as illustrated in Table 3.21. Finally, weightage of each feature is computed based on the contribution of a feature in terms of its individual final score, minimum, and maximum values of final scores using each filter measure; for example, in Table 3.21 the f_1 had the final score of 25, while 5 and 40 are the minimum and maximum values. Therefore, weightage of the f_1 feature will be $1 - ((25 - 5)/(40 - 5)) = 0.429$.

Table 3.21: Weightages of features using information gain filter measures.

	Features							
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Combined Position Score	25	05	40	30	22	11	33	14
Weightage	0.429	1	0	0.286	0.514	0.829	0.2	0.743

After computing a weightage value of the individual feature, multiply this value to each scaled ranks value to generate a new scaled value of the individual feature that will be used for computing the combined sum of all computed ranks step (line-13 of Algorithm 2). After applying weighting mechanism, the predictive accuracy and F-measures of the proposed uEFS methodology were computed, as shown in Table 3.22. The results of Table 3.22 show that the proposed methodology (without considering weighting mechanism) provides competitive results as compared to giving weightage to features.

The uEFS methodology was evaluated rigorously on textual and non-textual benchmark datasets having small to high dimensional data size and provides competitive results as compared to state-of-the-art feature selection methods, which indicates that our proposed ensemble approach is more

Table 3.22: Comparisons of predictive accuracy and F-measure with weightage mechanism.

Dataset	Predictive Accuracy (%) of uEFS		F-measure of uEFS	
	Without weightage	With weightage	Without weightage	With weightage
Cylinder-bands	81.11	**82.59**	0.809	**0.824**
Diabetes	**76.04**	**76.04**	**0.751**	0.750
Letter	**96.97**	96.35	**0.961**	0.949
Sonar	**80.29**	78.37	**0.802**	0.784
Waveform	**86.9**	80.46	**0.869**	0.803
Vehicle	**65.84**	**65.84**	**0.636**	0.634
Glass	**58.41**	**58.41**	**0.542**	**0.542**
Arrhythmia	**72.79**	67.04	**0.676**	0.599

robust across textual and non-textual datasets. The above-mentioned results also provide an evidence that the uEFS methodology is stable towards producing same and most likely higher predictive accuracy and f-measure values across a wide variety of datasets.

3.5 Conclusions

Features' selection is an active area of research for the data mining and text mining research community. In this study, we present a *univariate ensemble-based feature selection* (uEFS) methodology to select informative features from a given dataset. For the uEFS methodology, we first propose a *unified features scoring* (UFS) algorithm to evaluate the feature-set in a comprehensive manner for generating a final-ranked list of features. For defining a cut-off point to remove irrelevant features, we then propose a *threshold value selection* (TVS) algorithm to select a subset of features, which are deemed important for the domain knowledge construction. Extensive experimentation was performed in order to analyze the proposed uEFS methodology in different facets. The uEFS methodology was evaluated using standard non-textual as well as textual benchmark datasets and achieved (1) on average, a 7% increase in F-measure as compared to the baseline approach, and (2) on average, a 5% increase in predictive accuracy as compared to state-of-the-art methods. The current version of the UFS has been plugged into a recently developed tool, the

data-driven knowledge acquisition tool (DDKAT), to assist the domain expert in selecting informative features [158]. The current version of the UFS code and its documentation is open-source and can be downloaded from GitHub [159, 160].

Chapter 4

Domain Knowledge Construction

This chapter briefly describes a methodology to construct the machine-readable domain knowledge (i.e. structured declarative knowledge) from unstructured text. The proposed methodology constructs an ontology from unstructured textual resources in a systematic and automatic way, using artificial intelligence techniques with minimum intervention from a knowledge engineer.

4.1 Introduction

Knowledge is the wisdom of information that plays an important role in decision-making. It is able to distinguish between facts and information that is gained through experience and education. Declarative knowledge, also known descriptive knowledge, is a type of knowledge expressed in the form of unstructured sentences. An unstructured document is defined as a document having information in unexpected places [174], for example a hand written note or a dictation etc. In the health-care domain there exists a large volume of heterogeneous unstructured declarative knowledge in the form of medical progress notes, hospital discharge summaries, and clinical guidelines [175, 176].

Handling unstructured contents is the foundation to construct the declarative structured knowledge required for decision support as well as health and wellness systems. The unstructured forms of knowledge resources are important aspects to enable us to comprehend the contents and relationships of knowledge. This declarative knowledge can play an important role in real life applications for better analysis if the unprocessed text is transformed into structured contents (i.e. explicit knowledge). A huge amount of valuable textual data is available on the web, which has led to a corresponding interest in technology for automatically extracting relative information from open data, to convert it into declarative knowledge, and to represent it in a way, which is machine

interpretable. One way to represent this knowledge is the ontology, which represents a machine-readable reality using a restriction-free framework, where you can explicitly define, share, reuse, and or distribute information. An ontology has been considered as a common way to represent a real-world declarative knowledge [150].

For knowledge construction, various knowledge systems have come a long way, from manual knowledge curation to automatic data-driven knowledge generation. The major drivers of this transition were the size and complexity of data. Since large datasets cannot be efficiently analyzed manually, the automation process is essential [177]. Initially in this process of knowledge automation, knowledge engineers followed ad-hoc procedures [178]. Later on, more systematic methodologies were devised, which can be referred to as data-driven knowledge acquisition systems. To gain insights from unstructured data, data science (DS) was created, supporting both automatic and semi-automatic data analysis [179]. Data science is similar to Knowledge Discovery in Databases and is intricately linked to data-driven decision-making concepts [180]. It employs techniques and theories drawn from many fields such as data mining, machine learning, cluster analysis, classification, visualization, and databases [89]. The CRoss-Industry Standard Process for Data Mining (CRISP-DM) is a widely used systematic methodology for DS system development. According to a poll conducted in 2014, CRISP-DM was regarded as the leading methodology for data science projects, data mining, and analytics [181].

Considering the above discussion and the rapid increase in textual data rates, it is almost impossible to extract/construct machine-readable knowledge using manual approaches. The research community prefers to use natural language processing (NLP) techniques to resolve this problem. In the literature, most of the systems/methodologies [93–95] require high intervention of a knowledge engineer to translate unstructured text into a structured form and to resolve the construction of unambiguous machine-readable knowledge. We have responded to these deficiencies by including a methodology to construct the machine-readable domain knowledge (i.e. structured declarative knowledge) from unstructured text. The main motivation for proposing this approach is to automate the ontology development process without requiring extensive training in knowledge engineering, to reduce the human resource cost. The proposed methodology constructs an ontology from unstructured textual resources in a systematic and an automatic way using artificial

intelligence techniques with minimum intervention from a knowledge engineer. In addition, the proposed methodology covers all major phases of CRISP-DM to explain the end-to-end knowledge engineering process. For effective transformation, controlled natural language (CNL) is used, which constructs syntactically correct and unambiguous computer-processable texts.

4.2 Materials and methods

To construct the machine-readable domain knowledge from unstructured text, this section briefly describes (1) the proposed methodology and modules details, and (2) functional mapping of the proposed methodology to the phases of CRISP-DM. Each of these items is explained in the following subsections.

4.2.1 Proposed knowledge construction methodology

This section describes the workflow of the proposed methodology, as shown in Fig. 4.1, as well as the functionality of each module.

Text mining is the process of deriving high-quality information from an unstructured text. It involves the application of techniques from areas like information retrieval, natural language processing, information extraction, and data mining [66]. For constructing machine-readable domain knowledge from textual data, a workflow is shown in Figure 4.1, which consists of six modules, namely *text preprocessing*, *text transformation*, *feature selection*, *terms extraction*, *relations extraction*, and *model construction*.

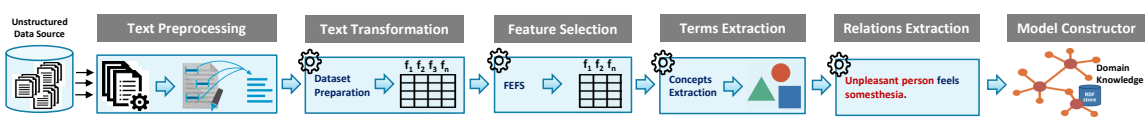


Figure 4.1: A workflow for domain knowledge construction methodology

The brief description of each module is described as follows:

Text preprocessing

The *Text Preprocessing* module applies various basic preprocessing techniques to prepare the textual data. This module consists of four components, namely *Tokenization* for chopping the given text into pieces (tokens), *Filtration* for removing the non-informative terms (such as the, in, a, an, with, etc.), *Tagging* for assigning each token with a parts-of-speech tag, such as noun, verb, etc., and *Normalization* for identifying the root/stem of a word. i.e. the words "connected" and "connecting" are stemmed to "connect".

Text transformation

This module computes the *Term Frequency – Inverse Document Frequency* (TF-IDF) of the extracted tokens to generate the feature vectors (tabular form) representing document instances.

Feature selection

This module applies the proposed feature selection methodology, uEFS to select the important features for domain knowledge construction.

Terms extraction

A concept expresses more concrete and accurate meanings than keywords do. For identifying concept relationships and building a domain ontology, there is need to extract concepts (i.e. named entities) from the given textual data. The Terms Extraction module configures an external thesaurus (i.e. Princeton's WordNet) to identify the concepts by mapping all nouns of the processed textual data with the concepts defined in a thesaurus. This module is responsible for identifying relevant terms.

Relations extraction

For generating concepts hierarchy to build a domain ontology, identification of concept relationships is needed, which can be achieved by using an external semantic lexicon. The Relations Extraction module extracts relations based on linguistic patterns using external semantic lexicons.

This module performs the semantic analyses to define the meanings of words and unambiguous relationships among concepts by mapping with standard or domain vocabularies. Finally, this module validates the generated knowledge from the domain expert before model construction.

Model construction

This module constructs syntactically correct and unambiguous machine processable text and then transforms the relations into structured ontological model, called as domain model, using controlled natural language (CNL). The CNL is preferred to construct the ontological model. As according to [118, 182, 183], CNL can transform the declarative unstructured knowledge into machine interpretable knowledge and can consume less memory as well as computing power.

In order to construct domain knowledge each above-mentioned module has performed some task(s) and used method(s) as illustrated in Table 4.1. For Text Preprocessing, Text Transformation, Terms Extraction, and Relation Extraction modules, the *RapidMiner Studio* was used [184], whereas the *ACE View* was used for the Model Constructing module. The *ACE View* is an ontology editor that uses *Attempto Controlled English (ACE)* to view and edit OWL ontology [185].

4.2.2 Functional mapping of the proposed knowledge construction methodology with phases of the CRISP-DM

CRISP-DM consists of six well-defined phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment [188]. The major goal of developing CRISP-DM was to establish a process model for end-to-end application execution.

This section gives a description of the functional mapping of the proposed methodology to the phases of CRISP-DM, as shown in Table 4.2, which details the tasks performed by the proposed methodology for each phase.

4.3 Realization of the domain knowledge construction methodology

In this section, a diabetes scenario is described to illustrate the proposed methodology. The scenario is explained below based on the above-mentioned modules.

Table 4.1: Methods used for constructing domain knowledge.

Process	Task	Method	Reason
Text preprocessing	Tokenization	English tokenizer	
	Filtration	Stopword removal	
	Normalization	Porters stemmer	
	Tagging	POS tagger	
Text transformation	Technique used	Term Frequency – Inverse Document Frequency (TF-IDF)	1. TF-IDF provides a good heuristic for determining likely candidate keywords [106]. 2. It is one of the best-known and most commonly used keyword extraction algorithms currently in use [107] when a document corpus is available.
Feature selection	Features ranking	UFS algorithm	
	Subset selection	TVS algorithm	
	Filtration	Label filter	
Terms extraction	Process	Nouns, Verbs, Adjectives, and Adverbs Identification	Penn Treebank [186] provides distinct coding for all classes of words having distinct grammatical behavior.
	Thesaurus used	Penn Treebank	
Relations extraction	Technique used	Lexical chaining and heuristics	Lexical chain is a well known technique for text connectivity [187] that locate terms and their sequence in accurate manner [106].
	Thesaurus used	Princeton's WordNet	
	Process	Hypernyms identification	
	Keep original tokens	True	
	Multiple meanings per word policy	Take all meanings per token	
	Multiple synset words	Take only first synset word	
	Validation	Domain expert	
Model construction	Language used	Attempto Controlled English (ACE)	ACE [8] is a logic-based knowledge representation language. 2. It uses the syntax of a subset of English. 3. It provides automatic and unambiguous translation of text into first-order logic.

Table 4.2: CRISP-DM phases and tasks performed in the proposed methodology [19].

Business understanding	**Data understanding**	**Data preparation**	**Modeling**	**Evaluation**	**Deployment**
Understand application domain	Search domain documents	Text tokenization	Select features	Evaluate the results of uEFS methodology	Plan deployment
Identify application goal	Collect initial documents	Remove stopwords	Extract terms	Evaluate the extracted terms	Monitor application impact
Identify application objectives	Analyze documents	Terms stemming	Extract relations	Evaluate the extracted relations	Maintain application
Analyze resource specification (software, hardware)	Remove irrelevant documents	POS tagging	Convert to ACE	Determine next steps	Prepare final report
Prepare application development plan	Store required documents	Text transformation	Construct model		Review application

The steps for realization of the domain knowledge construction methodology are:

1. Load the clinical documents of diabetes and non-diabetes domains.

2. Perform the text preprocessing task, including text tokenization, stopwords removal, tokens filtration, terms stemming, and POS tagging, on loaded documents.

3. Compute the TF-IDF of each term to generate the feature vectors for transforming the text into structured form as shown in Table 4.3.

4. Compute the ranks of each feature using proposed uEFS methodology, and then select the important features (words) of diabetes domain only as shown in Table 4.4.

5. Extract terms (words) after identification of nouns, verbs, adjectives, and adverbs using Penn Treebank as shown in Table 4.5.

6. Extract and identify all entities relations using the lexical chain technique and a heuristic approach. For example, lexical chain extracts '$symptom/$

Table 4.3: A partial view of feature vectors.

action	**agonist**	**.....**	**blood**	**bloodstream**	**bmi**	**.....**	**label**
0.0000	0.0044	.....	0.0119	0.0000	0.0155	.....	diabetes
0.0020	0.0005	.....	0.0510	0.0000	0.0079	.....	diabetes
0.0029	0.0204	.....	0.0323	0.0025	0.0247	.....	diabetes
0.0009	0.0039	.....	0.0306	0.0000	0.0000	.....	diabetes
0.0021	0.0008	.....	0.0530	0.0000	0.0055	.....	diabetes
0.0025	0.0025	.....	0.0816	0.0000	0.0066	.....	diabetes
0.0015	0.0042	.....	0.0431	0.0000	0.0190	.....	diabetes
0.0016	0.0042	.....	0.0437	0.0000	0.0192	.....	diabetes
0.0032	0.0023	.....	0.0303	0.0000	0.0000	.....	diabetes
0.0013	0.0000	.....	0.0000	0.0000	0.0000	.....	non-diabetes
0.0000	0.0000	.....	0.0000	0.0000	0.0000	.....	non-diabetes
0.0007	0.0000	.....	0.0013	0.0000	0.0000	.....	non-diabetes
0.0006	0.0000	.....	0.0007	0.0000	0.0000	.....	non-diabetes
0.0010	0.0000	.....	0.0000	0.0000	0.0000	.....	non-diabetes
0.0007	0.0000	.....	0.0006	0.0000	0.0000	.....	non-diabetes
0.0017	0.0000	.....	0.0021	0.0000	0.0000	.....	non-diabetes
0.0006	0.0000	.....	0.0000	0.0000	0.0000	.....	non-diabetes
0.0022	0.0000	.....	0.0019	0.0000	0.0000	.....	non-diabetes
0.0000	0.0000	.....	0.0000	0.0000	0.0000	.....	non-diabetes

Table 4.4: Top diabetes domain words extracted from clinical documents.

Diabetes domain words				
action	prevention	child	beverage	triglyceride
agonist	sick	cholesterol	bmi	unstable
antidiabetic	stage	dietary	mellitus	reduce
blood	type	eat	diagnose	condition
bodyweight	critical	education	diastolic	woman
chest	cycle	excretion	dietitian	adult
diabetes	drug	glucagon	episode	judgment
diabetic	energy	obese	fat	gestational
diet	external	overweight	foot	height
fatness	failure	plasma	glycemia	cough
glucose	food	pressure	hemoglobin	fatigue
glargine	goal	protection	hemoprotein	breakfast
hormone	healthy	urine	hospitalization	syndrome
insulin	level	complication	hypertension	vital
lifestyle	medication	exercise	injection	avoid
lower	substance	tired	intake	problem
monitor	yield	metformin	intensive	indicator
nutrition	activity	vision	habit	frequent
obesity	aged	hdl	goal	coma
visualize	influenza	hyperglycemia	disease	lispro
amount	adult	hypoglycemia	regular	hyper
walk	breathless	metabolic	pregnancy	thirst
drink	feet	protein	repeat	glimepiride
growth	person	weight	sugar	high
prevent	serum	training	systolic	loss

Table 4.5: Selected words for domain model construction.

Diabetes domain words along-with their weights	
<weight name="blood" value="0.998"/>	<weight name="lispro" value="0.362"/>
<weight name="diabetes" value="0.998"/>	<weight name="hyper" value="0.362"/>
<weight name="diabetic" value="0.998"/>	<weight name="thirst" value="0.362"/>
<weight name="diet" value="0.998"/>	<weight name="glimepiride" value="0.362"/>
<weight name="glucose" value="0.998"/>	<weight name="high" value="0.305"/>
<weight name="glargine" value="0.998"/>	<weight name="loss" value="0.305"/>
<weight name="insulin" value="0.998"/>	<weight name="feeling" value="0.279"/>
<weight name="obesity" value="0.998"/>	<weight name="edema" value="0.273"/>
<weight name="level" value="0.751"/>	<weight name="tension" value="0.273"/>
<weight name="feet" value="0.743"/>	<weight name="unpleasant" value="0.273"/>
<weight name="person" value="0.743"/>	<weight name="negative" value="0.256"/>
<weight name="serum" value="0.743"/>	<weight name="symptom" value="0.231"/>
<weight name="pressure" value="0.743"/>	<weight name="negative_stimulus" value="0.194"/>
<weight name="metformin" value="0.743"/>	<weight name="blood_disease" value="0.165"/>
<weight name="vision" value="0.743"/>	<weight name="bloodpressure" value="0.123"/>
<weight name="hdl" value="0.743"/>	<weight name="somesthesia" value="0.123"/>
<weight name="hyperglycemia" value="0.743"/>	<weight name="blurry" value="0.123"/>
<weight name="weight" value="0.587"/>	<weight name="medicine" value="0.108"/>
<weight name="glycemia" value="0.587"/>	<weight name="feel" value="0.108"/>
<weight name="hypertension" value="0.587"/>	<weight name="swallow" value="0.105"/>
<weight name="disease" value="0.485"/>	<weight name="oat" value="0.060"/>
<weight name="regular" value="0.485"/>	<weight name="urination" value="0.059"/>
<weight name="fatigue" value="0.388"/>	<weight name="hurt" value="0.059"/>
<weight name="indicator" value="0.373"/>	<weight name="stimulus" value="0.059"/>
<weight name="frequent" value="0.373"/>	<weight name="salmon" value="0.050"/>
<weight name="coma" value="0.362"/>	<weight name="felt" value="0.050"/>

$blood\_disease$' and '$symptom/feeling/somesthesia/unpleasant\_person/negative\_stimulus/hurt$' relations of '$symptom$' word.

7. Finally, for the model construction process, first construct the correct controlled natural language text for each identified relation between words as shown in Table 4.6.

Table 4.6: Identified relations of diabetes domain.

Attempto Controlled English (ACE) text	
feeling is a symptom.	high_obesity is a symptom.
somesthesia is a feeling.	over_weight is a symptom.
unpleasant_person feels somesthesia.	edema is a symptom.
unpleasant_person has negative_stimulus.	blood_serum is an indicator.
negative_stimulus is a hurt.	hdl is an indicator.
blood_disease is a symptom.	hyperglycemia is an indicator.
glycemia is glucose_level.	metformin is a medicine.
hyper_tension is bloodpressure.	regular_insulin is a medicine.
weightlost is a symptom.	swallow_feet is a symptom.
frequent_urination is a symptom.	glimepiride is a medicine.
high_thirst is a symptom.	lispro is a medicine.
high_fatigue is a symptom.	glargine is a medicine.

8. Write the correct controlled natural language text into the ACE editor (see Figure 4.2) to construct the domain model as shown in Figure 4.3.

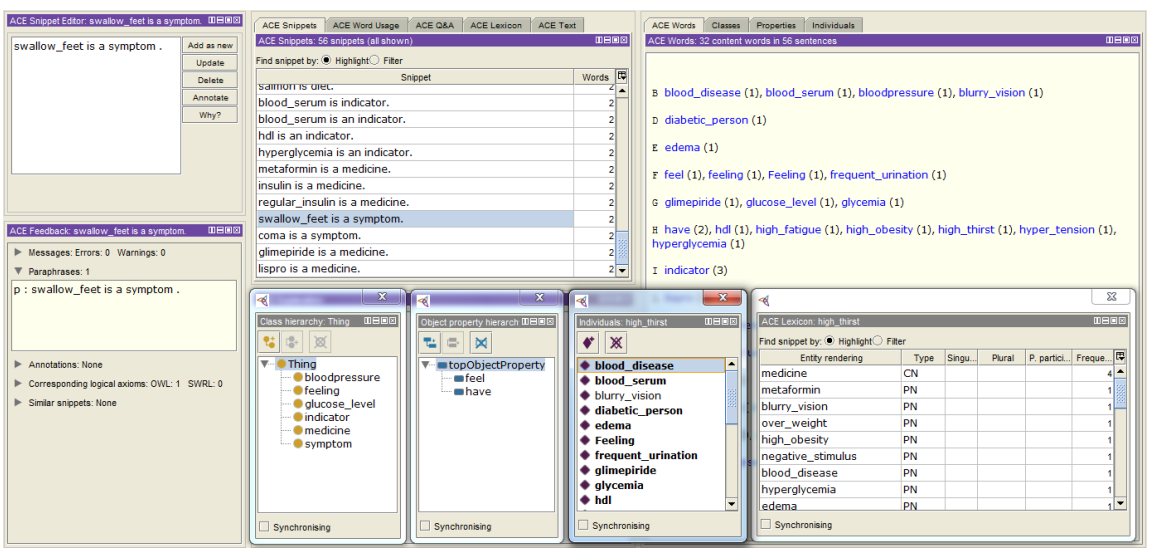


Figure 4.2: Domain model generation through ACE controlled natural language

Once the ontological models are built, they can be accessible and useable by the mapping

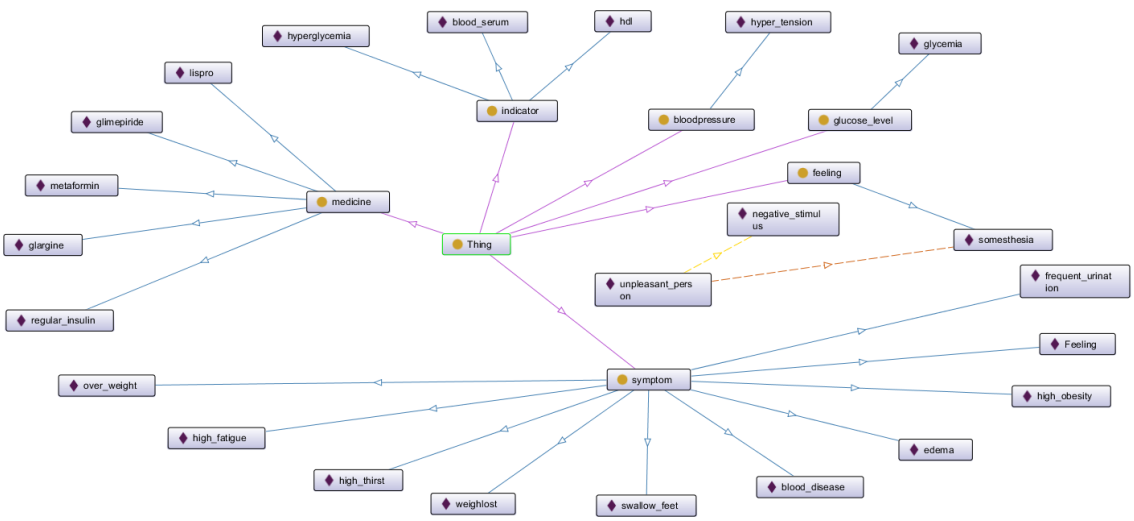


Figure 4.3: A partial view of the domain model

process [189] for decision-support system as well as education, health and wellness applications.

4.4 Conclusions

Declarative knowledge is one of the crucial component in the medical domain and constitutes unstructured representation. In current practices, it is very difficult, time-consuming, and costly to construct machine-readable declarative knowledge from domain documents. In this study, we present a methodology to construct the machine-readable domain knowledge (i.e. structured declarative knowledge) from unstructured text that can serve a broad range of applications such as decision support systems, as well as education, health, and wellness applications. The proposed methodology constructs an ontology from unstructured textual resources in a systematic and automatic way using artificial intelligence techniques with minimum intervention from a knowledge engineer.

Chapter 5

Case-Based Learning

This chapter covers the solution of the third set of research questions/challenges mentioned in the problem statement section of chapter 1. In this chapter, an interactive and effective case-based learning (CBL) approach is presented, which enables the medical teacher to create real-world CBL cases for their students with the support of their experiential knowledge and computer generated trends; review the student solutions, and give feedback and opinions to their students. This approach facilitates medical students to undertake CBL rehearsal with machine-generated domain knowledge support before attending an actual CBL class. In this chapter, a semi-automatic real-world clinical case creation, and case formulation techniques with domain knowledge support are introduced. To automate the proposed approach, an *interactive case-based learning system* (iCBLS) was designed and developed. To evaluate the proposed CBL approach, two studies were performed. The proposed approach was evaluated under the umbrella of the context/input/process/product (CIPP) model and achieved a success rate of more than 70% for student interaction, group learning, solo learning, and improving clinical skills. To exploit the IoT infrastructure for supporting flipped case-based learning in the cloud environment with state-of-the-art security and privacy measures, this chapter also presents an ***IoT**-based **F**lip **L**earni**n**g **P**latform*, called IoTFLiP and working scenario for the case-base flip learning using IoTivity.

5.1 Introduction

Medical education is an active area of research and has undergone significant revolution in the past few decades. In health education, the purpose of medical education programs is to: (1) develop educational leaders, (2) change the learners' knowledge, skills, or attitudes, and (3) improve educational structures [20]. Various teaching methodologies have been introduced in professional

health education [190], where active learning has gained a lot of attention around the world [22]. In active learning, instructions are given to students to actively engage them [23]. Case-Based Learning (CBL) is an active learning approach, which provides favorable circumstances to students in order to explore, question, discuss and share their experiential knowledge for improving their practical intelligence [22]. The term CBL was introduced in the medical area in 1912 [24] and proceeds in many forms, from simple hands-on, in-class exercises to semester long projects and/or case studies [25]. It focuses around clinical, community or scientific problems. According to McLean [24], "CBL is a tool that involves matching clinical cases in health care-related fields to a body of knowledge in that field, in order to improve clinical performance, attitudes, or teamwork".

The CBL approach is one of the successful approaches in student-based pedagogy and it is a widely used approach in various health-care training settings around the world [26–33]. This approach is used in different fields of medicine, namely medicine, dentistry, pharmacology, occupational and physical therapy, nursing, allied health fields, and child development. Similarly, it has been used in clinical as well as non-clinical courses such as nursing courses, adult health, mental health, pediatric, and obstetrical nursing courses, pathophysiology, statistics, law, school affairs, physics education, and research [22, 34, 35]. In addition, this approach has been utilized in various departments such as medical education, information technology, and quality improvement [24], and has also been practiced in rural as well as underserved areas [24]. These findings validate that CBL is used throughout the world across multiple fields, and is considered to be effective for medical and health profession's curricula [24].

In CBL practice, the clinical case is a key component in learning activities, which includes basic, social, and clinical studies of the patient [36]. In the medical domain, the clinical case provides a foundation to understand the situation of a disease and in recent trends; the real-life clinical case(s) are more emphasized for the practice of medical students [37–39]. In medical education, these cases enable the students to use their experiential knowledge to interpret them easily [22]. In the medical area, CBL facilitates students to learn the diagnosis and management of clinical cases [24], and prepares the participants to practice basic primary care and in critical situations [40]. The CBL approach promotes learning outcomes and builds confidence in students while

they are making decisions to practice in real life [30,41]. According to Thistlethwaite [36], "CBL promotes learning through the application of knowledge to clinical cases by students, enhancing the relevance of their learning and promoting their understanding of concepts". CBL is also known to be an effective learning approach for a small group of medical students at undergraduate, graduate, postgraduate education levels as well as for professional development [24,36,37,42,43].

Besides the benefits of CBL approach, there are also a few shortcomings of this approach. For example, in professional education for health and social care domains, students feel that classroom CBL activities require a significant amount of time [44]. Sometimes, students feel uncomfortable while participating in group learning activities and they prefer to work alone [45]. Normally, formal learning activities are performed without a real patient case [36], where interactions are often unplanned and rely on the goodwill of patients. In the specialized literature, medical education programs are considered to be complex due to their diverse interactions amongst participants and environments [20]. Discussion-based learning in a small-group like CBL, is considered to be a complex system [46]. In small-groups, multiple medical students are interacting and exchanging information with each other, where each student is also a complex system [47]. In health care professional education, students have to tackle uncertain situations due to the accumulation of diverse problems [48]. In such situations, everyone has their own judgment, opinion, and feedback and will consider this integral as well as appropriate for that situation. In such situations, experiential knowledge (EK) is thought-out as a resource [48], which can facilitate and provide lived knowledge to students. According to Willoughby [49], "Experiential knowledge is a knowledge of particular things gained by perception and experience". Experiential knowledge enables the individuals to capture practical experience for problem solving. It is considered a valuable resource for enhanced individual participation and user empowerment [48].

For problem-based learning, both human and computer can play a key role in the medical domain. Both of these have their own strengths and weaknesses [50,51]. For example, (1) human judgment is considered credible, (2) a human have common sense and can determine new rules 'off the shelf', (3) a human can easily identify trends or abnormality in visualization data. However, a human (1) cannot accomplish complex computation decisions, (2) cannot perform fast reasoning computations, (3) easily gets tired and bored. These weaknesses of humans can be mitigated by

collaborating with a computer. A computer has advantages over a human for these weaknesses. A computer can perform complex computation decisions, supported by fast reasoning computation, and does not tire.

Being a human, students are easily tired or bored, and tend to choose computer-based cases [36, 52] and opt for web-based cases as compared to lectures for their learning [53, 54]. Additionally, more attention is given to online/web-based learning environments [36]. In order to support the learning outcomes of students, a plethora of web-based learning systems have been developed [55–64]. A review of the literature shows that these systems either do not support computer-based interactive case authoring as well as its formulation, or without the support of acquiring real-world CBL cases or do not provide feedback to students. Currently, less attention is given to fill the gaps between human-based and computer-based learning.

Case-Based Learning (CBL) has become an effective pedagogy for student-centered learning in medical education, which builds its foundation on accumulated patient cases. Flip learning and Internet of Things (IoTs) concepts have gained much attention in recent years. These concepts with CBL can improve learning capabilities by providing real and evolutionary medical cases. The concepts also enable students to build confidence in decision-making, and to enhance teamwork environment efficiently.

Recent trends show that increasing attention is being paid to flipped learning approaches for boosting learning capabilities [145,155]. Currently, CBL is typically performed without exploiting the advantages of the flipped learning methodology, which has significant evidence supporting it over traditional learning methods [55,145,146,157]. As defined by Kopp [156], "Flipped learning is a technique in which an instructor delivers online instructions to students before and outside the class and guides them interactively to clarify problems. While in class, the instructor imparts knowledge in an efficient manner".

In order to support healthcare improvement, much work has been done to acquire information through IoT devices. However, there is still a lack of systems and frameworks to efficiently exploit IoT data and use it for the purpose of extracting knowledge, creating knowledge with partial involvement of the field expert, and using the acquired knowledge for providing real-time patient care and treatment. When designing any system, keeping the privacy of information,

providing on-demand services, and knowledge sharing among organizations are important parameters [145, 191]. For knowledge creation and acquisition, various learning models exist that need to be used for the real-time extraction of meaningful information from IoT devices and to make it shareable among caregivers, patients, and doctors/experts [136, 192]. Currently, the CBL lacks a development mechanism for real-world clinical cases using IoT infrastructure, and there is need to exploit existing IoT resources and infrastructure for boosting medical education. Very little attention is given to the development mechanisms of real-world clinical cases and most of the stakeholders, including learners, teachers, administrators, and other health professionals are interested in change [20].

Keeping in view all aforementioned facts, we focused on designing and developing an interactive computational e-learning platform by using CBL concepts so that medical students are can be provided with the following learning activities: (1) practicing real-world case(s) before and outside the class to determine the treatment of patients in an easy to use manner, (2) identifying the components of a medical chart (such as demographics, chief complaint, medical history, etc.) from a given clinical case, (3) constructing appropriate interpretations about a patient's problems to create a significant medical story using identified components within the context of his or her life, and (4) implanting clinical knowledge to obtain professional experience for effective learning purposes. In order to achieve these goals and expectations, this study was undertaken with the following objectives: (1) create a real-world online and computer-based clinical case using an experiential knowledge (see Sections 5.2.2 and 5.3.2); (2) identify basic science information relevant to patient data for their practice with a support of machine-generated domain knowledge(see Sections 5.2.3 and 5.3.3); and (3) design an IoT-based platform that can be used for medical, as well as other domains for effective and enriched learning (see Section 5.5).

In this chapter, an *interactive Case-Based Learning System* (iCBLS) based on the current CBL practices in the *School of Medicine, University of Tasmania, Australia* was designed and developed. The proposed iCBLS provides features such as: an online learning environment, interactiveness, flexibility, display of the entire collection of data at one place, a paging facility, and support for in-line reviewing to edit and delete the displayed data. The iCBLS consists of three modules: (i) *system administration* (SA), (ii) *clinical case creation* (CCC), and (iii) *case formulation* (CF).

The SA module manages multiple types of users and it maintains the hierarchy of courses, their units, and clinical cases for each unit. Similarly, the CCC module is based on an innovative semi-automatic approach that consists of three steps. First, graphs are generated from a patient's vital signs with a single click. In the second step, a clinical case is generated automatically by integrating basic, history, and vital signs information. Finally, in the third step, the medical teacher utilizes his/her experiential knowledge and refines the generated case in order to create the real-world clinical case. The CF module is based on identification of the medical-chart's components in order to formulate the summaries of CBL cases through the intervention of medical students' as well as teachers' knowledge, as well as the provision of feedback from the teacher. In addition, the CF module enables the students to practice real-world case(s) with machine-generated domain knowledge support before and outside the class.

This study also introduced an ***IoT**-based **F**lip **L**earn**i**ng **P**latform*, called IoTFLiP, that integrates the features of existing IoT resources. The IoTFLiP exploits the IoT infrastructure to support flipped case-based learning in the cloud environment with state-of-the-art security and privacy measures for potentially personalized medical data. It also provides support for application delivery in private, public, and hybrid approaches. Due to the low cost, reduced sensing devices' size, support of IoTs, and recent flip learning concepts can enhance medical students' academic and practical experiences. To demonstrate the working scenario of the proposed IoTFLiP platform, a real-time data through IoTs gadgets is collected to generate a real-life situation case for a medical student using iCBLS.

The key contributions of this research are as follows:

1. This work focuses on developing an intelligent computational e-learning platform for CBL in medicine that enriches and enhances the learning experience for medical students.

2. The chapter shows the design and development of an interactive CCC module that supports an innovative method to real-world clinical case creation using a semi-automatic approach.

3. The chapter shows the design and development of an interactive CF module that provides a flexible case formulation environment.

This chapter is organized as follows: Section 5.2 covers the methodology of the proposed CBL

approach; the iCBLS along with a case study scenario is discussed in Section 5.3. Section 5.4 provides the details of evaluations performed along with results, while Section 5.5 presents the IoTFLiP architecture and working scenario for the case-base flip learning using IoTivity. Section 5.6 discusses the significance, challenges and limitations of the proposed system. Section 5.7 concludes the chapter with a summary of the research findings.

5.2 Materials and Methods

To develop an interactive CBL system to prepare medical students for their real-world clinical practice before and outside the class, this section describes the architecture of the proposed system and detailed methodologies used for *Clinical Case Creation* and *Case Formulation* modules.

5.2.1 Proposed system architecture

The functional architecture of the proposed system is described as shown in Figure 5.2, which consists of four modules, namely *Graphical User Interface*, *System Administration*, *Clinical Case Creation*, and *Case Formulation*. Three types of users - *administrator*, *medical teacher*, and *medical students* interact with the iCBLS through the *Graphical User Interface* module. The functionalities of the iCBLS are illustrated in Figure 5.1.

Using this tool, the *Administrator* manages courses by specifying course details, modules, and allotments. The *Medical teacher* manages CBL cases and their model solutions, evaluates student solutions, and provides feedback to students. The *Medical student* formulates case summaries (history, examination, and investigations) with the help of domain knowledge to solve the CBL case, views other available solutions, and receives feedback from the medical teacher. The detailed role description of each user is shown pictorially in Section 5.3 Figure 5.6.

The functionality of each module is described as follows.

The functionality of the Graphical User Interface module

The *Graphical User Interface* module provides an interface to all users to interact with the other three aforementioned modules. This module provides a flexible environment by facilitating: (1) an

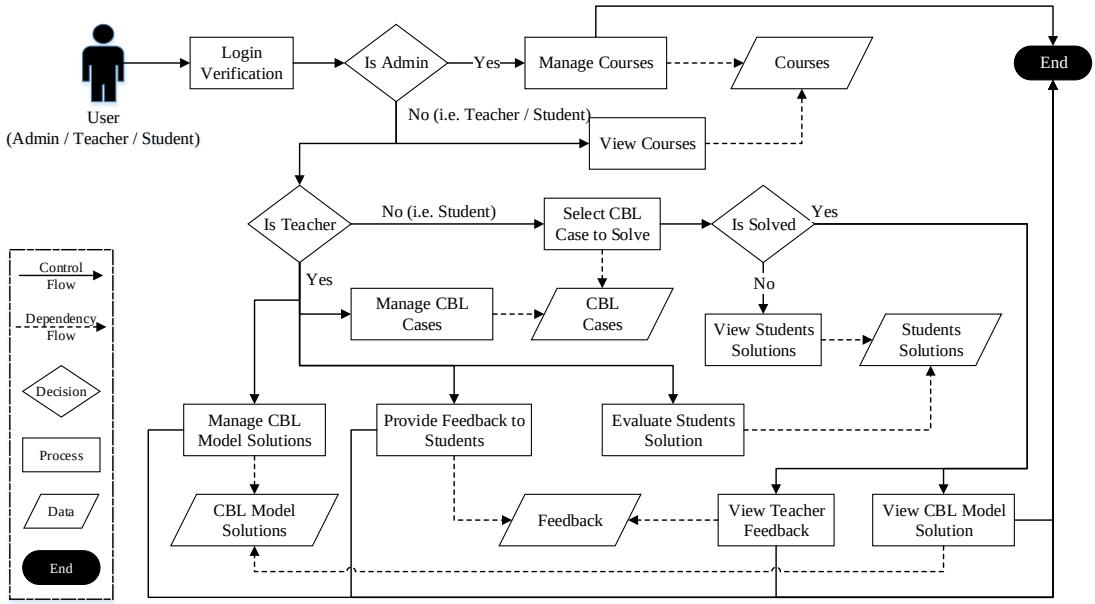


Figure 5.1: iCBLS flow chart

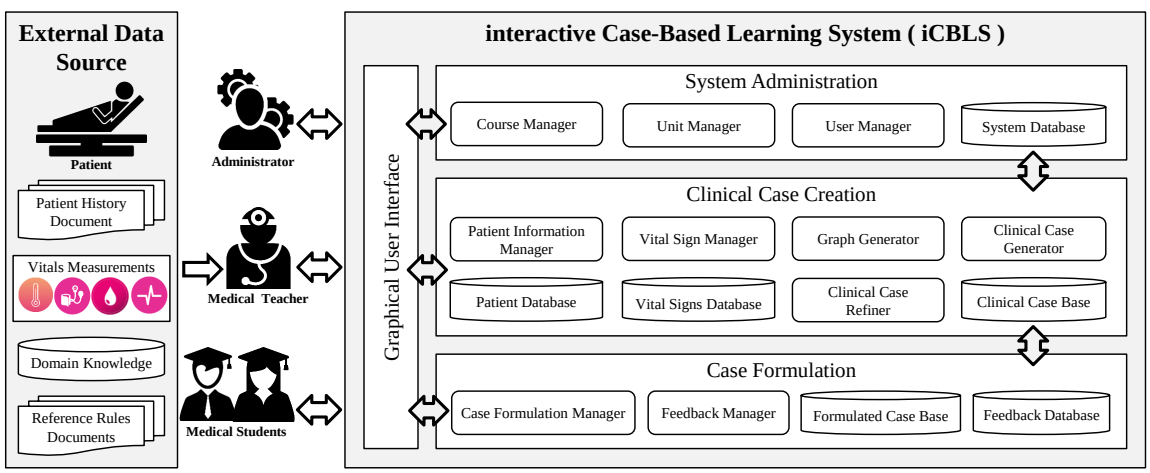


Figure 5.2: Functional architecture of the iCBLS

easy and user-friendly paging facility, (2) a display of the entire collection of data, and (3) support for inline editing to edit and delete the displayed data.

The functionality of the System Administration module

The iCBLS provides support for managing numerous courses, where each course consists of multiple units e.g. *'CBL Cases'* is one course that includes two units, namely *'Fundamentals of Clinical Science'* and *'Functional Clinical Practice'*. Multiple students are able to enrol in each unit. The *administrator* is assumed to be the coordinator that manages the CBL administration and interacts with *System Administration* module, as shown in Figure 5.2. The administrator manages the hierarchy of courses, their units, and users' relations with units by using the *Course Manager*, *Unit Manager*, and *User Manager* components to store the information into the *System Database*. Moreover, the administrator manages two types of users, namely *medical teacher* and *medical student*. In addition to this, the administrator assigns the courses' units to the individual medical teacher and enrols the medical students to each unit. All aforementioned information is stored and managed in *System Database*. The detailed flow diagram of *System Administration* module is described and shown in Figure 5.3.

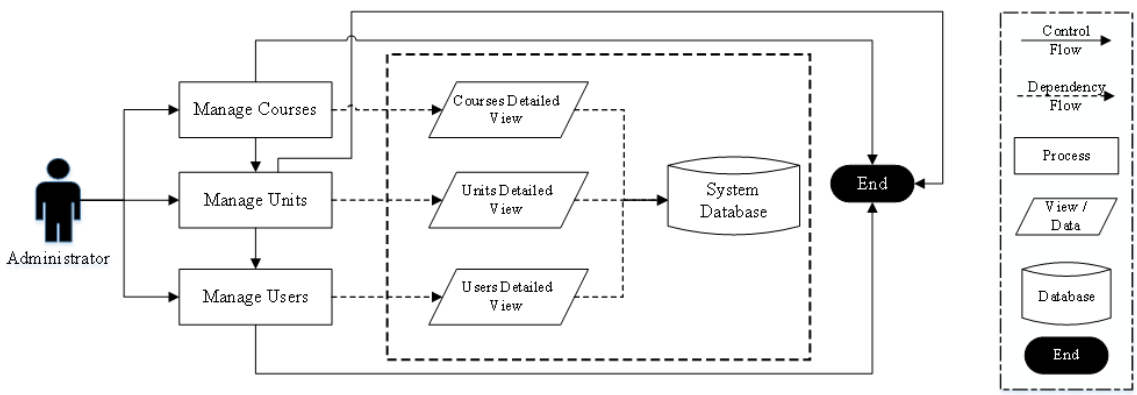


Figure 5.3: Flow diagram of system administration module

The functionality of the Clinical Case Creation module

The *Clinical Case Creation* module is used to create real-world clinical cases. The *medical teacher* who interacts with this module is assumed to be a medical expert that interacts with patients either at private clinics or at hospitals. This module consists of five components as follows: *Patient Information Manager* for managing patient's basics and history information, *Vital Sign Manager* for

managing the categories and measurement information of patient's vital signs, *Graph Generator* for generating and visualizing vital signs, both individual and average values, *Clinical Case Generator* for auto-generating a clinical case by integrating basic information, patient history, vitals' (a.k.a. vital signs) information and finally, *Clinical Case Refiner* for refining the auto integrated case. This module also requires real-world patients' and vital signs reference rules' information (see Table 5.3 in Section 5.2.2) that is obtained from *External Data Source*, which includes *Patient*, *Patient History Document*, *Vitals' Measurements*, and *Reference Rules' Documents* as data sources.

The functionality of the Case Formulation module

The *Case Formulation* module is intended for (1) identifying the components of a medical chart (such as demographics, chief complaint, medical history etc.) from a given clinical case, (2) allowing the medical students to write their observations for each component by utilizing domain knowledge and finally, (3) receiving feedback from the medical teacher. This module helps medical students to understand the causes of patient behaviors and symptoms, to formulate summaries of CBL cases and to get feedback about self-formulated cases from their medical teacher. The *medical students* as well as *medical teacher* interact with this module. This module is comprised of two components: *Case Formulation Manager* for managing formulated cases that are created by students as well as teachers, *Feedback Manager* for providing teachers' feedback to individual students. This module also requires domain knowledge that is obtained from *External Data Source*, which includes *Domain Knowledge* as knowledge source.

5.2.2 Clinical case creation methodology

This section briefly describes the procedure for creating a real-world clinical case in the proposed system (iCBLS) using an innovative semi-automatic approach as shown in Figure 5.4. As mentioned in some studies [59, 193, 194], a clinical case is generally written as a problem which includes basic personal information, reported complaints, history and physical examinations, imaging studies, vital signs, clinical signs and symptoms, laboratory results, findings, diagnoses, discussions, comments, and learning points. In this study, *patient basic information*, *patient history*,

and *vital signs* information are considered as components of a real-world clinical case.

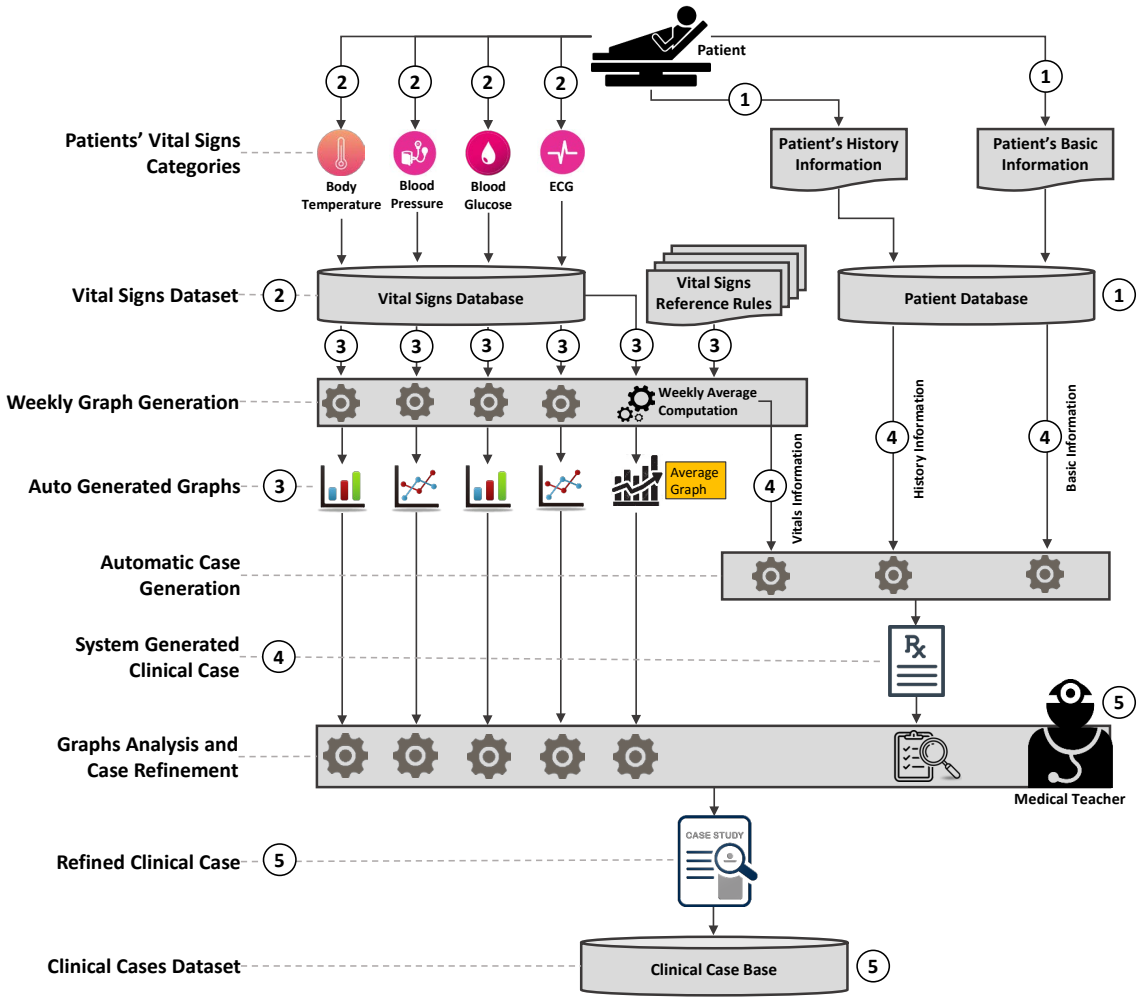


Figure 5.4: Real-world clinical case creation steps

Five steps are involved for real-world clinical case creation, which are shown in Figure 5.4. First, the medical teacher converses with the patient and records the patient's basic information such as the patient's name, gender and age. Following this, the patient's history is recorded, this covers medical history, family history, symptoms review and food habits etc. This information is stored in the *Patient Database*. In the second step, the patient's vital signs are recorded in the *Vital Signs Database*. In this study, body temperature, blood pressure, blood glucose, and heart rate vital signs categories are considered, which are helpful for patient treatment and disease

diagnosis [143, 195]. These vital signs are measured by traditional devices such as thermometers for body temperature, sphygmomanometers for blood pressure, blood glucose meters for blood glucose, and stethoscopes for heart rate. However, this vital signs information can also be captured with the help of *RFID* technology and sensors through wearable devices [196, 197]. Multiple IoT gadgets are available to measure these vitals; they are presented in Table 5.1.

Table 5.1: IoT gadgets for collecting vital signs.

Vital sign	**Available devices**
1. Blood Glucose	iHealth's Blood Glucose Monitor, iHealth Align, iBG Star, etc
2. Blood Pressure	iHealth Wireless Blood Pressure Monitors, Omron BP786, Microlife WatchBP home A, QardioArm Blood Pressure Monitor, etc
3. Heart Rate	LG gear watch, Wellograph, Polar V800, Mio LINK, Epson Pulse Watch, Spree Headband, etc

Weekly graphs are generated from a patient's vital signs data in the third step. For visualization, line and bar graphs are used, and the weekly average value for each vital sign's category is computed and a separate graph is generated. In addition, reference ranges, as defined in Table 5.3, for each vital sign category, are shown in each graph in order to assist with interpretation. In the fourth step, the patient's basic information along-with history and vital signs' data are integrated to create the system-generated clinical case. Finally, in the fifth step, the medical teacher visualizes the system-generated case as well as all auto-generated graphs. After visualization and analysis, the medical teacher refines the auto-generated case as shown in Table 5.2 and stores this in the *Clinical Case Base* for medical students' practice.

The aforementioned process of real-world clinical case creation for multiple patients is briefly described in Algorithm-5. This algorithm takes *basic information* (i.e., BI), *patient's history* (i.e., PH), and *vitals' information* (i.e., VI) as input and then sequentially passes through mandatory steps to create the multiple real-world clinical cases. The output of this algorithm is used as input for Algorithm-6, which is described in following subsection.

Algorithm 5: Creation of Real-World Clinical Case($D = BI, PH, VI$)

Input : $D = BI, PH, VI$: Input dataset (basic information, patient history, vitals' information)
Output: $CC-$ Real-world clinical case

/* $D = p_1, p_2, p_3, ..., p_n$ where D represents data for n patients */;
for $\forall\, p_i \in D$ **do**
  /* Get the basic information e.g. gender, age; and patient's history e.g. medical history, family history, symptoms for each patient p_i */;
  $BI_i \leftarrow getBasicInformation(D.p_i)$;
  $PH_i \leftarrow getPatientHistory(D.p_i) : p_i = ph_1, ph_2, ph_3, ..., ph_n$;
  /* Vitals' information VI_i consists of vital's category and its measurements. Firstly, select the vital sign category e.g. systolic blood pressure for each patient p_i */;
  selectVitalSign(VS) : $VS = vs_1, vs_2, vs_3, ..., vs_n$;
  for $\forall\, vs_j \in VS$ **do**
    $M_k = m_1, m_2, m_3, ..., m_n$ // no. of measurements for vs_j ;
    for $\forall\, m_i \in M_k$ **do**
      /* Get vital sign measurements for each vital sign category vs_j */;
      $m_i \leftarrow getVSMeasurement(D.p_i, vs_j)$;
    end
    /* Compute the average values for each vital sign category vs_j */;
    $vsmAvg_i \leftarrow \sum_{i=1}^{size(M_k)} m_i / size(M_k)$;
    /* Plot the individual and average graph for each category vs_j */;
    $trendgraph \leftarrow plotVSMeasurementGraph(D.p_i, vs_j)$;
    $meangraph \leftarrow plotVSMeasurementAverageGraph(vsmAvg_i)$;
  end
  /* Generate the case by integrating BI_i, PH_i, and $vsmAvg_i$ for each patient p_i */;
  $SGC_i \leftarrow generateCase(BI_i, PH_i, vsmAvg_i)$;
  /* Analyze the patient auto generated graphs */;
  $AG_i \leftarrow analyseGraphs(meangraph, trendgraph)$;
  /* Refine the generated case based on the personal knowledge and graphical analytic */;
  $CC_i \leftarrow refineCase(SGC_i, AG_i)$;
  return CC_i : *clinical case*
end

Table 5.2: Example real-world CBL case.

Case Outline
Mr. X, a 65 years old corporate sector worker, came to a medical expert with a few complaints. He said that he is providing financial consulting to various clients. He added that his office hours are 8:30 am to 6:00 pm. Since his job is related to office work, he has little physical activity. He used to drink regularly and likes to eat fatty and oily foods. He says he has become exhausted very easily for the last few weeks. He feels fatigued and breathless after walking only 100 m. He reported experiencing blurry vision and weight-loss. He said that he has never experienced these problems before. He was on no medications. He was 183 cm tall and weighed 196 lbs. He had a family history of hypertension and hyperglycemia. The expert was worried about his health and cautioned him to be more conscious of his health. In order to observe his vital signs, the expert suggested that he use wearable devices to measure his blood pressure, glucose level, and heart-rate. On examination, the results were: Systolic Blood Pressure = 135.24 mmHg, Diastolic Blood Pressure = 89.33 mmHg, Glucose Level in fasting = 145.43 mg/dL, Glucose Level in random = 247.36 mg/dL, Heart Rate = 90.14 bpm, Body Temperature = 98.69

5.2.3 Case formulation methodology

Case formulation is a commonly taught clinical skill and it is the foundation for balanced treatment planning that develops with practice and clinical experience [202–204]. In case formulation, clinicians determine the treatment of their patients and treatment of each particular patient is different from that of other patients [202]. Case formulation has a vital role in clinical decision-making [203] which is emphasized in many published documents [204]. It is frequently emphasized to practitioners to develop professional competency in case formulation for their professional training as well as continuing medical education. Case formulation has multiple definitions and contents in various approaches [202]. As described by Godoy and Haynes [204], ”Case formulation is an individualized integration of multiple judgements about a patient’s problems and goals, the casual variables that most strongly influence them, and additional variables that can affect the focus, strategies, and results of treatment with a patient”. Formulating a clinical case involves constructing appropriate interpretations about a patient’s problem to create a significant medical story within the context of his or her life [203].

As case formulation has multiple definitions, in this study case formulation means identification of a medical-chart’s components from a given clinical case and then writing personal observations for each component. As mentioned in some studies [59,205], demographics, chief complaint,

Table 5.3: Vital signs reference ranges with interpretations

Vital Sign	**Categories**	**Reference Range**	**Interpretation**
Blood Pressure(*mmHg*) [198]	Systolic Blood Pressure (SBP)	SBP ≤ 119	normal
		$120 \leq$ SBP ≤ 139	prehypertension
		$140 \leq$ SBP ≤ 159	hypertension stage 1
		$160 \leq$ SBP ≤ 180	hypertension stage 2
		SBP ≥ 181	hypertensive crisis
	Diastolic Blood Pressure (DBP)	DBP ≤ 79	normal
		$80 \leq$ DBP ≤ 89	prehypertension
		$90 \leq$ DBP ≤ 99	hypertension stage 1
		$100 \leq$ DBP ≤ 110	hypertension stage 2
		DBP ≥ 111	hypertensive crisis
Blood Glucose(*mg/dL*) [198, 199]	Fasting Blood Glucose (FBG)	FBG ≤ 69	hypoglycemia
		$70 \leq$ FBG ≤ 99	normal
		$100 \leq$ FBG ≤ 126	pre-diabetic
		FBG ≥ 127	diabetic
	Random Blood Glucose (RBG)	RBG ≤ 139	normal
		$140 \leq$ RBG ≤ 199	pre-diabetic
		RBG ≥ 200	diabetic
Heart Rate(*bpm*) [200, 201]	Resting Heart Rate (RHR)	RHR ≤ 59	bradycardia
		$60 \leq$ RHR ≤ 100	normal
		RHR ≥ 101	tachycardia
	Sleeping Heart Rate (SHR)	$40 \leq$ SHR ≤ 50	normal
	Irregular Heart Rate (IHR)	IHR == true	arrhythmia
Body Temperature($^\circ F$) [201]	Body Temperature (BT)	$97.7 \leq$ BT ≤ 99.5	normal

medical history, habits, family history, medicines, allergies, physical exam, tests ordered, initial diagnosis, differential diagnosis, test results, final diagnosis, treatment, recommendations, and prognosis are considered as the components of medical-chart.

As described in Figure 5.5, the authorized medical student views the allotted courses. For case formulation, the student first selects the CBL case. After clinical assessment of the selected case, the student conceptualizes the information and identifies the components of the medical chart. Following this, the student then gets the domain knowledge to record his/her personal observations.

During the formulation process, the student can also get help from available formulated cases that are completed by other medical students. After case formulation, students get feedback from their teacher in order to improve their concepts and knowledge.

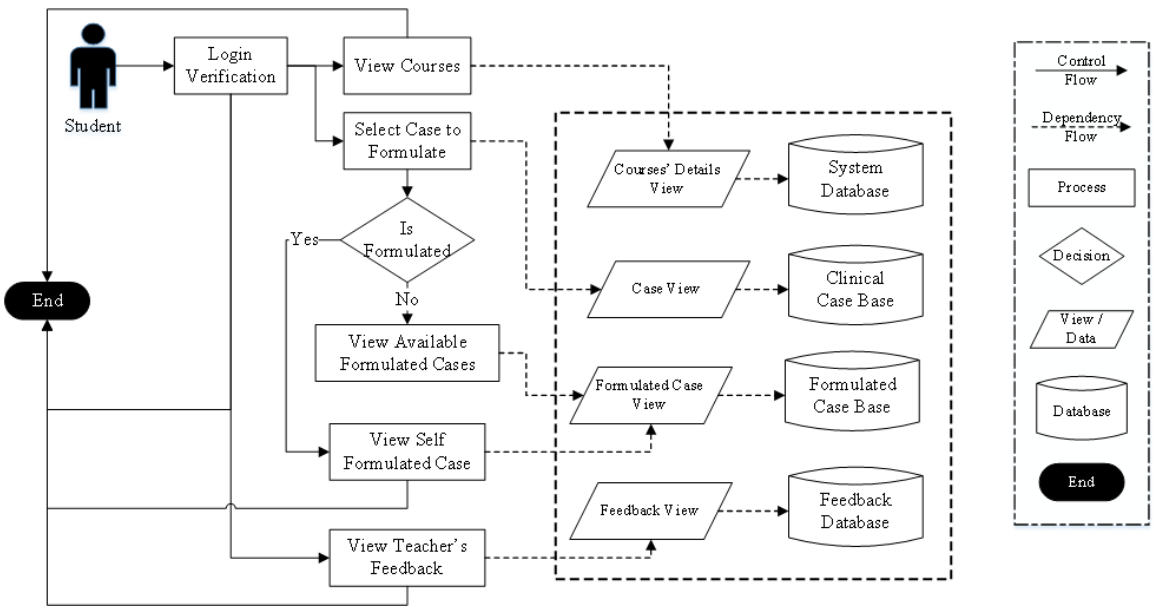


Figure 5.5: Flow diagram of case formulation module

The process of case formulation briefly is described in Algorithm-6. This algorithm takes a clinical case (i.e., CC) as an input and sequentially passes this through mandatory steps to resolve the clinical case in terms of creating a medical-chart.

5.3 Simulation of iCBLS

The design of the iCBLS is based on the current CBL practices whose working principle is explained with the help of a *Glycemia* case study. Using this system, the medical teacher can create cross-domain clinical case(s) and then students can formulate summaries of cases before attending the actual CBL class for practice. Moreover, the teacher can review the students' formulated summaries and can provide feedback on their solutions. The output of this system is the course's information, real-world cases, health records, formulated cases, and the teacher's feedback.

The iCBLS is an interactive as well as flexible online software system, which manages mul-

Algorithm 6: Case Formulation($D = CC[Ref.$**Algorithm 1**$]$)

Input : $D = CC$: Input dataset (clinical case)
Output: $CF-$ Case Formulation

if $Verify(D)$ **then**
  /* For creating the medical charts, add the components of medical charts e.g. presenting complaints, previous medications for D cases */;
  $MCC \leftarrow addMedicalChartComponent(D)$: $D = mcc_1, mcc_2, mcc_3, ..., mcc_n$;
  for $\forall\, mcc_m \in D$ **do**
    /* Get domain knowledge to add observations e.g. felt fatigue, breathlessness of each chart component mcc_m */;
    $Obs \leftarrow addObservations(mcc_m)$;
  end
  /* Case formulation includes information of medical charts component and observations */;
  $CF \leftarrow caseFormulation(MCC, Obs)$;
  return $CF : case\ formulation$
else
  $Error(message)$;
end

tiple types of users according to their roles and privileges. It has been implemented in *C#* using *SQL Server 2008 R2* and *Bootstrap* as the front-end framework. In this system, nested *GridView* controls are used to manage the hierarchies of courses or cases. Similarly, *Stored Procedures* are created to decrease roundtrip response times and avoid code redundancy, as well as to simplify maintenance and enhancement. Both *GridView* and *Stored Procedure* techniques allow for increased system flexibility.

The role description of this system is shown in Figure 5.6, it depicts types of system users, main options available in iCBLS for each user, and detailed functionalities of each main option.

5.3.1 Case study: Glycemia case

For in-depth study or analysis of real-world or imagined scenarios, the case study is used as a training tool to explain development factors in the case. In this case study, a *Glycemia* patient was monitored regularly, who visits a hospital for clinical check-ups. The medical teacher interacts directly with the patient to obtain his demographics, daily routine activities, medication history (if any), and family history information. The medical expert obtains the patient's basic information and initial history through dialogue and available patient records.

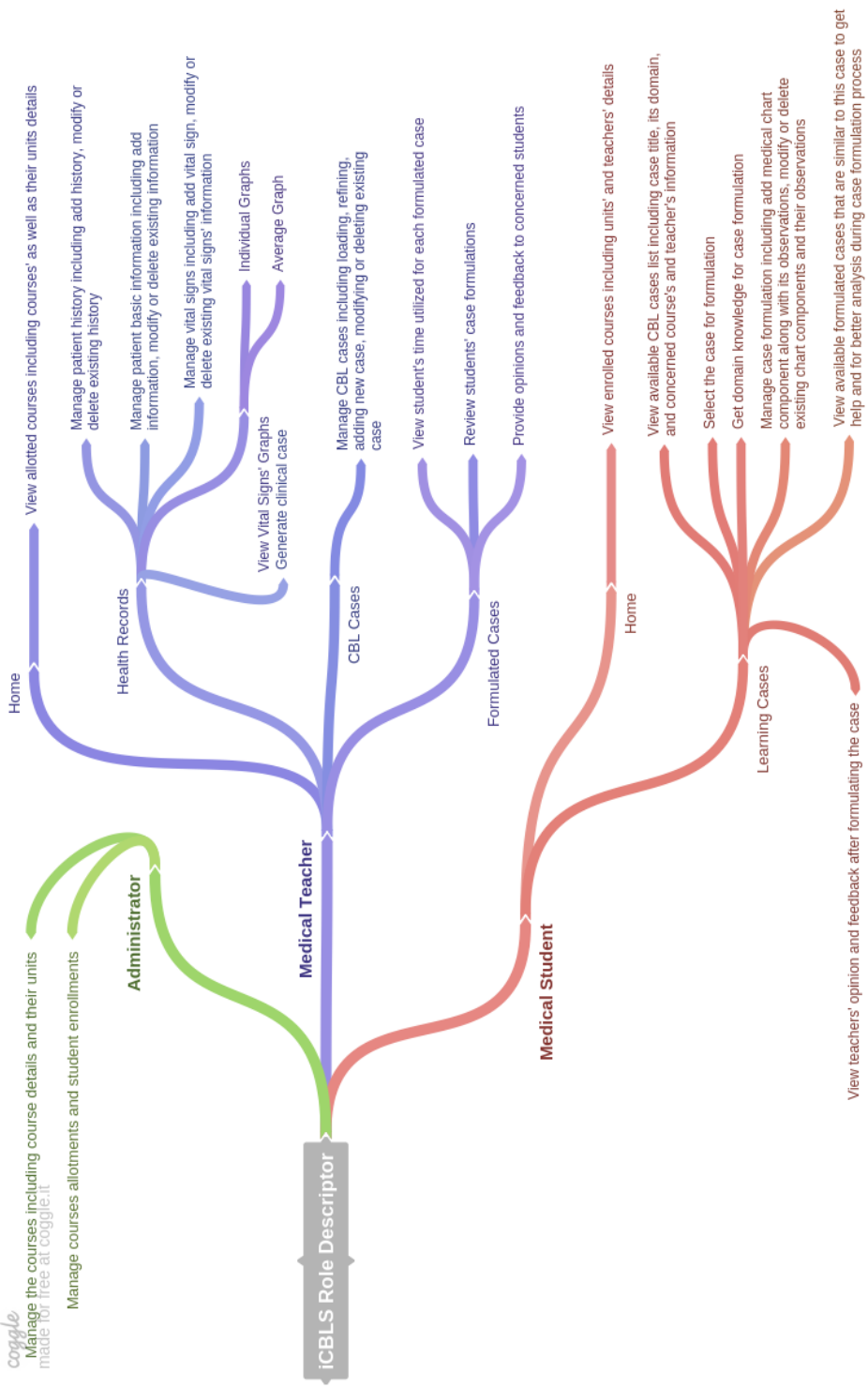


Figure 5.6: iCBLS role descriptor

The medical teacher requires the log of vital signs to understand the severity of disease; therefore, it is advisable that the patient's vital signs such as *body temperature*, *blood pressure*, *glucose level*, and *heart rate* are recorded on a regular basis. The teacher also suggests that the patient's blood glucose level should be monitored in the morning with fasting as well as measured 2 hours after lunch and dinner. The patient then records their vital signs information three times a day for one week, based on the teacher's instructions.

5.3.2 Clinical case creations

The process of real-world clinical case creation is described through the steps that are explained as follows.

Step-1: Record basic information and history information for the patient

In order to execute the scenario for creating a CBL case, the medical teacher uses the patient's basic information e.g. patient name, gender, age. This information is added into the system after clicking the *Add Patient* link as shown in Figure 5.7(1a). After successful addition, the system refreshes the patient pane as shown in Figure 5.7(1b). Similarly, after adding a patient record, the system displays the history pane to enable history details to be added, by clicking the *Add Patient History* link as shown in Figure 5.7(2a). The system then refreshes the history pane as shown in Figure 5.7(2b). Once patient information is added, the teacher can easily modify or delete the record at any time using the *Edit* or *Delete* links as shown in Figure 5.7.

Step-2: Record patient's vital signs information

For inclusion of vital signs information, the medical teacher uses the *Add Vital Sign Info.* link shown in Figure 5.7(3a). After doing this, the system displays the list of vital signs as shown in Figure 5.8(a). The teacher clicks the '+' icon to see a child grid that provides options for adding a vital sign measurement as shown in Figure 5.8(a). In the expanded grid view, the '+' icon is changed to '–' icon. For a better view, a paging concept is also implemented as shown in Figure 5.8(a). The teacher enters the vital signs data into iCBLS. To enter date and time information, the system provides a calendar to the teacher for user-friendliness as shown in Figure 5.8(b). When

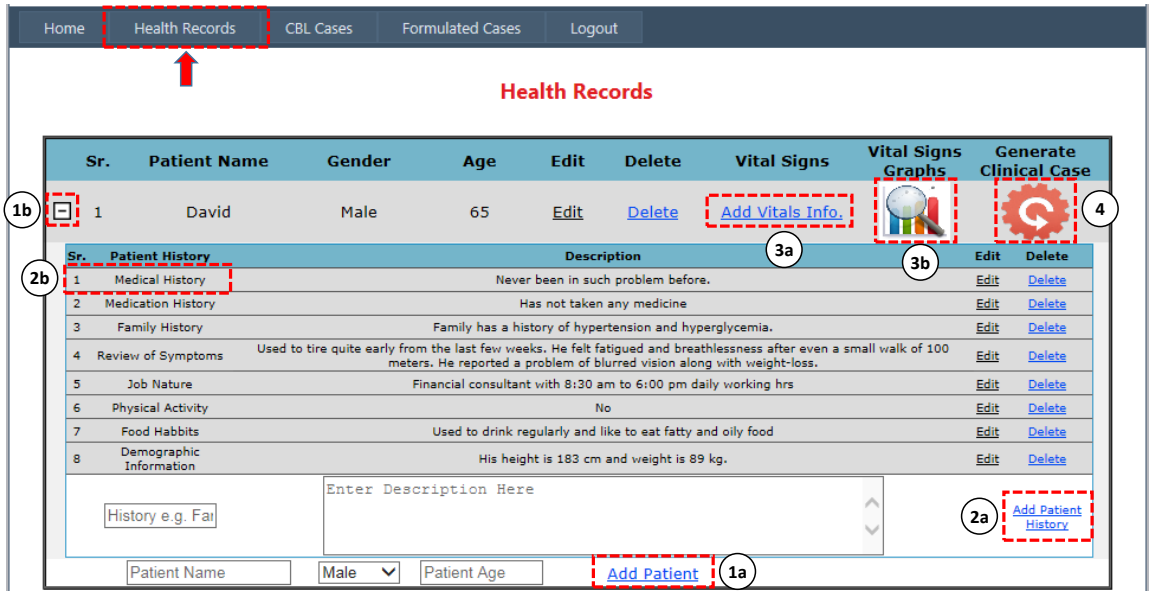
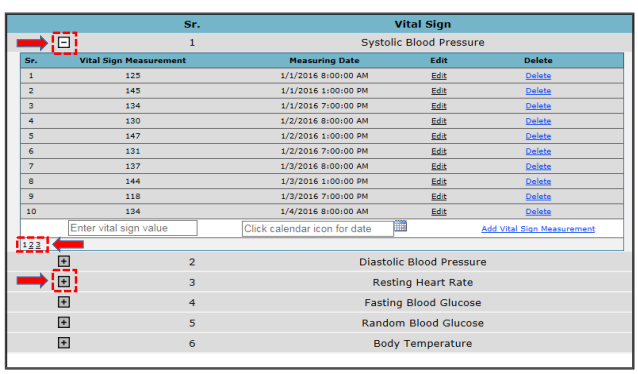
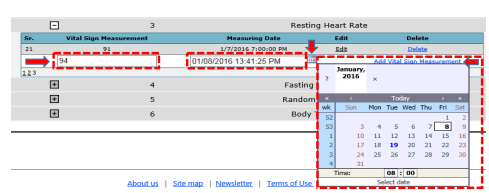


Figure 5.7: Health record management interface

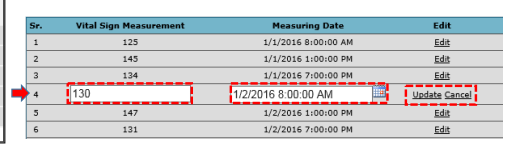
modifying existing measured values, the teacher clicks the *Edit* link. The system then shows the relevant data in an editable form as shown in Figure 5.8(c). After modification, the teacher clicks the *Update* link. The system then updates the existing data and refreshes the grid.



(a) Patient's vital signs measurement information



(b) Adding vital sign measurement value and date



(c) Modifying vital sign measurement value and date

Figure 5.8: Managing vital signs information view

Step-3: Generate and visualize the vital signs graphs

Visualization is the presentation of data in a format, which is easily understandable. It is a key feature used to analyse and interpret measured data. Once the *Vital Signs Graph* link icon, as shown in Figure 5.7(3b) is clicked, the system generates auto-scaled trend charts for each vital sign category using their measured values and then visualizes them as shown in Figure 5.9. Moreover, charts are also auto divided into different areas based on the previously mentioned reference ranges. In Figure 5.9, each vital sign graph is divided into different areas depending on their reference ranges. Each range has its own interpretation in each vital sign category. For example, in Figure 5.9(a), the *Systolic Blood Pressure* (SBP) graph shows three areas having ranges ≤ 119 (Normal Range), $120-139$ (Pre-hypertension), and $140-159$ (Hypertension Stage-1) as defined in Table 5.3. These ranges help medical teachers to analyse and interpret any vital signs trends easily. The system computes the average of each vital sign and generates the average trend chart for each vital sign category as shown in Figure 5.10.

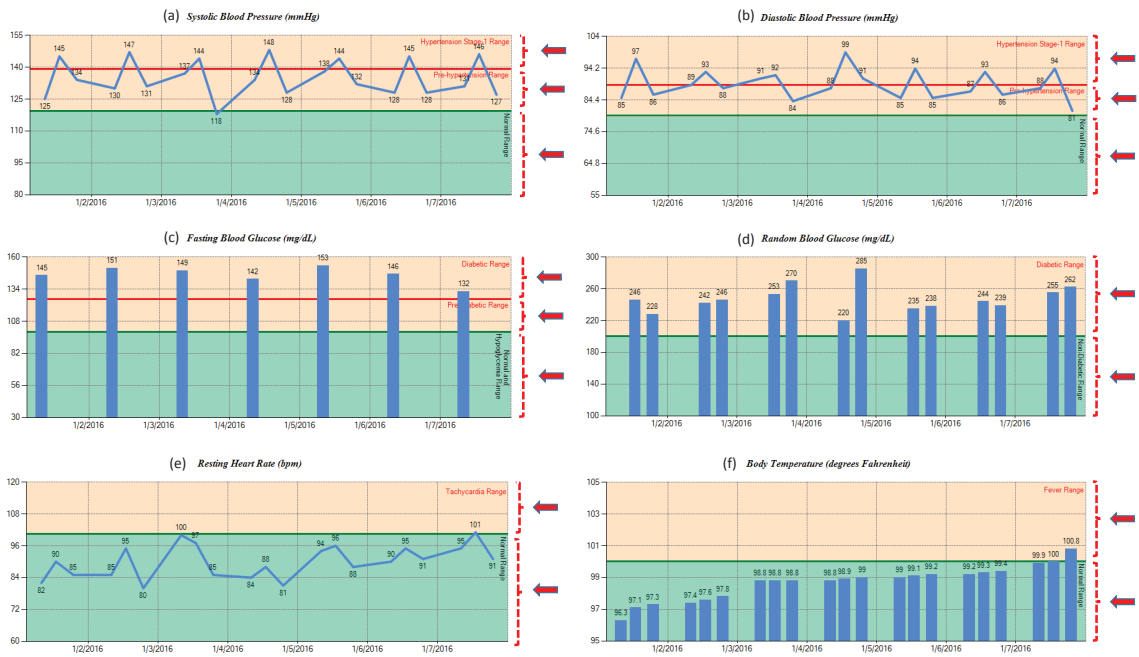


Figure 5.9: Weekly trends of patient's vital signs information

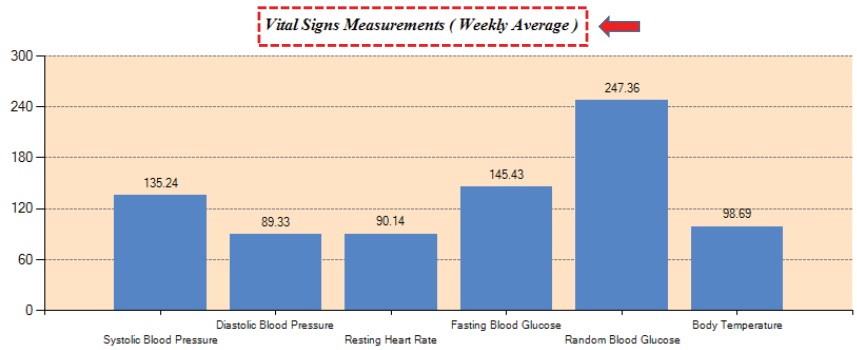


Figure 5.10: Weekly average chart of measured patient's vital signs

Step-4: Generate clinical case

Once the basic information, patient history, and vital signs information are recorded into iCBLS, the system generates the clinical case when the *Generate Clinical Case* link icon is selected as shown in Figure 5.7(4). The system integrates all this information as described in Step-1 and Step-3 to generate the new clinical case labelled (2) that is shown in Figure 5.11.

Step-5: Refine clinical case

After generating a new clinical case, the medical teacher interacts with the iCBLS and loads the system generated case, as shown in Figure 5.11(2), by clicking the *Load Case* link as shown in Figure 5.11(1). Once the case is loaded, the medical teacher enters *Case Title* and selects *Case Domain*, *Unit Title*, and *Difficulty Level* of the case as shown in Figure 5.11(3)-(6). Following this, the teacher utilizes his/her experiential knowledge and enriches the system generated case, as shown in Figure 5.11(7), based on the personal knowledge and graphical trends' information shown in Figures 5.9-5.10. In Figure 5.11, labels 2 and 7 show the comparison between the system generated and teacher-enriched case. After enriching the clinical case description, the teacher clicks the *Add Case* link, as shown in Figure 5.11(8), in order to store newly created CBL case into *Case Base*.

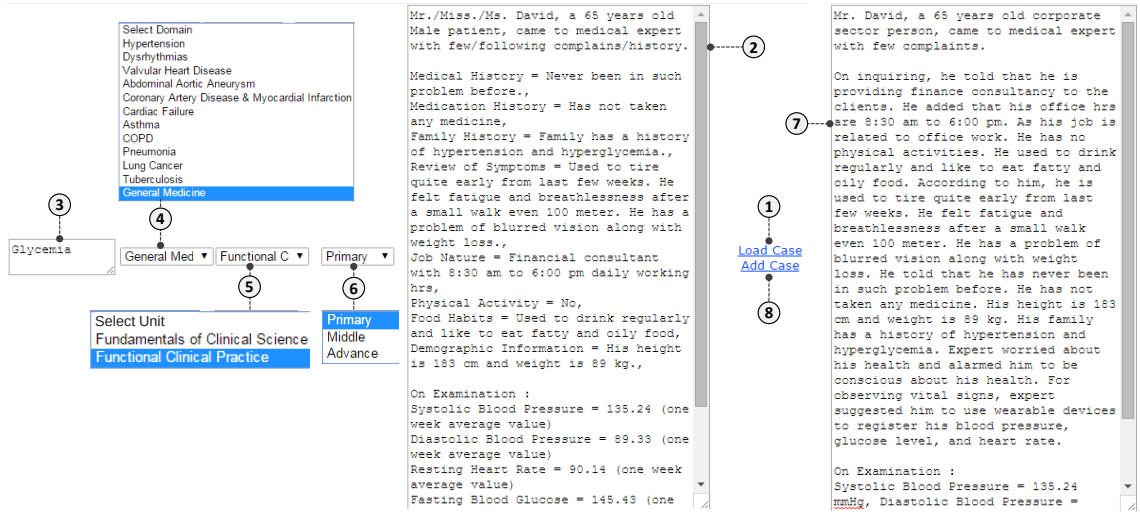


Figure 5.11: Real-world clinical case creation steps

5.3.3 Case formulation

After the medical teacher creates the CBL case, the system automatically updates the list of cases available to students for their practice along with related information. In order to start the case formulation, the student loads the interface, which is shown in Figure 5.12. A timer starts at the back-end of this interface until the submission of this formulation. The timer helps the teacher to assess the future difficulty level of a case for that particular group of students. As depicted in Figure 5.12, the interface is divided into three sections. The first section provides the case description, while the second section shows the medical chart that includes students' entered chart-components such as *Previous Medication* and their observations such as *No medicine mention*. Initially this section is blank. As students add chart components this section updates and expands dynamically. This section also enables medical students to view the domain knowledge to record their personal observations. Finally, the third section shows the list of students who submitted their formulation and solutions for that particular case. After completing the formulation of a CBL case, students submit their data. During the submission process, the system records the total time taken by each student.

Once students have submitted their solutions the teacher reviews the medical chart and analyses student capabilities by considering their submitted solution along with the time taken to con-

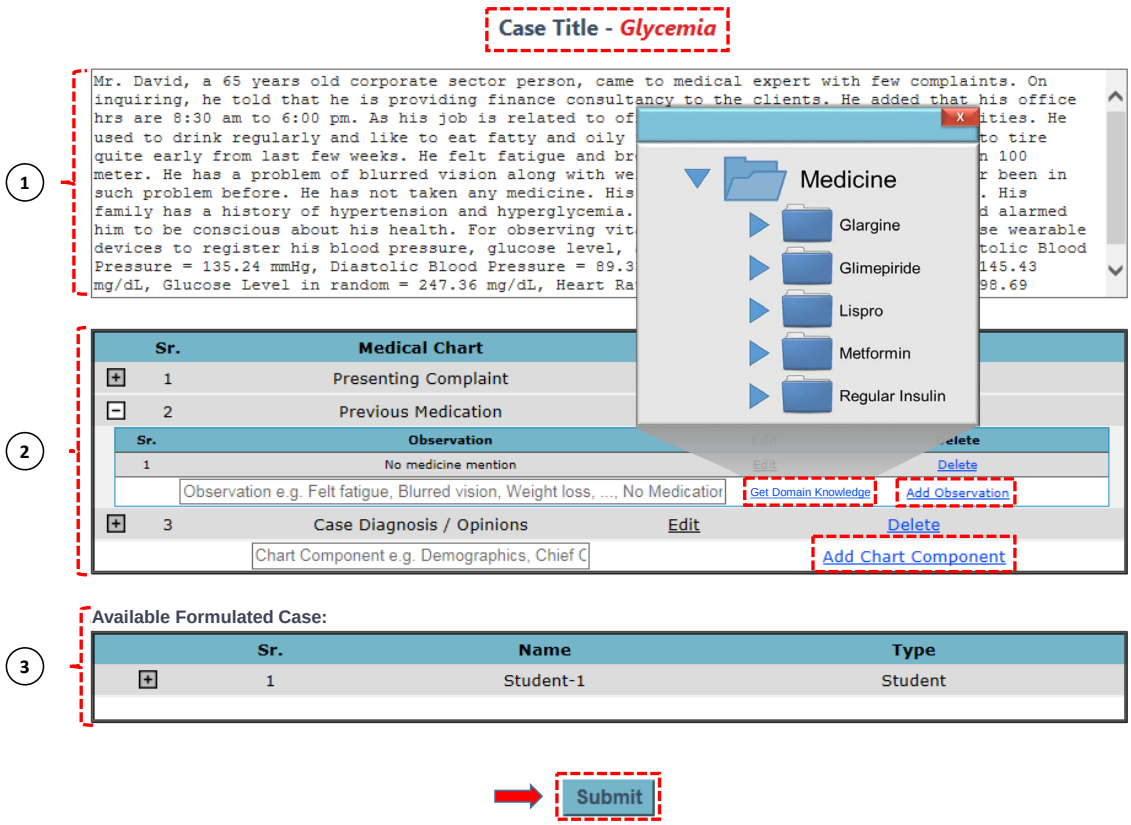


Figure 5.12: Student view for case formulation

struct it. After reviewing the submitted formulation, the teacher enters their opinions and feedback for each student in each case through the feedback interface as shown in Figure 5.13. This feedback enables students to improve their learning conceptualization and increase their understanding, which contributes to their evolution of knowledge [206]. Once experiential experts induce their practical knowledge through feedback, the students are empowered to utilize this knowledge for better clinical competency [48].

5.4 System Evaluation

In specialised literature, medical education programs are considered to be complex due to their diverse interactions amongst participants and environment [20]. Discussion-based learning in a small-group, like CBL, is considered to be a complex system [46]. In small-groups, multiple

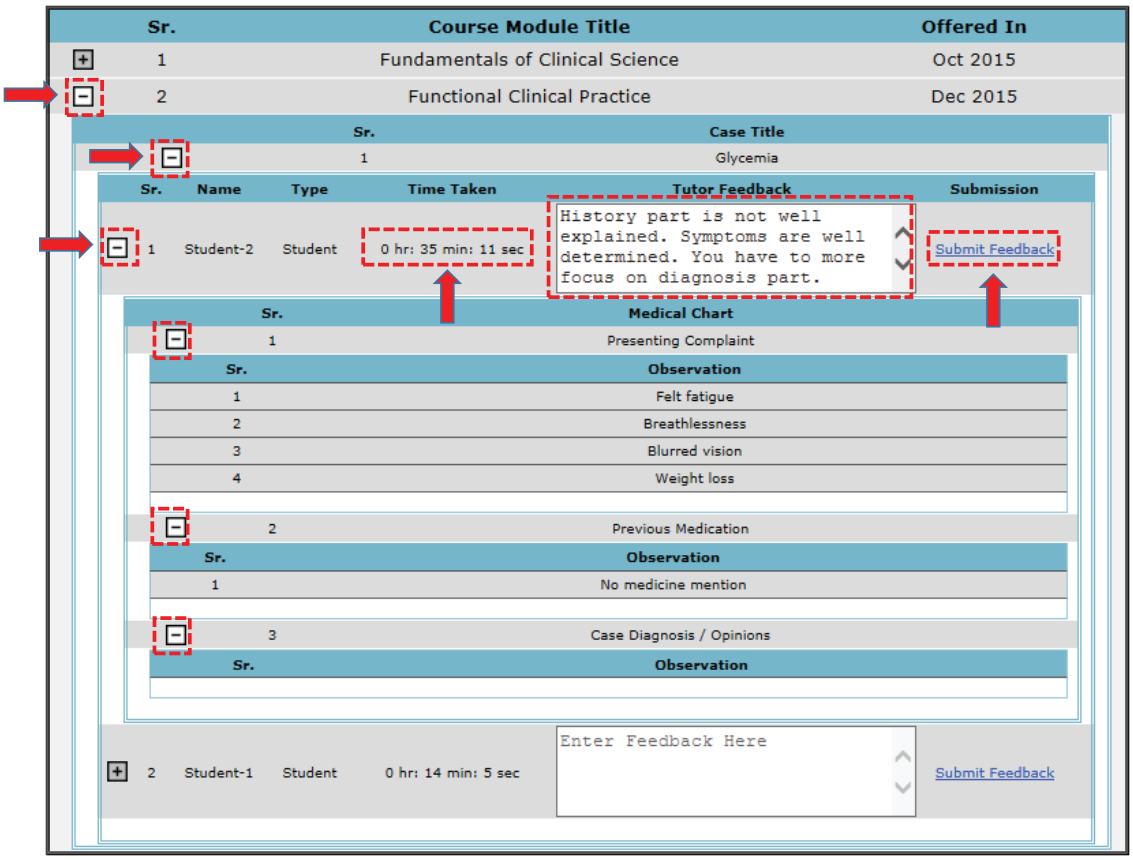


Figure 5.13: Tutor view for providing feedback

medical students are interacting and exchanging information with each other, where each student is also a complex system [47]. For evaluation of complex systems, the CIPP (context/input/process/product) model is most widely used in the literature [207–211] and is considered as a powerful approach [20]. This model is used for evaluating as well as improving ongoing medical education programs; it is also consistent with system theory, and to some degree, with complexity theory [20, 211]. For holistic understanding, the proposed system is evaluated under the umbrella of the CIPP model.

The evaluation phase of any system involves studying, investigating and judging the importance of the information for making a decision about the worth of an education program [20, 212]. In the health profession education field new developments in system evaluation are evolving, which are not yet ready for mainstream approaches [213]. Developments are still based on

outcome-based evaluation, which is considered not to be sufficient for evaluating the health profession [213]. Furthermore, predicting the outcome of an education program is limited if we have an incomplete view of a program [20]. For evaluation of health professionalism, the program's context and process elements of the CIPP model are widely used factors for assessing health professionalism using surveys and informal interviews [210, 213].

For holistic understanding, the proposed system is evaluated in heterogeneous environments by involving multiple stakeholders and using multiple methods such as quantitative methods (e.g. surveys) and qualitative methods (e.g. interviews and focus groups) under the umbrella of the CIPP model. The functional mapping of the evaluation approach used in iCBLS's evaluation, with each element of CIPP model are illustrated in Table 5.4. In the first element of the CIPP model, heterogeneous environments, surveys, interviews, and focus groups are considered for *context* study, while for *input* study, literature review, other learning projects visitation, and expert consultation are performed in the second element. In the third element, the establishment of evaluation questions, data collection as well as participant interviews are covered for analysis purposes as to whether iCBLS is delivered in the manner in which we intended. Finally, the last element is used for assessing the outcome of the proposed system through positive or negative feedback and it also assesses the degree to which the target is achieved.

Table 5.4: CIPP elements and tasks performed in iCBLS [20]

Context	**Input**	**Process**	**Product**
Heterogeneous environments	Literature review	Establish the evaluation questions	Judgements of the system
Surveys	Visiting standard learning programs	Collect the data	Assessment of achieved targets
Interview	Consulting expert	Participant interviews	Interviews about system's outcomes
Focus groups			Surveys

In this study, the *product* element of the CIPP model is responsible for investigating the impact of the proposed CBL system usability in terms of students' interaction and the system effectiveness for students' learning, which is explained in the following subsections. For both environ-

ments, survey-based as well as interview-based system evaluations are selected after performing beta testing on a given scenario with control information. In each survey, multiple evaluation questions are selected and prepared as shown in Figures C.1, C.2, and C.3 in Appendix C. The questions are considered as important factors for system evaluation, to help understand the success or shortcomings of the system [20]. A CBL case is created through iCBLS and made available to all users to assess the impact of the developed system. Moreover, in each environment, the system is first introduced and demonstrated before the survey and interview are completed. The evaluation setup for both environments is illustrated in Table 5.5.

Table 5.5: Evaluations setup for the iCBLS

Evaluation Criteria	**Environment-I (Users Interaction Evaluation)**	**Environment-II (Learning Effectiveness Evaluation)**
Primary hypothesis	Flexible and easy to learn	System appropriateness with respect to students' learning
Secondary hypothesis	Minimum memory load and efficiency (minimum actions required)	System suitability with respect to students' level and user friendly system
Variables	System capability, Operation learning, Screen flow, Interface consistency, Interface interaction, Minimal action, Memorization	Appropriate for group learning, Appropriate for solo learning, Useful for improving clinical skills, Performing tasks straightforward
Options and weightages set for each question	Excellent (10), Good (8), Above Average (6), Average (4), Poor (2)	Five options from 1 to 5 representing poor to excellent and quantified in multiple of 20
Survey method	Google docs (Online), 1-on-1	Google docs (Online), 1-on-1, small groups at the hospital
Number of users	209 (different years students and professionals)	

5.4.1 Users interaction evaluation

This subsection describes the system evaluation in terms of interaction [214]. We compiled the feedback provided by the users to draw the holistic picture of the system, which is illustrated in Table 5.6. Overall, we found that *interaction* of the system through the interface was generally valued by the users, whereas, *load on the users' memory* was criticized as experiential knowledge of students relies on memory and recognition [215] and due to scattered knowledge, it is difficult

to obtain [48]. The results, as illustrated in Table 5.6, clearly show that users were quite satisfied with the *system capabilities*, *operating learning*, *screen flow*, and *interface interaction*, which were greater than 70%. The area of *consistency* and *load on user memory* due to surplus steps needs improvement as the system's interface was not able to satisfy the users. It was also inferred that the display of error and support message windows has further room for improvement.

Table 5.6: Summarized response with respect to categories results

Evaluation Criteria		**Sub-categories Response**	**Categories Response**	
Categories	**Sub-categories**	**(out of 10)**	**(Average)**	**(%)**
System Capability	System reliability	7.5555	7.8148	78.15
	Designed for all levels of users	8.0740		
Operation Learning	Learning to operate the system	7.2963	7.2037	72.04
	Reasonable Data grouping for easy learning	7.1111		
Screen Flow	Reading characters on the screen	6.9629	7.0555	70.56
	Organization of information	7.1481		
Interface Consistency	Consistency across the label format and location	7.1111	6.6851	66.85
	Consistent symbols for graphic data standard	6.2592		
Interface Interaction	Flexible data entry design	8.0000	8.1481	81.48
	Zooming for display expansion	8.2962		
Minimal Action	Wizard-based information management	6.7407	6.0185	60.19
	Provision of default values	5.2962		
Memorization	Highlighted selected information	4.8148	4.8148	48.15

We classify our users into 3 groups on the basis of their responses which are; those who evaluated the system as poor; those who evaluated it as average and above average; and those

who evaluated it as good and excellent. In order to assess an evaluation criteria of the system, the comparison of evaluation for various categories is depicted in Figure 5.14. The details of these results are given in Tables 5.7 and 5.8.

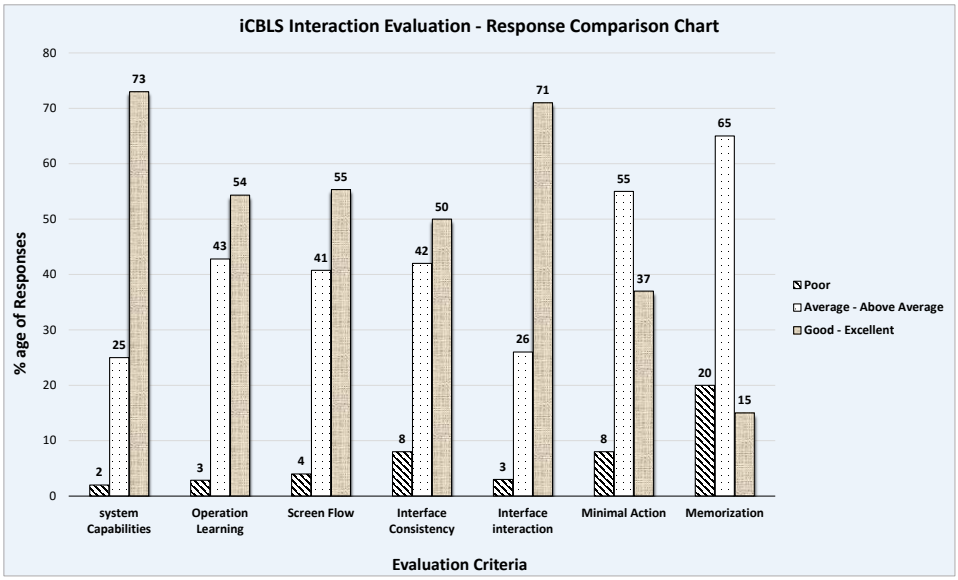


Figure 5.14: iCBLS interaction evaluation - response comparison chart

As represented in Figure 5.14, the confidence on *system capabilities* and *interface interaction* was measured as about 70% from all users. Approximately 50% of users considered the *interface consistency*, *screen flow* and *operation learning* aspect as an appealing factor. Moreover, less than 40% of users were satisfied with the factors like *load on human memory* and with the *number of actions performed*, in order to achieve a particular task. Finally, for the evaluation of the system, on average, 42% of users responded with their level of satisfaction as medium level.

Tables 5.7 and 5.8 present the detailed results of the proposed system's interaction, where results with bold size are depicted in Figure 5.14.

5.4.2 Learning effectiveness evaluation

This evaluation captures educational viewpoints and highlights the aspects that are technically inclined. We compiled the feedback from users as shown in Figure 5.15 and found that *system appropriateness* with respect to *group learning* was mostly appreciated by the users.

Table 5.7: Interaction evaluations results.

Evaluation criteria		**Poor**	**Average**	**Above average**	**Good**	**Excellent**
Categories	**Sub-categories**	**(%)**	**(%)**	**(%)**	**(%)**	**(%)**
System capability	System reliability	2	14	14	45	25
	Designed for all levels of users	2	7	15	35	41
	Average	**2**	**10.5**	**14.5**	**40**	**33**
	Range average	**2**	**25**		**73**	
Operation learning	Learning to operate the system	4	12	23	36	25
	Reasonable Data grouping for easy learning	2	8	43	34	13
	Average	**3**	**10**	**33**	**35**	**19**
	Range average	**3**	**43**		**54**	
Screen flow	Reading characters on the screen	4	15	27	38	16
	Organization of information	4	8	32	32	24
	Average	**4**	**11.5**	**29.5**	**35**	**20**
	Range average	**4**	**41**		**55**	
Interface consistency	Consistency across the label format and location	4	15	23	38	20
	Consistent symbols for graphic data standard	12	19	27	33	9
	Average	**8**	**17**	**25**	**35.5**	**14.5**
	Range average	**8**	**42**		**50**	
Interface interaction	Flexible data entry design	5	6	23	37	29
	Zooming for display expansion	1	3	20	25	51
	Average	**3**	**4.5**	**21.5**	**31**	**40**
	Range average	**3**	**26**		**71**	

Table 5.8: Interaction evaluations results (cont.).

Evaluation criteria		Poor	Average	Above average	Good	Excellent
Categories	**Sub-categories**	**(%)**	**(%)**	**(%)**	**(%)**	**(%)**
Minimal action	Wizard-based information management	0	14	35	45	6
	Provision of default values	16	32	29	23	0
	Average	**8**	**23**	**32**	**34**	**3**
	Range average	**8**	**55**		**37**	
Memorization	Highlighted selected information	20	41	24	12	3
	Average	**20**	**41**	**24**	**12**	**3**
	Range average	**20**	**65**		**15**	

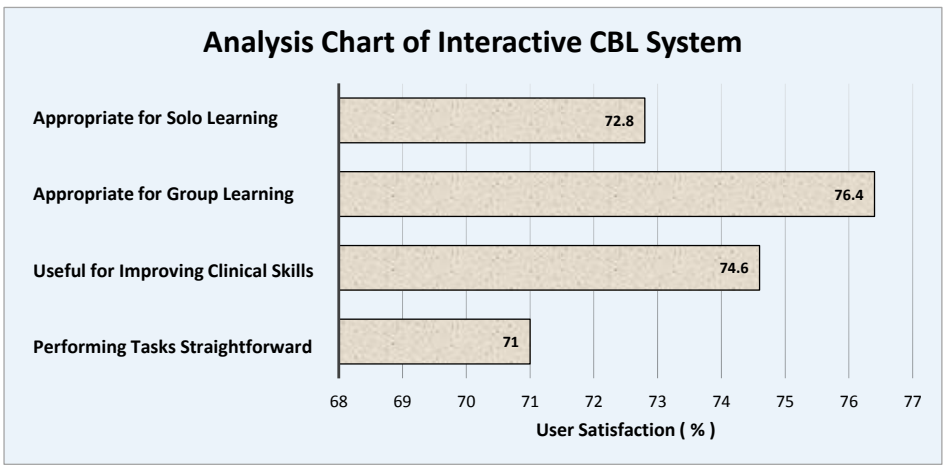


Figure 5.15: System effectiveness summary chart

Figure 5.15 clearly represents that users were quite satisfied with the *system appropriateness* for group as well as solo learning, *system usefulness* with respect to enhancing clinical skills, and *user friendliness* of the system, which were greater than 70%. We also evaluated our system to check *suitability* and *appropriateness* for different course-year levels of medical students. The system achieved votes for year-levels 2 or 3 that showed confidence on system suitability for these students, which is the stage where students begin to do placements at hospitals.

We also conducted an open-ended survey evaluation in order to analyse whether the proposed online interactive CBL system contributed to effective medical knowledge and skill learning. All 155 first-year medical students in the *University of Tasmania* used the system for one semester and were asked to provide information on their learning experiences and perceptions through an open-ended survey with 3 different questions. Open-ended questions normally aim to collect more detailed information and actionable insights since they allow the freedom and space to answer in as much detail as the respondents would like to give. The aim of the conducted survey was to encourage students to share their medical skill learning experience by using the proposed CBL system. The table 5.9 shows the open-ended survey questions for learning efficiency evaluation.

Table 5.9: Open-ended Survey Question for Learning Effiency Evaluation

Q. #	**Open-ended Survey Questions**
1	What did you like most about the computer-based tutorial preparation module?
2	What did you like least about the computer-based tutorial preparation module?
3	Are there any areas where you think the Case-Based Learning tutorial program can improve?

Responses to our survey evaluation with 155 students can be summarized as follows:

(Q1) Key phrases from answers to the first question were ‘self-learning’, ‘independent thinking’, ‘gaining more professional knowledge’ and ‘distance learning’. The majority of students felt that CBL encouraged them to be active learners, and to use logic to think and learn with real-world cases. The system also allowed students to access the learning materials (real-world problems observation, problem-solving skill learning, and teachers’ feedback) in rural settings, and students felt this sort of online system could help support this lack of resources.

(Q2) The key phrase from answers to the second question was ‘senior level education’. Further to that, some students felt this system is not suitable for very junior students (i.e. first-years) as they have not had the exposure to clinical environments to understand what sort of content they were given in such a system format without some guidance. However, other students thought that it was great opportunity to review their learned knowledge and skills as first-year students.

(Q3) Key phrases from answers to the third question were 'time consuming work', 'tutor engagement', 'improvement of feedback interface'. Some students mentioned that it would be better to have more tutor support or feedback on their answers through the system interface in real time.

The evaluation of any medical education program can be affected by participants' characteristics, the domain knowledge, and the environment in which the system operates [216]. As it is an initial concept, we do believe that with increased usage of the system this efficiency may increase for complicated scenarios and it will help students to understand the real world's patient-medical scenario in an efficient and accurate manner [193].

5.5 IoT-based Flip Learning Platform (IoTFLiP)

To exploit the IoT infrastructure for supporting flipped case-based learning in the cloud environment with state-of-the-art security and privacy measures for potentially personalized medical data, this section describes the IoTFLiP architecture and working scenario for the case-base flip learning using IoTivity.

5.5.1 Proposed platform architecture

This section describes the architecture of the proposed IoT-based platform, called IoTFLiP, as shown in Fig. 5.16, and the functionalities of its layers. The IoTFLiP integrates the features of existing individual platforms and can be used for medical as well as other domains.

Figure 5.16 is composed of eight layers, which are abstractly divided into 2 blocks on the basis of communication and resources, called *local* and *cloud* processing blocks. The first four layers, namely *Data Perception, Data Aggregation and Preprocessing, Local Security*, and *Access Technologies Layers* deal with communication and resources locally, while the remaining four layers, namely *Cloud Security, Presentation, Application and Service*, and *Business Layers* deal at the cloud level. These layers cover important features including data interoperability for handling data heterogeneity, smart gateway communication for reducing network traffic burden, fog computation for resource management to avoid delayed information sharing, multiple levels

of storage and communication securities, error handling while transcoding, application delivery policies, and business policies. Moreover, these layers provide state-of-the-art security as well as privacy measures for potentially personalized data, and give support for application delivery in private, public, and hybrid approaches. Further details for each layer are given below.

Data perception layer

In this layer, the identification of devices is performed, where devices are used to monitor, track, and store patients' vital signs, statistics or medical information. The devices include Google Gear[1], Google Glass[2], patient monitoring sensors, smart meters, wearable health monitoring sensors, video cameras, and smart phones.

Data aggregation and preprocessing layer

This layer is divided into *Data Aggregation* and *Data Preprocessing* modules. The *Data Aggregation* module deals with heterogeneous data interoperability, load balancing, and smart data communication issues i.e. communicating only when required, by either storing the data locally, temporarily, or discarding it when not required. This data aggregation & preprocessing requires resources, which are not available in relatively less rich sensor nodes and other perception layer devices. Therefore, *fog* is incorporated here. Fog computing is a small cloud that acts as an extended cloud to the edge of the network [140]. In order to perform the rich tasks and filtering of communication, which sensors and light IoTs are not capable of doing, smart gateways are used [141]. Similarly, the *Data Preprocessing* module filters the irrelevant data for faster communication and then transcodes it by encoding, decoding, and translation.

Local security layer

Security is the degree of protection from unauthorized users and attacks. Security of patient information is the most ethical issue. Patient always remains cautious about sharing personal medical information with others. In order to secure the temporary storage and for fog to cloud communication, a *Local Security Layer* is introduced. This layer addresses where security is required and

[1]https://store.google.com/product/samsung_gear_live
[2]https://en.wikipedia.org/wiki/Google_Glass

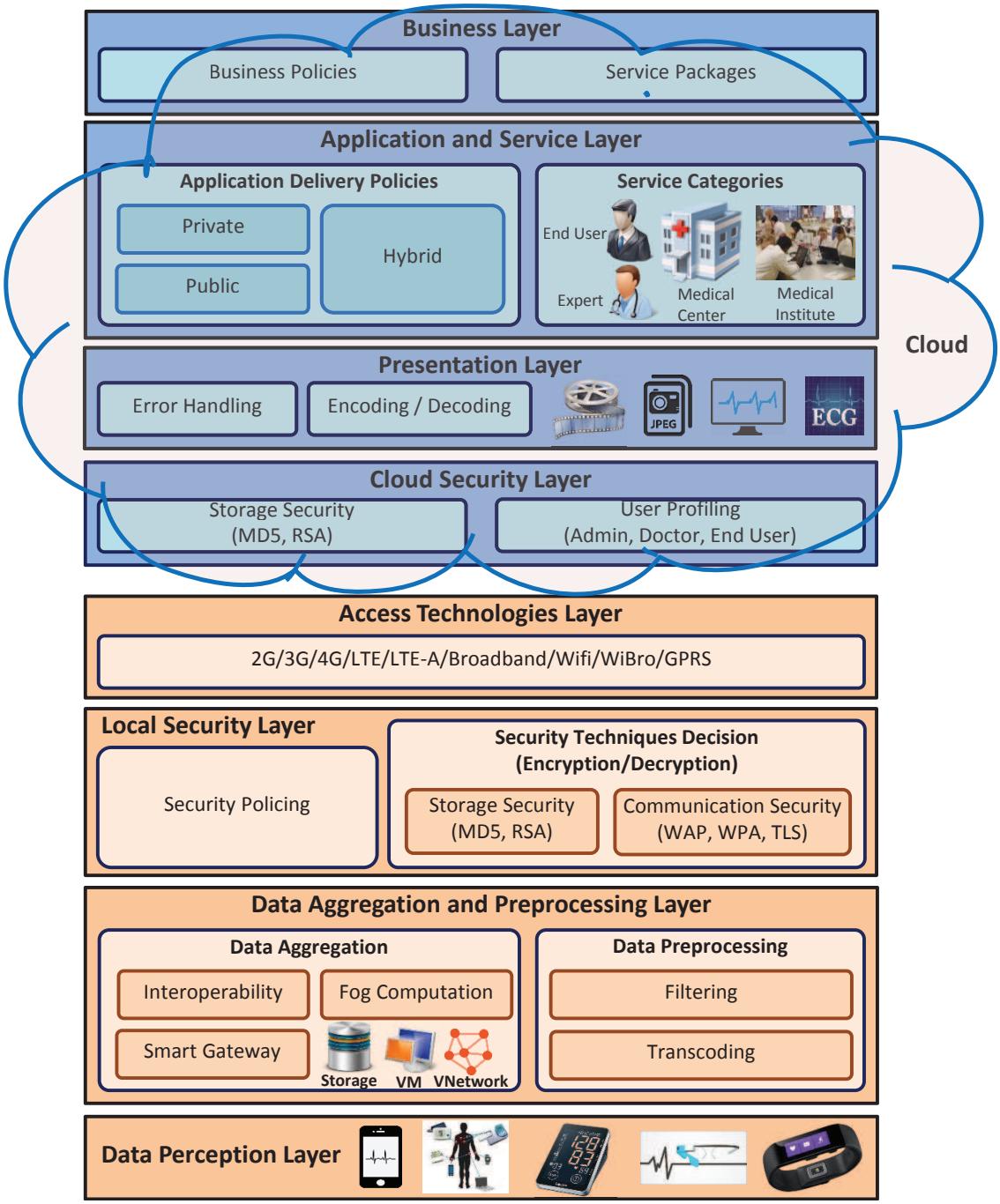


Figure 5.16: IoT-based flip learning platform (IoTFLiP) architecture

which security technique is needed. Also, security policies are defined in this layer, in which decision of operations e.g. whether to be encrypted or not, are made. In order to assess where security

is required, if the communication is local, temporary storages are used which require local security. Similarly, based on application requirement, it has been decided whether fast communication will be feasible or slow. For example, for the case of patient monitoring urgency, security may not be affordable. In that case, we need fast communication. For answering which security technique for storage or protocol for communication are chosen, it has been decided based on the application requirement. For storage security, *Message-Digest* algorithm (MD5), *Rivest-Shamir-Adleman* algorithm (RSA), *Digital-Signature-Algorithm* (DSA), and so on, while for communication security, *Wireless Application Protocol* (WAP), *Wi-Fi Protected Access* (WPA), and *Transport Layer Security* (TLS) can be used.

Access technologies layer

Various access networks exist for communication with cloud resources like WiFi, WiBro, GPRS, LTE, etc. This layer selects the access technology based on the requirement and availability of services.

Cloud security layer

Once data moves from local processing blocks to cloud processing blocks, security of data storage is an important aspect in order to secure it from various types of cloud-users. *Secured User* profiling can also be an important fact. This layer deals with storage security and user profiling. Security techniques are chosen based on user profiling.

Presentation layer

The main purpose of this layer is to deal with encoding, decoding, and error handling during data transformation. This layer converts data into a proper, understandable format e.g. ECG graph, pulse rate, angiography, prescription text, picture, video etc.

Application and service layer

In this layer, *Application Delivery Policies* are defined in terms of private, public or hybrid access. Based on the service scope, delivery policies are chosen. Also, services are categorized based

on the requirements from ordinary user access to admin user access. For example, one service is categorized into two parts. One part is accessible to everyone, while other part is restricted. The same categorization can be applicable for medical center administration and medical institutes.

Business layer

This layer deals with the business policies and services packages in terms of free or subscribed rates. The packages offerings are according to the usage.

5.5.2 Working scenario

In this section, the working scenario for case-base flip learning using IoTivity is described through steps as shown in Fig. 5.17. This scenario covers CBL case creation, case formulation, case evaluation, case feedback, and storing medical knowledge. In Fig. 5.17, the steps 1 to 5 belong to *Data Perception*, *Data Aggregation and Preprocessing*, *Local Security*, and *Access Technologies layers* of the IoTFLiP, while steps 6 to 10 belong to *Cloud Security*, *Presentation*, *Application and Service*, and *Business layers* of the IoTFLiP.

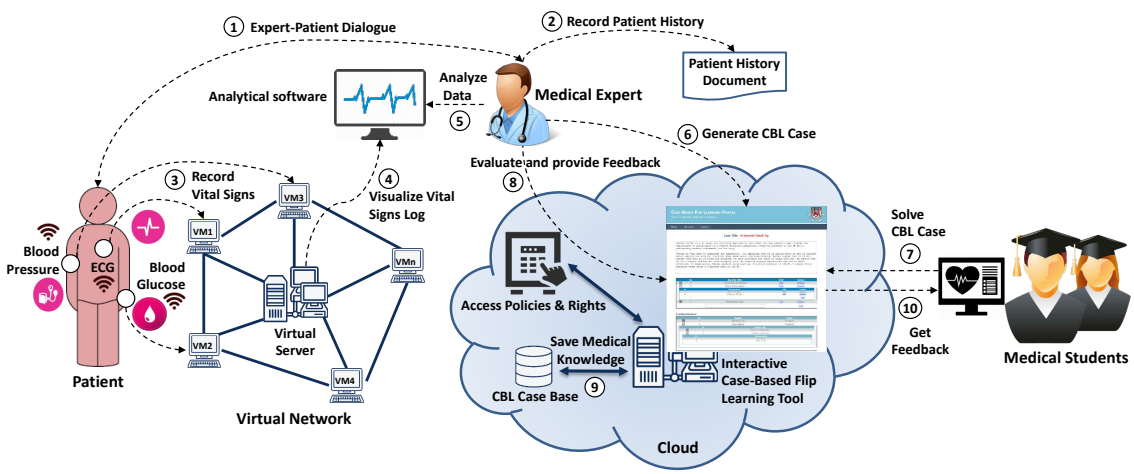


Figure 5.17: Working scenario for case-based flip learning

In this study, for generating a realistic CBL case scenario, a patients' dataset was prepared with the help of a medical expert and a knowledge engineer, as illustrated in Table 5.10. This dataset can be easily generated by available IoT gadgets, which are mentioned in Step-3. For the

Table 5.10: Patients' vital signs data

ID	Age	Gender	Systolic BP[a]	Diastolic BP	GL[b] at Fasting	GL at Random	Heart Rate
1.	**65**	**M**	**135**	**89**	**145**	**247**	**90**
2.	57	F	130	87	110	160	95
3.	54	M	139	92	90	130	89
4.	16	M	136	85	85	120	79
5.	9	M	123	75	80	125	130
6.	35	F	125	84	90	125	80
7.	3	F	110	78	70	125	130
8.	35	M	110	78	85	115	63
9.	45	M	123	85	80	130	85
10.	43	M	127	85	130	180	84

[a] Blood Pressure, [b] Glucose Level

patients' dataset, over the period of one week, three times a day, data is prepared by considering the valid ranges and important facts from available online resources[3,4,5]. The expert built 10 CBL case scenarios based on the prepared patient data shown in Table 5.10, in which the one shown with bold text is considered as an example in this study. These scenarios were of primary level difficulty and related to the general medicine domain.

The process of creating a real-life situation case for medical students is described through steps, as shown in Figure 5.17, that are explained as follows.

Step-1:

The expert interviews with patient to get the basic information such as patient name, gender, age, etc. Patients' names are not revealed in the Table 5.10 but we collected that in order to distinguish the patients. The exact age and gender will be used in clustering them into a specific age and gender group.

[3]Categories for Blood Pressure Levels in Adults, http://www.nhlbi.nih.gov/health/health-topics/topics/hbp

[4]Heart rates in different circumstances, https://en.wikipedia.org/wiki/Heart_rate, http://www.medicalnewstoday.com/ articles/235710.php

[5]Blood Sugar Levels for Adults With Diabetes, http://www.webmd.com/diabetes/normal-blood-sugar-levels-chart-adults, http://www.diabetes.co.uk/diabetes_care/blood-sugar-level-ranges.html

Step-2:

During the interview, experts note down the patient's history information, including review of symptoms, medication history, and family history.

Step-3:

After advice from the expert, the patient uses the wearable devices to record his vital signs of *blood pressure*, *glucose level*, and *heart rate*. These vitals are helpful for treatment and for disease diagnosis [143, 195]. To measure these vitals, multiple IoT gadgets are available, which are illustrated in Table 5.1.

Step-4:

Once vital signs are collected, the medical expert analyzes the patient's data by viewing through the graphical interfaces that are shown in Figure 5.10.

Step-5:

With analysis and processing of this data, the medical expert interprets the vital signs information, which are one-week average values such as *Systolic Blood Pressure* = 135.24 mmHg and other vitals shown in Figure 5.10.

Step-6:

The expert integrates patient history and vital signs to generate a new real-world CBL case as represented in Table 5.3.

Step-7:

Medical students solve the new real-world created case by interpreting the patient's problems. They create a significant medical story within the context of his or her life and then submit their interpretations.

Step-8:

The expert evaluates student interpretations and provides feedback to each student.

Step-9:

The iCBLS stores student interpretations along with tutor opinions; these will be helpful for computerized feedback in the future [217, 218].

Step-10:

Students receive the expert's feedback to improve their concepts and learning for their evolving knowledge.

5.6 Discussion about Significance, Challenges and Limitations of the Work

This study addresses an issue of great interest to many readers who have an interest in teaching and learning in medicine with regard to how to foster medical trainees' collaborative learning skills as a lifelong learning endeavour, using advanced technology. The main aim of every medical student is to interact with patients and to experience a variety of cases during their clinical practice period. The proposed system, iCBLS, provides the facilities for creating a real-life situation clinical case, practicing that case before and outside the class, and finally getting feedback from experts to evolve their knowledge. This system supports distance learning and provides maximum time management flexibility to each student. In addition, this system has the capability to generate useful information as well as knowledge which is then stored in a continuous manner that can be helpful in future for computerized feedback, intensive learning, better clinical competence, and transferring expertise among experts and students. Based on the aforementioned system's characteristics, we do believe that the iCBLS will be effective in professional learning.

During the real-time implementation of our proposed system, we encountered several challenges. Some of the key challenges we attempted to resolve were the hierarchical management of data, abstraction of logic, avoidance of code redundancy, and analysis of the vital signs data. To manage the addition, modification, deletion, paging and nested hierarchy of data, we have used data grids. Similarly, for abstraction or obscuration of logic and to avoid code redundancy, we have used the stored procedures. Moreover, for analyses of vital signs data, we have generated individual as well as average graphs based on reference ranges.

Limitations of the proposed approach include lack of real-time integration systems due to the .NET framework; no user interface was created for the administrator to manage course allotments and enrolments; no connection with IoT devices to collect vital signs data was developed, nor did

the system perform data validation for invalid values. Finally, the real-world clinical case creation process currently does not include image support.

5.7 Conclusions

This study describes how to foster medical trainees' collaborative learning skills as a lifelong learning endeavor using advanced technology with the support of online learning and real-world clinical cases. Practicing real-world clinical cases before and outside the class can promote learning capabilities, save class time for effective discussion, and enhance the academic experience of medical students. For this purpose, we have developed a CBL system, iCBLS, which fills the gaps between human-based and computer-based learning and utilizes the strength of both human (experiential knowledge) and computer (domain knowledge). The iCBLS creates real-world clinical cases using a semi-automatic approach with the support of their experiential knowledge, gets the domain knowledge to formulate the summaries of CBL cases and provides feedback for formulated cases. The iCBLS is developed based on the current CBL practices in Australia. iCBLS formulates the summaries of CBL cases through synergies of students as well as medical expert knowledge. This system manages multiple types of users according to their roles and privileges. In addition, this system also supports a number of features such as displaying the entire collection of data at one place, a paging facility, and support for in-line reviewing to edit and delete the displayed data. The working principle of the iCBLS is explained with the help of a *Glycemia* case study. Two types of evaluations under the umbrella of the CIPP model have been performed in heterogeneous environments. The iCBLS achieves a success rate of more than 70% for students' *interaction*, *group learning*, *solo learning*, and *improving clinical skills*. This success rate indicates that iCBLS effectively supports the learning of medical students. In addition to that, the system is most likely recommended for the year level 2-3 medical students.

Due to low cost and with reduced sensing devices size, support of IoTs for providing real and evolutionary medical cases, as well as support of recent flip learning concepts can enhance medical students' academic and practical experience. To exploit the IoT infrastructure to support flipped case-based learning in the cloud environment, an *IoT-based Flip Learning Platform*, called IoTFLiP is also presented in this study, with state-of-the-art security and privacy measures for

potentially personalized medical data. It also provides the support for application delivery in private, public, and hybrid approaches. The proposed platform integrates the features of existing individual platforms and can be used for medical as well as other domains.

Chapter 6

Conclusion and Future Direction

This chapter concludes the thesis and provides future directions in this research area. It also describes the potential applications of the proposed methodology.

6.1 Conclusion

The case-based learning (CBL) approach has been receiving attention in medical education, as it is a student-centered teaching methodology that exposes students to real-world scenarios that need to be solved using their reasoning skills and existing theoretical knowledge. Being human, students feel that traditional CBL activities require a significant amount of time and they get tired. In recent trends, more attention is given to e-learning environments for clinical practice of medical students as compared to lectures for their learning. In order to support the learning outcomes of students a plethora of web-based learning systems have been developed; however, most of them either do not support computer-based interactive case authoring as well as its formulation, without the support of acquiring real-world CBL cases, or do not provide feedback to students. Currently, very little attention is given to fill the gaps between human-based and computer-based learning. Medical literature contains a lot of useful knowledge in textual form, which can be beneficial for computer-based CBL practice. For an automated CBL, a structured knowledge construction is a challenging task. The key challenge in this regard is to select appropriate features from a larger set of features. The feature selection task requires two basic steps, ranking and filtering. Here the former step requires ranking of all features, while the latter involves filtering out irrelevant features based on some threshold value. In this regard, several feature selection methods with well-documented capabilities and limitations have already been proposed. Similarly, a feature ranking task is also important as it requires optimal cut-off value to select important features from

a list of candidate features. However, the availability of a comprehensive feature ranking and filtering approach, which alleviates the existing limitations and provides an efficient mechanism for achieving optimal results, is a major problem.

Keeping in view all above-mentioned facts and to take care of the students' learning systems, this research investigated case-based learning and proposed an interactive medical learning framework to utilize the strength of both human (experiential knowledge) and computer (domain knowledge) for preparing medical students for clinical practice. For effective and enriched learning purposes, this research includes a method to construct the domain model that will provide domain knowledge to medical students for intensive learning in the future. Finally, to construct a reliable domain model, this research investigated a feature selection methodology and proposed an efficient and comprehensive ensemble-based feature selection methodology to select informative features from a larger set of features. The key contributions of this research are as follows:

1. Introduced an efficient and comprehensive Univariate Ensemble-based Feature Selection (uEFS) methodology to select informative features from a larger set of features. For the uEFS methodology:

   (a) Proposed an innovative Unified Features Scoring (UFS) algorithm to generate a final ranked list of features after a comprehensive evaluation of a feature set. The UFS algorithm ranks the features without using any learning algorithm, high computational cost, and any individual statistical biases of state-of-the-art feature ranking methods. The current version of the UFS code and its documentation are freely available and can be downloaded from the GitHub open source platform [159, 160].

   (b) Proposed an innovative Threshold Value Selection (TVS) algorithm to define a cut-off point for removing irrelevant features and selecting a subset of features, which are deemed important for domain knowledge construction.

   (c) Performed extensive experimentation to evaluate the uEFS methodology using standard benchmark datasets; the results show that the uEFS methodology provides competitive accuracy and achieved (1) on average around a 7% increase in f-measure, and (2) on average around a 5% increase in predictive accuracy as compared to state-of-the-art methods.

2. Introduced an interactive and effective Case-Based Learning (CBL) approach to utilize the strength of both experiential knowledge and domain knowledge. The proposed approach enables the medical teacher to create real-world CBL cases for their students with the support of their experiential knowledge and computer-generated trends, review the student solutions, and give feedback and opinions to their students. This approach facilitates medical students to do CBL rehearsal with a machine-generated domain knowledge support before attending actual CBL classes. For an automated CBL:

   (a) Introduced semi-automatic real-world clinical case creation, and case formulation techniques.

   (b) Designed and developed an interactive Case-Based Learning System (iCBLS) to automate the proposed approach.

   (c) Performed two studies to evaluate the proposed CBL approach under the umbrella of the Context/Input/Process/Product (CIPP) model and achieved a success rate of more than 70% for student interaction, group learning, solo learning, and improving clinical skills.

   (d) Introduced an IoT-based Flip Learning Platform (IoTFLiP) to exploit the IoT infrastructure for supporting flipped case-based learning in a cloud environment with state-of-the-art security and privacy measures.

6.2 Future Direction

This research investigated feature selection methodologies to construct reliable domain knowledge for case-based learning and proposed an ensemble-based feature selection methodology for an automated CBL approach. Possible future directions include:

1. Currently, the proposed methodology incorporates state-of-the-art univariate filter measures to consider the relevance aspect of feature ranking and ignores the features' redundancy aspect that is also an important factor for selecting informative features from a larger set of features. In the future, we will extend the methodology for incorporating multi-variate measures to consider the redundancy aspect of features subset selection.

2. Similarly, the proposed methodology does not evaluate the suitability of a measure, or it's precision. In order to consider that factor, we will also investigate the application of fuzzy-logic for determining the cut-off threshold value in the future.

3. Furthermore, the proposed methodology takes 0.37 sec more time than state-of-the-art filter measures on a Intel (R) Core (TM) i5-2500 CPU @ 3.30GHz 3.30 GHz machine. The proposed algorithm is written in JAVA language, which has multiple packages dependencies and increases the computation time due to the cold start problem (NP-hard). In the future, we can reduce the cold start problem by optimizing the code and its dependencies. To measure the scalability of the proposed algorithm, our plan is to perform this methodology in a parallel execution environment.

4. Finally, the proposed CBL approach does not support an interactive question-answering technique. In the future, we will extend the current CBL approach towards a QA-based (Question-Answer) learning environment.

6.3 Potential Applications

In this section, we briefly describe the overall advantages of the proposed methodology and two real-world potential applications where the advantage of features ranking is highlighted.

Based on empirical as well as experiment analysis of the proposed methodology, the advantages of our proposed uEFS methodology for feature selection include that it:

- Provides competitive accuracy and achieved (1) on average around a 7% increase in f-measure, and (2) on average around a 5% increase in predictive accuracy as compared to state-of-the-art methods.
- Performs simple and fast computation
- Is not dependent on the classification algorithm
- Generally have less computational costs than wrapper and hybrid methods
- Is better suited to high dimensional datasets

- Computes rank of the features without any individual statistical biases of state-of-the-art feature ranking methods.

The proposed uEFS methodology contributes to feature selection, which is the key step in many decision support systems. The following are two real-world applications, where the proposed methodology is utilized.

1. One of the applications of features ranking is the data understanding phase of the data mining process, where data is closely inspected, which is crucial for the next phase, data preparation. For realization, the current version of the proposed UFS algorithm has been plugged into a recently developed tool, called *data-driven knowledge acquisition tool* (DDKAT) [158] to assist the domain expert in selecting informative features for the data preparation phase of *cross-industry standard process for data mining* (CRISP-DM). The DDKAT supports an end-to-end knowledge engineering process for generating production rules from a dataset.

2. A huge amount of valuable textual data is available on the web, which has led to a corresponding interest in technology for automatically extracting relative information from open data, which can then be converted into domain knowledge. In order to construct reliable domain knowledge, appropriate feature selection is another application of the proposed methodology. The feature selection (FS) task can also be performed manually by a human expert; however, this is considered as an expensive and time-consuming task; thus an automatic FS is necessary. The proposed methodology selects the important features for domain knowledge construction.

Bibliography

[1] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[2] S. Sadeghi and H. Beigy, "A new ensemble method for feature ranking in text mining," *International Journal on Artificial Intelligence Tools*, vol. 22, no. 03, p. 1350010, 2013.

[3] W. Altidor, *Stability analysis of feature selection approaches with low quality data*. Florida Atlantic Uni., 2011.

[4] M. H. Haggag, "Keyword extraction using semantic analysis," *International Journal of Computer Applications*, vol. 61, no. 1, 2013.

[5] J. Feng, F. Xie, X. Hu, P. Li, J. Cao, and X. Wu, "Keyword extraction based on sequential pattern mining," in *proceedings of the third international conference on internet multimedia computing and service*. ACM, 2011, pp. 34–38.

[6] A. B. Abacha and P. Zweigenbaum, "Automatic extraction of semantic relations between medical entities: a rule based approach," *Journal of biomedical semantics*, vol. 2, no. 5, p. S4, 2011.

[7] A. Azcarraga, M. D. Liu, and R. Setiono, "Keyword extraction using backpropagation neural networks and rule extraction," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–7.

[8] T. Kuhn, "Controlled english for knowledge representation," Ph.D. dissertation, Citeseer, 2009.

[9] P. E. Lutu and A. P. Engelbrecht, "A decision rule-based method for feature selection in predictive data mining," *Expert Systems with Applications*, vol. 37, no. 1, pp. 602–609, 2010.

[10] M. Makrehchi, "Feature ranking for text classifiers," Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Waterloo, 2007.

[11] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992*. Elsevier, 1992, pp. 249–256.

[12] R. Stoean and F. Gorunescu, "A survey on feature ranking by means of evolutionary computation," *Annals of the University of Craiova-Mathematics and Computer Science Series*, vol. 40, no. 1, pp. 100–105, 2013.

[13] S. Whiteson, P. Stone, K. O. Stanley, R. Miikkulainen, and N. Kohl, "Automatic feature selection in neuroevolution," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1225–1232.

[14] Y. Dhote, S. Agrawal, and A. J. Deen, "A survey on feature selection techniques for internet traffic classification," in *Computational Intelligence and Communication Networks (CICN), 2015 International Conference on*. IEEE, 2015, pp. 1375–1380.

[15] S. Doraisamy, S. Golzari, N. Mohd, M. N. Sulaiman, and N. I. Udzir, "A study on feature selection and classification techniques for automatic genre classification of traditional malay music." in *ISMIR*, 2008, pp. 331–336.

[16] C. Liu, W. Wang, Q. Zhao, X. Shen, and M. Konan, "A new feature selection method based on a validity index of feature subset," *Pattern Recognition Letters*, vol. 92, pp. 1–8, 2017.

[17] C. Sarkar, S. Cooley, and J. Srivastava, "Robust feature selection technique using rank aggregation," *Applied Artificial Intelligence*, vol. 28, no. 3, pp. 243–257, 2014.

[18] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for ddos detection in cloud computing,"

EURASIP Journal on Wireless Communications and Networking, vol. 2016, no. 1, p. 130, 2016.

[19] Ó. Marbán, G. Mariscal, and J. Segovia, "A data mining & knowledge discovery process model," *Data Mining and Knowledge Discovery in Real Life Applications*, vol. 2009, p. 8, 2009.

[20] A. W. Frye and P. A. Hemmer, "Program evaluation models and related theories: Amee guide no. 67," *Medical teacher*, vol. 34, no. 5, pp. e288–e299, 2012.

[21] G. Flores-Mateo and J. M. Argimon, "Evidence based practice in postgraduate healthcare education: a systematic review," *BMC health services research*, vol. 7, no. 1, p. 1, 2007.

[22] S. Demircioğlu and G. S. Selçuk, "The effect of the case-based learning method on high school physics students' conceptual understanding of the unit on energy," in *Asia-Pacific Forum on Science Learning and Teaching*, vol. 17. The Education University of Hong Kong, Department of Science and Environmental Studies, 2016, pp. 1–25.

[23] A. Samsudin, A. Suhandi, D. Rusdiana, I. Kaniawati, and B. Coştu, "Investigating the effectiveness of an active learning based-interactive conceptual instruction (albici) on electric field concept," in *Asia-Pacific Forum on Science Learning and Teaching*, vol. 17. The Education University of Hong Kong, Department of Science and Environmental Studies, 2016, pp. 1–41.

[24] S. F. McLean, "Case-based learning and its application in medical and health-care fields: a review of worldwide literature," *Journal of Medical Education and Curricular Development*, vol. 3, pp. JMECD–S20 377, 2016.

[25] M. Shepherd and B. Martz, "Problem based learning systems and technologies," in *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, 2005, pp. 39–39.

[26] R. Sule, "Medical students and faculty perceptions towards a case based learning intervention at an indian medical college," Ph.D. dissertation, McMaster University, 2016.

[27] A. A. Osinubi and K. O. Ailoje-Ibru, "A paradigm shift in medical, dental, nursing, physiotherapy and pharmacy education: From traditional method of teaching to case-based method of learning-a review," *Annual Research and Review in Biology*, vol. 4, no. 13, pp. 2053–2072, 2014.

[28] A. L. Warren and T. Donnon, "Optimizing biomedical science learning in a veterinary curriculum: a review," *Journal of veterinary medical education*, vol. 40, no. 3, pp. 210–222, 2013.

[29] Ottawa, "A guide to case based learning, faculty of medicine, university of ottawa," http://www.medicine.uottawa.ca/facdev/assets/documents/The%20Case%20Based%20Learning%20Process.pdf, 2017, accessed: 2017-01-20.

[30] Cardiff, "Cardiff university: What is case based learning?" http://medicine.cf.ac.uk/medical-education/undergraduate/why-choose-cardiff/our-curriculum/what-case-based-learning-copy/, 2015, accessed: 2016-08-24.

[31] M. J. Patil and S. Karadesai, "To determine the effectiveness of case based tutorials as compared to traditional tutorials in microbiology," *National Journal of Integrated Research in Medicine*, vol. 7, no. 2, pp. 5–8, 2016.

[32] O. Eseonu, R. Carachi, and N. Brindley, "Case-based anatomy teaching: a viable alternative?" *The clinical teacher*, vol. 10, no. 4, pp. 236–241, 2013.

[33] S. Gade and S. Chari, "Case-based learning in endocrine physiology: an approach toward self-directed learning and the development of soft skills in medical students," *Advances in physiology education*, vol. 37, no. 4, pp. 356–360, 2013.

[34] T. T. T. FISH, "If we teach them to fish: Solving real nursing problems through problem-based learning," *Annual Review of Nursing Education Volume 3, 2005: Strategies for Teaching, Assessment, and Program Planning*, p. 109, 2005.

[35] K. Hoffman, M. Hosokawa, R. Blake Jr, L. Headrick, and G. Johnson, "Problem-based learning outcomes: ten years of experience at the university of missouri?columbia school of medicine," *Academic Medicine*, vol. 81, no. 7, pp. 617–625, 2006.

[36] J. E. Thistlethwaite, D. Davies, S. Ekeocha, J. M. Kidd, C. MacDougall, P. Matthews, J. Purkis, and D. Clay, "The effectiveness of case-based learning in health professional education. a beme systematic review: Beme guide no. 23," *Medical teacher*, vol. 34, no. 6, pp. e421–e444, 2012.

[37] S. R. Stewart and L. S. Gonzalez, "Instruction in professional issues using a cooperative learning, case study approach," *Communication Disorders Quarterly*, vol. 27, no. 3, pp. 159–172, 2006.

[38] C. M. Bowe, J. Voss, and H. Thomas Aretz, "Case method teaching: An effective approach to integrate the basic and clinical sciences in the preclinical medical curriculum," *Medical teacher*, vol. 31, no. 9, pp. 834–841, 2009.

[39] J. P. Schoeman, M. Van Schoor, L. L. Van der Merwe, and R. Meintjes, "A case-based, small-group cooperative learning course in preclinical veterinary science aimed at bridging basic science and clinical literacy," *Journal of the South African Veterinary Medical Association*, vol. 80, no. 1, p. 31, 2009.

[40] T. Revel and H. Yussuf, "Taking primary care continuing professional education to rural areas: Lessons from the united arab emirates," *Australian Journal of Rural Health*, vol. 11, no. 6, pp. 271–276, 2003.

[41] Ottawa, "A guide to case based learning, faculty of medicine, university of ottawa," http://www.medicine.uottawa.ca/facdev/assets/documents/The%20Case%20Based%20Learning%20Process.pdf, 2010, accessed: 2017-01-20.

[42] K. Brown, M. Commandant, A. Kartolo, C. Rowed, A. Stanek, H. Sultan, K. Tool, and V. Wininger, "Case based learning teaching methodology in undergraduate health sciences," *Interdisciplin. J. Health Sci*, vol. 2, no. 2, pp. 47–65, 2011.

[43] J. T. Hansen and S. K. Krackov, "The use of small group case-based exercises in human gross anatomy: A method for introducing active learning in a traditional course format," *Clinical Anatomy*, vol. 7, no. 6, pp. 357–366, 1994.

[44] A. Rodriguez-Barbero and J. Lopez-Novoa, "Teaching integrative physiology using the quantitative circulatory physiology model and case discussion method: evaluation of the learning experience," *Advances in Physiology Education*, vol. 32, no. 4, pp. 304–311, 2008.

[45] A. Roehl, S. L. Reddy, and G. J. Shannon, "The flipped classroom: An opportunity to engage millennial students through active learning," *Journal of Family and Consumer Sciences*, vol. 105, no. 2, pp. 44–49, 2013.

[46] S. Mennin, "Small-group problem-based learning as a complex adaptive system," *Teaching and Teacher Education*, vol. 23, no. 3, pp. 303–313, 2007.

[47] S. Mennin, "Teaching, learning, complexity and health professions education," *J Int Assoc Med Sci Educat*, vol. 20, pp. 162–165, 2010.

[48] E. Baillergeau and J. W. Duyvendak, "Experiential knowledge as a resource for coping with uncertainty: evidence and examples from the netherlands," *Health, Risk & Society*, vol. 18, no. 7-8, pp. 407–426, 2016.

[49] D. Willoughby and S. Philosophy, "Experiential knowledge," Available online: https://simplyphilosophy.org/study/experiential-knowledge/, 2018, accessed: 2018-02-03.

[50] S. Halim, "Human and computer," Available online: https://www.comp.nus.edu.sg/~stevenha/viz/appendixC_hci.pdf, 2018, accessed: 2018-02-03.

[51] M. M. Cummings, "Man versus machine or man + machine?" *IEEE Intelligent Systems*, vol. 29, no. 5, pp. 62–69, 2014.

[52] C. Gopalan, "The impact of rapid change in educational technology on teaching in higher education," *HAPS Educator*, vol. 20, no. 4, pp. 85–90, 2016.

[53] J. B. Morrow, D. Sepdham, L. Snell, C. Lindeman, and A. Dobbie, "Evaluation of a web-based family medicine case library for self-directed learning in a third-year clerkship," *Family medicine*, vol. 42, no. 7, pp. 496–500, 2010.

[54] V. Kulak and G. Newton, "A guide to using case-based learning in biochemistry education," *Biochemistry and Molecular Biology Education*, vol. 42, no. 6, pp. 457–473, 2014.

[55] M. Ali, H. S. M. Bilal, J. Hussain, S. Lee, and B. H. Kang, "An interactive case-based flip learning tool for medical education," in *Inclusive Smart Cities and e-Health*. Springer, 2015, pp. 355–360.

[56] UTMB, "Design a case, university of texas medical branch - utmb," http://www.designacase.org/default.aspx, 2013, accessed: 2016-12-04.

[57] UNM, "Extension for community healthcare outcomes - echo, the university of new mexico," http://echo.unm.edu/, 2016, accessed: 2016-12-04.

[58] F.-M. Shyu, Y.-F. Liang, W. Hsu, J.-J. Luh, and H.-S. Chen, "A problem-based e-learning prototype system for clinical medical education," *Medinfo*, vol. 11, no. Pt 2, pp. 983–987, 2004.

[59] Y.-M. Cheng, K. Sheng-Huang, L. Shi-Jer, and S. Ru-Chu, "The effect of applying online pbl case system to multiple disciplines of medical education," *TOJET: The Turkish Online Journal of Educational Technology*, vol. 11, no. 4, pp. 283–294, 2012.

[60] C. DiLullo, H. J. Morris, and R. M. Kriebel, "Clinical competencies and the basic sciences: an online case tutorial paradigm for delivery of integrated clinical and basic science content," *Anatomical sciences education*, vol. 2, no. 5, pp. 238–243, 2009.

[61] S. Suebnukarn and P. Haddawy, "Comet: A collaborative tutoring system for medical problem-based learning," *Intelligent Systems, IEEE*, vol. 22, no. 4, pp. 70–77, 2007.

[62] M. Sharples, N. Jeffery, B. du Boulay, B. Teather, D. Teather, and G. du Boulay, "Structured computer-based training in the interpretation of neuroradiological images," *International journal of medical informatics*, vol. 60, no. 3, pp. 263–280, 2000.

[63] M. Boubouka, "A web-based case-based learning environment–use in the didactics of informatics," Ph.D. dissertation, National and Kapodistrian University of Athens, 2013.

[64] L.-S. Chen, Y.-M. Cheng, W. Sheng-Feng, C. Yong-Guo, and C.-H. Lin, "Applications of a time sequence mechanism in the simulation cases of a web-based medical problem-based

learning system," *Journal of Educational Technology & Society*, vol. 12, no. 1, pp. 149–161, 2009.

[65] S. Philosophy, "Factual knowledge," Available online: https://simplyphilosophy.org/study/factual-knowledge/, 2018, accessed: 2018-02-03.

[66] P. Baitule and V. Chole, "A review on improved text mining approach for conversion of unstructured to structured text'," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 12, pp. 156–159, 2014.

[67] W. Altidor, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Ensemble feature ranking methods for data intensive computing applications," in *Handbook of data intensive computing*. Springer, 2011, pp. 349–376.

[68] S. Joseph, C. Mugauri, and S. Sumathy, "Sentiment analysis of feature ranking methods for classification accuracy," in *IOP Conference Series: Materials Science and Engineering*, vol. 263. IOP Publishing, 2017, p. 042011.

[69] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 313–325.

[70] G. Wang, Q. Song, H. Sun, X. Zhang, B. Xu, and Y. Zhou, "A feature subset selection algorithm automatic recommendation method," *Journal of Artificial Intelligence Research*, 2013.

[71] L. A. Belanche and F. F. González, "Review and evaluation of feature selection algorithms in synthetic problems," *arXiv preprint arXiv:1101.2320*, 2011.

[72] R. C. Prati, "Combining feature ranking algorithms through rank aggregation," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 1–8.

[73] L. Rokach, B. Chizi, and O. Maimon, "Feature selection by combining multiple methods," in *Advances in Web Intelligence and Data Mining*. Springer, 2006, pp. 295–304.

[74] K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, and M. Sebag, "Ensemble feature ranking," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2004, pp. 267–278.

[75] I. Slavkov, B. Zenko, and S. Dzeroski, "Evaluation method for feature rankings and their aggregations for biomarker discovery." in *MLSB*, 2010, pp. 122–135.

[76] O. Rusu, I. Halcu, O. Grigoriu, G. Neculoiu, V. Sandulescu, M. Marinescu, and V. Marinescu, "Converting unstructured and semi-structured data into knowledge," in *Roedunet International Conference (RoEduNet), 2013 11th*. IEEE, 2013, pp. 1–4.

[77] A. Onan and S. Korukoğlu, "A feature selection model based on genetic rank aggregation for text sentiment classification," *Journal of Information Science*, vol. 43, no. 1, pp. 25–38, 2017.

[78] A. Sharma and S. Dey, "Performance investigation of feature selection methods and sentiment lexicons for sentiment analysis," *IJCA Special Issue on Advanced Computing and Communication Technologies for HPC Applications*, vol. 3, pp. 15–20, 2012.

[79] J. Novaković, "Toward optimal feature selection using ranking methods and classification algorithms," *Yugoslav Journal of Operations Research*, vol. 21, no. 1, 2016.

[80] E. Tuv, A. Borisov, and K. Torkkola, "Feature selection using ensemble based ranking against artificial contrasts," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 2181–2186.

[81] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Information Sciences*, vol. 282, pp. 111–135, 2014.

[82] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[83] Y. Chen, Y. Li, X.-Q. Cheng, and L. Guo, "Survey and taxonomy of feature selection algorithms in intrusion detection system," in *Information security and cryptology*. Springer, 2006, pp. 153–167.

[84] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and information systems*, vol. 12, no. 1, pp. 95–116, 2007.

[85] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *ICML*, vol. 1. Citeseer, 2001, pp. 74–81.

[86] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys, "Robust biomarker identification for cancer diagnosis with ensemble feature selection methods," *Bioinformatics*, vol. 26, no. 3, pp. 392–398, 2010.

[87] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[88] S. I. Ali and W. Shahzad, "A feature subset selection method based on symmetric uncertainty and ant colony optimization," in *Emerging Technologies (ICET), 2012 International Conference on*. IEEE, 2012, pp. 1–6.

[89] I. Wikimedia Foundation, "Data science," https://en.wikipedia.org/wiki/Data_science, 2017, accessed: 2017-02-18.

[90] G. Gupta, *Introduction to data mining with case studies*. PHI Learning Pvt. Ltd., 2014.

[91] R. Feldman, Y. Aumann, M. Finkelstein-Landau, E. Hurvitz, Y. Regev, and A. Yaroshevich, "A comparative study of information extraction strategies," *Computational Linguistics and Intelligent Text Processing*, pp. 21–34, 2002.

[92] A. Doan, R. Ramakrishnan, and S. Vaithyanathan, "Managing information extraction: state of the art and research directions," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 799–800.

[93] J. Rajni and S. Taneja, "U-struct: A framework for conversion of unstructured text documents into structured form," in *Advances in Computing, Communication, and Control*, 2013, pp. 59–69.

[94] C. Friedman, L. Shagina, Y. Lussier, and G. Hripcsak, "Automated encoding of clinical documents based on natural language processing," *Journal of the American Medical Informatics Association*, vol. 11, no. 5, pp. 392–402, 2004.

[95] F. Leao, K. Revoredo, and F. Baiao, "Learning well-founded ontologies through word sense disambiguation," in *2013 Brazilian Conference on Intelligent Systems*, 2013, pp. 195–200.

[96] P. Reuss, K.-D. Althoff, W. Henkel, M. Pfeiffer, O. Hankel, and R. Pick, "Semi-automatic knowledge extraction from semi-structured and unstructured data within the omaha project," in *International Conference on Case-Based Reasoning*. Springer, 2015, pp. 336–350.

[97] R. Bhowmik, "Keyword extraction from abstracts and titles," in *Southeastcon, 2008. IEEE*. IEEE, 2008, pp. 610–617.

[98] S. Loh, L. K. Wives, and J. P. M. de Oliveira, "Concept-based knowledge discovery in texts extracted from the web," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 29–39, 2000.

[99] P.-I. Chen and S.-J. Lin, "Automatic keyword prediction using google similarity distance," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1928–1938, 2010.

[100] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 101–110.

[101] M. Wenchao, L. Lianchen, and D. Ting, "A modified approach to keyword extraction based on word-similarity," in *Intelligent computing and intelligent systems, 2009. ICIS 2009. IEEE international conference on*, vol. 3. IEEE, 2009, pp. 388–392.

[102] J. Liu and J. Wang, "Keyword extraction using language network," in *Natural Language Processing and Knowledge Engineering, 2007. NLP-KE 2007. International Conference on*. IEEE, 2007, pp. 129–134.

[103] H. S. Al-Khalifa and H. C. Davis, "Folksonomies versus automatic keyword extraction: An empirical study," *IADIS International Journal On Computer Science And Information Systems (IJCSIS)*, vol. 1, pp. 132–143, 2006.

[104] S. Beliga, A. Meštrović, and S. Martinčcić-Ipšić, "Toward selectivity based keyword extraction for croatian news," *arXiv preprint arXiv:1407.4723*, 2014.

[105] B.-Y. Kang and S.-J. Lee, "Document indexing: a concept-based approach to term weight estimation," *Information processing & management*, vol. 41, no. 5, pp. 1065–1080, 2005.

[106] B. Lott, "Survey of keyword extraction techniques," *UNM Education*, vol. 50, 2012.

[107] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[108] J. D. Wren, R. Bekeredjian, J. A. Stewart, R. V. Shohet, and H. R. Garner, "Knowledge discovery by automated identification and ranking of implicit relationships," *Bioinformatics*, vol. 20, no. 3, pp. 389–398, 2004.

[109] G. Ercan and I. Cicekli, "Using lexical chains for keyword extraction," *Information Processing & Management*, vol. 43, no. 6, pp. 1705–1714, 2007.

[110] L. Gazendam, C. Wartena, and R. Brussee, "Thesaurus based term ranking for keyword extraction," in *Database and Expert Systems Applications (DEXA), 2010 Workshop on*. IEEE, 2010, pp. 49–53.

[111] T. Kuhn, "A survey and classification of controlled natural languages," *Computational Linguistics*, vol. 40, no. 1, pp. 121–170, 2014.

[112] M. Miyabe and H. Uozaki, "Controlled natural language simplifying language use," in *LREC2014Workshop-CNL Proceedings*, 2014, pp. 1–2.

[113] R. N. Shiffman, G. Michel, M. Krauthammer, N. E. Fuchs, K. Kaljurand, and T. Kuhn, "Writing clinical practice guidelines in controlled natural language," in *International Workshop on Controlled Natural Language*. Springer, 2009, pp. 265–280.

[114] R. N. Shiffman, G. Michel, R. M. Rosenfeld, and C. Davidson, "Building better guidelines with bridge-wiz: development and evaluation of a software assistant to promote clarity, transparency, and implementability," *Journal of the American Medical Informatics Association*, vol. 19, no. 1, pp. 94–101, 2012.

[115] H. Safwat and B. Davis, "A brief state of the art of cnls for ontology authoring," in *International Workshop on Controlled Natural Language*. Springer, 2014, pp. 190–200.

[116] S. Williams, R. Power, and A. Third, "How easy is it to learn a controlled natural language for building a knowledge base?" in *International Workshop on Controlled Natural Language*. Springer, 2014, pp. 20–32.

[117] R. Schwitter, "English as a formal specification language," in *Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on*. IEEE, 2002, pp. 228–232.

[118] R. Denaux, *Intuitive ontology authoring using controlled natural language*. University of Leeds, 2013.

[119] R. Power, D. Scott, and R. Evans, "What you see is what you meant: direct knowledge editing with natural language feedback." in *ECAI*, vol. 98, 1998, pp. 677–681.

[120] R. Schwitter, A. Ljungberg, and D. Hood, "Ecole–a look-ahead editor for a controlled language," *Proceedings of EAMT-CLAW03, May*, vol. 1517, 2003.

[121] R. Schwitter, "Creating and querying formal ontologies via controlled natural language," *Applied Artificial Intelligence*, vol. 24, no. 1-2, pp. 149–174, 2010.

[122] T. Kuhn, "The understandability of owl statements in controlled english," *Semantic Web*, vol. 4, no. 1, pp. 101–115, 2013.

[123] T. Kuhn, "How to evaluate controlled natural languages," *arXiv preprint arXiv:0907.1251*, 2009.

[124] R. Jones, "Problem-based learning: description, advantages, disadvantages, scenarios and facilitation," *Anaesthesia and intensive care*, vol. 34, no. 4, p. 485, 2006.

[125] K. Brown, M. Commandant, A. Kartolo, C. Rowed, A. Stanek, H. Sultan, K. Toor, and V. Wininger, "Case based learning teaching methodology in undergraduate health sciences," *Interdisciplinary Journal of Health Sciences*, vol. 2, no. 2, pp. 48–66, 2016.

[126] E. Crowther and S. Baillie, "A method of developing and introducing case-based learning to a preclinical veterinary curriculum," *Anatomical sciences education*, vol. 9, no. 1, pp. 80–89, 2016.

[127] L. F. Scavarda, B. Hellingrath, T. Kreuter, A. M. T. Thomé, M. X. Seeling, J.-H. Fischer, and R. Mello, "A case method for sales and operations planning: a learning experience from germany," *Production*, vol. 27, no. SPE, pp. 1–17, 2017.

[128] G. F. Grauer, S. D. Forrester, C. Shuman, and M. W. Sanderson, "Comparison of student performance after lecture-based and case-based/problem-based teaching in a large group," *Journal of veterinary medical education*, vol. 35, no. 2, pp. 310–317, 2008.

[129] I. Umbrin, "Difference between problem based learning pbl and case based learning cbl, 2014," http://www.slideshare.net/izzaumbrin/difference-between-problem-based-learning-pbl-and-case-based-learning-cbl, 2014, accessed: 2017-01-21.

[130] J. Popay and G. Williams, "Public health research and lay knowledge," *Social science & medicine*, vol. 42, no. 5, pp. 759–768, 1996.

[131] P. Storkerson, "Experiential knowledge, knowing and thinking," in *Experiential Knowledge, Method & Methodology: International Conference*, 2009, pp. 1–18.

[132] J. W. Duyvendak, *De planning van ontplooiing: wetenschap, politiek en de maakbare samenleving*. Sdu, 1999.

[133] L. Prior, "Belief, knowledge and expertise: the emergence of the lay expert in medical sociology," *Sociology of health & illness*, vol. 25, no. 3, pp. 41–57, 2003.

[134] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," in *CISCO white paper, 2011*, vol. 1, 2011, p. 14.

[135] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[136] A. J. Jara, M. A. Zamora, and A. F. Skarmeta, "An internet of things–based personal device for diabetes therapy management in ambient assisted living (aal)," *Personal and Ubiquitous Computing*, vol. 15, no. 4, pp. 431–440, 2011.

[137] S. Hussain, J. H. Bang, M. Han, M. I. Ahmed, M. B. Amin, S. Lee, C. Nugent, S. McClean, B. Scotney, and G. Parr, "Behavior life style analysis for mobile sensory data in cloud computing through mapreduce," *Sensors*, vol. 14, no. 11, pp. 22 001–22 020, 2014.

[138] O. Banos, M. B. Amin, W. A. Khan, M. Afzel, M. Ahmad, M. Ali, T. Ali, and S. .....and Lee, "An innovative platform for person-centric health and wellness support," in *In Bioinformatics and Biomedical Engineering*. Springer, 2015, pp. 131–140.

[139] D. Niewolny, "How the internet of things is revolutionizing healthcare," in *White paper*, 2013, pp. 1–8.

[140] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for iot," in *Advanced Information Networking and Applications (AINA), 2015. Proceedings. IEEE 29th International Conference on*. IEEE, 2015, pp. 687–694.

[141] M. Aazam, P. P. Hung, and E.-N. Huh, "Smart gateway based communication for cloud of things," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014. Proceedings. IEEE Ninth International Conference on*. IEEE, 2014, pp. 1–6.

[142] C.-W. Chang, P. Lin, C.-W. Tseng, Y.-K. Kong, W.-C. Lien, M.-C. Wu, and C.-Y. Wu, "Poster: Design and implementation of mobile e-learning platform for medical training," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2015, pp. 385–386.

[143] W. Zhao, C. Wang, and Y. Nakahira, "Medical application on internet of things," in *Communication Technology and Application (ICCTA 2011), IET International Conference on*. IET, 2011, pp. 660–665.

[144] J. L. Bishop and M. A. Verleger, "The flipped classroom: A survey of the research," in *ASEE National Conference Proceedings, Atlanta, GA*, vol. 30(9), 2013, pp. 1–18.

[145] M. B. Gilboy, S. Heinerichs, and G. Pazzaglia, "Enhancing student engagement using the flipped classroom," *Journal of nutrition education and behavior*, vol. 47, no. 1, pp. 109–114, 2015.

[146] S. E. Street, K. O. Gilliland, C. McNeil, and K. Royal, "The flipped classroom improved medical student performance and satisfaction in a pre-clinical physiology course," *Medical Science Educator*, vol. 25, no. 1, pp. 35–43, 2015.

[147] M. Attik, "Using ensemble feature selection approach in selecting subset with relevant features," in *International Symposium on Neural Networks*. Springer, 2006, pp. 1359–1366.

[148] L. E. A. Santana, D. F. de Oliveira, A. M. Canuto, and M. C. de Souto, "A comparative analysis of feature selection methods for ensembles with different combination methods," in *2007 International Joint Conference on Neural Networks*. IEEE, 2007, pp. 643–648.

[149] M. Ali, S. Lee, and B. H. Kang, "Udekam: A methodology for acquiring declarative structured knowledge from unstructured knowledge resources," in *Machine Learning and Cybernetics (ICMLC), 2016 International Conference on*, vol. 1. IEEE, 2016, pp. 177–182.

[150] C.-S. Lee, Y.-F. Kao, Y.-H. Kuo, and M.-H. Wang, "Automated ontology construction for unstructured text documents," *Data & Knowledge Engineering*, vol. 60, no. 3, pp. 547–566, 2007.

[151] A. Houser, "Framemaker: Structured or unstructured?" http://www.writersua.com/articles/frame/, 2004.

[152] R. Jindal and S. Taneja, "U-struct: A framework for conversion of unstructured text documents into structured form," in *Advances in Computing, Communication, and Control*. Springer, 2013, pp. 59–69.

[153] A. Beigzadeh and F. Haghani, "Active learning methods: a way of tackling large classroom setting," *Strides in Development of Medical Education*, vol. 13, no. 1, pp. 107–113, 2016.

[154] M.-S. Yoo and H.-R. Park, "Effects of case-based learning on communication skills, problem-solving ability, and learning motivation in nursing students," *Nursing & health sciences*, vol. 17, no. 2, pp. 166–172, 2015.

[155] P. N. Kiat and Y. T. Kwong, "The flipped classroom experience," in *Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on*. IEEE, 2014, pp. 39–43.

[156] S. Kopp, "What is the flipped classroom?" http://ctl.utexas.edu/teaching/flipping-a-class/what, 2004, accessed: 2016-09-07.

[157] M. Ali, H. S. M. Bilal, M. A. Razzaq, J. Khan, S. Lee, M. Idris, M. Aazam, T. Choi, S. C. Han, and B. H. Kang, "Iotflip: Iot-based flip learning platform for medical education," *Digital Communications and Networks*, vol. 3, no. 3, pp. 188—-194, 2017.

[158] M. Ali, R. Ali, W. A. Khan, S. C. Han, J. Bang, T. Hur, D. Kim, S. Lee, and B. H. Kang, "A data-driven knowledge acquisition system: An end-to-end knowledge engineering process for generating production rules," *IEEE Access*, vol. 6, no. 99, pp. 15 587–15 607, 2018.

[159] M. Ali, "Ufs - unified features scoring code, version 1.0," Available online: https://github.com/ubiquitous-computing-lab/Mining-Minds/blob/master/knowledge-curation-layer/DDKAT/src/main/java/org/uclab/mm/kcl/ddkat/dataselector/FeatureEvaluator.java, 2017, accessed: 2018-04-04.

[160] M. Ali, "A documentation of ufs for features scoring," Available online:

https://github.com/ubiquitous-computing-lab/Mining-Minds/tree/gh-pages/doc/kcl-doc/DDKAT/doc/org/uclab/mm/kcl/ddkat/dataselector, 2017, accessed: 2018-04-04.

[161] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[162] A. Ahmad and L. Dey, "A feature selection technique for classificatory analysis," *Pattern Recognition Letters*, vol. 26, no. 1, pp. 43–56, 2005.

[163] A. Grigorev, "Rule-based classifier," Available online: http://mlwiki.org/index.php/Rule-Based_Classifier#One_Rule_Algorithm, 2014, accessed: 2018-06-13.

[164] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with relieff," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997.

[165] A. Yousefpour, R. Ibrahim, and H. N. A. Hamed, "Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis," *Expert Systems with Applications*, vol. 75, pp. 80–93, 2017.

[166] G. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing microarray gene expression data*. John Wiley & Sons, 2005, vol. 422.

[167] M. Humphrey, S. J. Cunningham, and I. H. Witten, "Knowledge visualization techniques for machine learning," *Intelligent Data Analysis*, vol. 2, no. 4, pp. 333–347, 1998.

[168] G. Williams, "Cross validation, data mining, desktop survival guide, 2010," https://www.togaware.com/datamining/survivor/Cross_Validation.html, 2010, accessed: 2017-02-18.

[169] R. Joshi, "Accuracy, precision, recall & f1 score: Interpretation of performance measures," http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/, 2018, accessed: 2018-03-26.

[170] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble feature selection: homogeneous and heterogeneous approaches," *Knowledge-Based Systems*, vol. 118, pp. 124–139, 2017.

[171] Z. Liu, X. Lv, K. Liu, and S. Shi, "Study on svm compared with the other text classification methods," in *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, vol. 1. IEEE, 2010, pp. 219–222.

[172] M. Ali, A. M. Qamar, and B. Ali, "Data analysis, discharge classifications, and predictions of hydrological parameters for the management of rawal dam in pakistan," in *2013 12th International Conference on Machine Learning and Applications*, vol. 1, 2013, pp. 382–385.

[173] I. Tsamardinos, E. Greasidou, M. Tsagris, and G. Borboudakis, "Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation," *arXiv preprint arXiv:1708.07180*, 2017.

[174] R. Van Ittersum and E. Spalding, "Understanding the Difference Between Structured and Unstructured Documents," www.disusa.com, Tech. Rep., 2005.

[175] D. Ferrucci and A. Lally, "Uima: an architectural approach to unstructured information processing in the corporate research environment," *Natural Language Engineering*, vol. 10, no. 3-4, pp. 327–348, 2004.

[176] M. Á. Sicilia, *Interoperability in Healthcare Information Systems: Standards, Management, and Technology: Standards, Management, and Technology*. IGI Global, 2013.

[177] M. Zorrilla and D. García-Saiz, "A service oriented architecture to provide data mining services for non-expert data miners," *Decision Support Systems*, vol. 55, no. 1, pp. 399–411, 2013.

[178] R. Ali, M. H. Siddiqi, and S. Lee, "Rough set-based approaches for discretization: A compact review," *Artificial Intelligence Review*, vol. 44, no. 2, pp. 235–263, 2015.

[179] V. Dhar, "Data science and prediction," *Communications of the ACM*, vol. 56, no. 12, pp. 64–73, 2013.

[180] F. Provost and T. Fawcett, "Data science and its relationship to big data and data-driven decision making," *Big Data*, vol. 1, no. 1, pp. 51–59, 2013.

[181] P.-S. Gregory, "Kdnuggets methodology poll," http://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html, 2014, accessed: 2017-01-12.

[182] T. Kuhn, "Authoring tools for ace," http://attempto.ifi.uzh.ch/site/docs/authoring_tools.html, 2007.

[183] T. Yue, L. C. Briand, and Y. Labiche, "A systematic review of transformation approaches between user requirements and analysis models," *Requirements Engineering*, vol. 16, no. 2, pp. 75–99, 2011.

[184] O. Mohammed, S. Mohammed, J. Fiaidhi, S. Fong, and T.-h. Kim, "Clinical narratives context categorization: The clinician approach using rapidminer," *International Journal of Bio-Science and Bio-Technology*, vol. 6, no. 4, pp. 45–56, 2014.

[185] K. Kaljurand, "Ace view—an ontology and rule editor based on attempto controlled english." in *OWLED*, 2008.

[186] A. Taylor, M. Marcus, and B. Santorini, "The penn treebank: an overview," in *Treebanks*. Springer, 2003, pp. 5–22.

[187] A. Ghosh, "Bengali text summarization using singular value decomposition," Ph.D. dissertation, Jadavpur University, 2014.

[188] C. Shearer, "The crisp-dm model: the new blueprint for data mining," *Journal of data warehousing*, vol. 5, no. 4, pp. 13–22, 2000.

[189] J. Patrick, Y. Wang, and P. Budd, "An automated system for conversion of clinical notes into snomed clinical terminology," in *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*. Australian Computer Society, Inc., 2007, pp. 219–226.

[190] S. Nirmalya, T. Kaushik, and D. Rituparna, "Students' opinion towards audio-visual aids used in lecture classes," *IOSR Journal of Dental and Medical Sciences*, vol. 14, no. 4, pp. 96–100, 2015.

[191] C. F. Herreid and N. A. Schiller, "Case studies and the flipped classroom," *Journal of College Science Teaching*, vol. 42, no. 5, pp. 62–66, 2013.

[192] M. Fahim, M. Idris, R. Ali, C. Nugent, B. Kang, E.-N. Huh, and S. Lee, "Athena: a personalized platform to promote an active lifestyle and wellbeing based on physical, mental and social health primitives," *Sensors*, vol. 14, no. 5, pp. 9313–9329, 2014.

[193] P. Koles, S. Nelson, A. Stolfi, D. Parmelee, and D. DeStephen, "Active learning in a year 2 pathology curriculum," *Medical education*, vol. 39, no. 10, pp. 1045–1055, 2005.

[194] L. S. Behar-Horenstein, F. A. Catalanotto, and M. M. Nascimento, "Anticipated and actual implementation of case-based learning by dental faculty members during and after training," *Journal of dental education*, vol. 79, no. 9, pp. 1049–1060, 2015.

[195] N. Bui and M. Zorzi, "Health care applications: a solution based on the internet of things," in *Applied Sciences in Biomedical and Communication Technologies, 2011. Proceedings. ACM 4th International Symposium on*. ACM, 2011, p. 131.

[196] K. Ullah, M. A. Shah, and S. Zhang, "Effective ways to use internet of things in the field of medical and smart health care," in *Intelligent Systems Engineering (ICISE), 2016 International Conference on*. IEEE, 2016, pp. 372–379.

[197] M. Maksimović and V. Vujović, "Internet of things based e-health systems: Ideas, expectations and concerns," in *Handbook of Large-Scale Distributed Computing in Smart Healthcare*. Springer, 2017, pp. 241–280.

[198] R. Ali, J. Hussain, M. H. Siddiqi, M. Hussain, and S. Lee, "H2rm: A hybrid rough set reasoning model for prediction and management of diabetes mellitus," *Sensors*, vol. 15, no. 7, pp. 15 921–15 951, 2015.

[199] A. D. Association, "Diagnosing diabetes and learning about prediabetes," http://www.diabetes.org/diabetes-basics/diagnosis/, 2016, accessed: 2016-09-17.

[200] I. Wikimedia Foundation, "Heart rate," https://en.wikipedia.org/wiki/Heart_rate, 2016, accessed: 2016-08-17.

[201] N. Azmi and L. M. Kamarudin, "Enabling iot: Integration of wireless sensor network for healthcare application using waspmote," in *AIP Conference Proceedings 1808*. AIP Publishing, 2017, pp. 020 010 1–5.

[202] P. Sturmey, *Clinical case formulation: Varieties of approaches*. John Wiley & Sons, 2009.

[203] M. Adam Blatner, "The art of case formulation," http://www.blatner.com/adam/psyntbk/formulation.html, 2006, accessed: 2016-12-18.

[204] A. Godoy and S. N. Haynes, "Clinical case formulation," *European Journal of Psychological Assessment*, vol. 27, no. 1, pp. 1–3, 2011.

[205] WJU, "Creating a medical chart, wheeling jesuit university / center for educational technologies," http://www.e-missions.net/cybersurgeons/?/medchart/, 2016, accessed: 2016-12-17.

[206] P. J. White, M. Heidemann, M. Loh, and J. J. Smith, "Integrative cases for teaching evolution," *Evol. Educ. Outreach*, vol. 6, pp. 1–7, 2013.

[207] W. Warju, "Educational program evaluation using cipp model," *Innovation of Vocational Technology Education*, vol. 12, no. 1, pp. 36–42, 2016.

[208] A. D. Al-Khathami, "Evaluation of saudi family medicine training program: The application of cipp evaluation format," *Medical teacher*, vol. 34, no. sup1, pp. S81–S89, 2012.

[209] R. Chinta, M. Kebritchi, and J. Elias, "A conceptual framework for evaluating higher education institutions," *International Journal of Educational Management*, vol. 30, no. 6, pp. 989–1002, 2016.

[210] Y. Steinert, S. Cruess, R. Cruess, and L. Snell, "Faculty development for teaching and evaluating professionalism: from programme design to curriculum change," *Medical education*, vol. 39, no. 2, pp. 127–136, 2005.

[211] D. L. Stufflebeam and C. L. Coryn, *Evaluation theory, models, and applications*. John Wiley & Sons, 2014, vol. 50.

[212] D. A. Cook, "Twelve tips for evaluating educational programs," *Medical teacher*, vol. 32, no. 4, pp. 296–301, 2010.

[213] F. Haji, M.-P. Morin, and K. Parker, "Rethinking programme evaluation in health professions education: beyond 'did it work?'," *Medical education*, vol. 47, no. 4, pp. 342–351, 2013.

[214] J. P. Chin, V. A. Diehl, and K. L. Norman, "Development of an instrument measuring user satisfaction of the human-computer interface," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1988, pp. 213–218.

[215] D. Rubin, "Memory in oral traditions: The cognitive psychology of counting-out rhymes, ballads, and epics," 1995.

[216] R. Geyer, A. Mackintosh, and K. Lehmann, *Integrating UK and European social policy: the complexity of Europeanisation*. Radcliffe Publishing, 2005.

[217] W. Dubitzky, A. Schuster, J. Hughes, and D. Bell, "An advanced case-knowledge architecture based on fuzzy objects," *Applied Intelligence*, vol. 7, no. 3, pp. 187–204, 1997.

[218] D. Kilroy, "Problem based learning," *Emergency medicine journal*, vol. 21, no. 4, pp. 411–413, 2004.

Appendix A

List of Acronyms

Acronyms

In alphabetical order:

ACE Attempto Controlled English

CBL Case-Based Learning

CIPP Context/Input/Process/Product

CNL Controlled Natural Language

CRISP-DM Cross Industry Standard Process for Data Mining

CS Chi-Square

DDKAT Data-Driven Knowledge Acquisition Tool

DM Data Mining

DS Data Science

EFS Ensemble Feature Selection

FS Feature Selection

GR Gain Ratio

iCBLS Interactive Case-Based Learning System

IG Information Gain

IoT Internet of Things

IoTFLiP IoT-based Flip Learning Platform

kNN k-Nearest Neighbors

PBL Problem-Based Learning

POS Part of Speech

S Significance

SVM Support Vector Machine

SU Symmetric Uncertainty

TF-IDF Term Frequency - Inverse Domain Frequency

TM Text Mining

uEFS Univariate Ensemble-based Feature Selection

TVS Threshold Value Selection

UFS Unified Features Scoring

WEKA Waikato Environment for Knowledge Analysis

Appendix B

UFS Algorithm - Source Code

```
/**
 * Copyright [2017] [Maqbool Ali]
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
import java.io.File;
import java.util.ArrayList;
import org.apache.commons.io.FileUtils;
import org.apache.wink.json4j.JSONArray;
import org.apache.wink.json4j.OrderedJSONObject;
import weka.attributeSelection.ChiSquaredAttributeEval;
import weka.attributeSelection.GainRatioAttributeEval;
import weka.attributeSelection.InfoGainAttributeEval;
import weka.attributeSelection.Ranker;
import weka.attributeSelection.SignificanceAttributeEval;
```

```
import weka.attributeSelection.SymmetricalUncertAttributeEval;
import weka.core.Instances;
import weka.core.converters.CSVLoader;

// TODO: Auto-generated Javadoc
/**
* This class computes the features' scores.
*/
public class FeatureEvaluator {

    /** The features titles list */
    private ArrayList<String> featureTitles;

    /** The features scores list */
    private ArrayList<Double> featureScores;

    /** The features weights list */
    private ArrayList<Double> featureWeights;

    /** The features priorities list */
    private ArrayList<Double> featurePriorities;

    /** base directory to store resource data files */
    private static final String BASE_DIR = System.getProperty("user.home") + "/resources/";

    /**
     * Constructor to instantiate a new FeatureEvaluator object.
     *
     * @param json the data string
     * @param data the data set
     * @throws Exception the exception
     */
```

```
public FeatureEvaluator(String json, Instances data) throws Exception {

    this.featureTitles = new ArrayList<String>();
    this.featureScores = new ArrayList<Double>();
    this.featureWeights = new ArrayList<Double>();
    this.featurePriorities = new ArrayList<Double>();

   OrderedJSONObject jsonObject = new OrderedJSONObject(json.toString());
     JSONArray jsontokenArray = jsonObject.getJSONArray("unprocessed_data");
     String csvString="";
     String str;
     for(int i=0;i<jsontokenArray.length();i++){
      str= jsontokenArray.get(i).toString();
          str = str.substring(1, str.length()-1);
          csvString += str +"\n";
          }

     String filePath = BASE_DIR + "InputDataSet.csv";
     File file =new File(filePath);
     // if file does not exists, then create it
     if (!file.exists())
      file.createNewFile();

     FileUtils.writeStringToFile(file,csvString);

    CSVLoader loader=new CSVLoader();
    loader.setSource(new File(filePath));
    data=loader.getDataSet();

    if (data.classIndex() == -1)
          data.setClassIndex(data.numAttributes() - 1);

    int numUnlabeledAttributes = data.numAttributes()-1;
```

```
double[] minmaxValues = new double[2];
double min, max;

String [] options = new String [1];
options [0] = "-T -1.7976931348623157E308 -N -1";
Ranker atrank = new Ranker();
atrank.setOptions(options);

weka.attributeSelection.AttributeSelection atsel = new
    weka.attributeSelection.AttributeSelection();

// Information Gain Attribute Evaluator
InfoGainAttributeEval infoGainAttrEval = new InfoGainAttributeEval();
atsel.setEvaluator(infoGainAttrEval);
atsel.setSearch(atrank);
atsel.SelectAttributes(data);
double[] infoGainRanks = new double[numUnlabeledAttributes];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 infoGainRanks[i] = Math.round(10000 * infoGainAttrEval.evaluateAttribute(i)) /
     10000d;
  }
minmaxValues = computerMinMaxValues(infoGainRanks);
min = minmaxValues[0];
max = minmaxValues[1];
double[] scaledInfoGainRanks = new double[numUnlabeledAttributes];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 scaledInfoGainRanks[i] = Math.round(10000 * ((infoGainRanks[i]-min)/(max-min))) /
     10000d;
 }

// Gain Ratio Attribute Evaluator
GainRatioAttributeEval gainRatioAttrEval = new GainRatioAttributeEval();
atsel.setEvaluator(gainRatioAttrEval);
```

```
atsel . setSearch ( atrank ) ;
atsel . SelectAttributes (data) ;
double [] gainRatioRanks = new double[numUnlabeledAttributes ];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 gainRatioRanks[i] = Math.round(10000 * gainRatioAttrEval . evaluateAttribute (i)) /
      10000d;
 }
minmaxValues = computerMinMaxValues(gainRatioRanks);
min = minmaxValues[0];
max = minmaxValues[1];
double [] scaledGainRatioRanks = new double[numUnlabeledAttributes ];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 scaledGainRatioRanks[i] = Math.round(10000 * ((gainRatioRanks[i]-min)/(max-min))) /
      10000d;
 }

// Chi Squared Attribute Evaluator
ChiSquaredAttributeEval chiSquaredAttrEval = new ChiSquaredAttributeEval () ;
atsel . setEvaluator ( chiSquaredAttrEval ) ;
atsel . setSearch ( atrank ) ;
atsel . SelectAttributes (data) ;
double [] chiSquaredRanks = new double[numUnlabeledAttributes ];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 chiSquaredRanks[i] = Math.round(10000 * chiSquaredAttrEval . evaluateAttribute (i)) /
      10000d;
 }
minmaxValues = computerMinMaxValues(chiSquaredRanks);
min = minmaxValues[0];
max = minmaxValues[1];
double [] scaledChiSquaredRanks = new double[numUnlabeledAttributes ];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 scaledChiSquaredRanks[i] = Math.round(10000 * ((chiSquaredRanks[i]-min)/(max-min)))
      / 10000d;
```

```
 }

// Symmetrical Uncert Attribute Evaluator
SymmetricalUncertAttributeEval symmetricalUncertAttrEval = new
    SymmetricalUncertAttributeEval();
atsel.setEvaluator(symmetricalUncertAttrEval);
atsel.setSearch(atrank);
atsel. SelectAttributes(data);
double[] symmetricalUncertRanks = new double[numUnlabeledAttributes];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 symmetricalUncertRanks[i] = Math.round(10000 *
     symmetricalUncertAttrEval. evaluateAttribute(i)) / 10000d;
 }
minmaxValues = computerMinMaxValues(symmetricalUncertRanks);
min = minmaxValues[0];
max = minmaxValues[1];
double[] scaledSymmetricalUncertRanks = new double[numUnlabeledAttributes];
for (int i = 0; i < numUnlabeledAttributes; i++) {
 scaledSymmetricalUncertRanks[i] = Math.round(10000 *
     ((symmetricalUncertRanks[i]-min)/(max-min))) / 10000d;
 }

// Significance Attribute Evaluator
 SignificanceAttributeEval significanceAttrEval = new SignificanceAttributeEval();
atsel.setEvaluator( significanceAttrEval );
atsel.setSearch(atrank);
atsel. SelectAttributes(data);
double[] significanceRanks = new double[numUnlabeledAttributes];
for (int i = 0; i < numUnlabeledAttributes; i++) {
  significanceRanks[i] = Math.round(10000 * significanceAttrEval. evaluateAttribute(i))
     / 10000d;
 }
minmaxValues = computerMinMaxValues(significanceRanks);
```

```
min = minmaxValues[0];
max = minmaxValues[1];
double[] scaledSignificanceRanks = new double[numUnlabeledAttributes];
for (int i = 0; i < numUnlabeledAttributes; i++) {
  scaledSignificanceRanks[i] = Math.round(10000 *
      ((significanceRanks[i]-min)/(max-min))) / 10000d;
 }

double attributeSum;

double[] combinedRanks = new double[numUnlabeledAttributes];
double combinedranksSum = 0;

for (int i = 0; i < numUnlabeledAttributes; i++) {
 attributeSum = scaledInfoGainRanks[i] + scaledGainRatioRanks[i] +
      scaledChiSquaredRanks[i] + scaledSymmetricalUncertRanks[i] +
      scaledSignificanceRanks[i];
 combinedRanks[i] = Math.round(10000 * attributeSum) / 10000d;
 combinedranksSum = combinedranksSum + combinedRanks[i];
 }

double [][] tempArray = new double[numUnlabeledAttributes][2];
String [] attributesTitles = new String[numUnlabeledAttributes];
double[] attributesScores = new double[numUnlabeledAttributes];
double[] attributesWeights = new double[numUnlabeledAttributes];
double[] attributesPriorities = new double[numUnlabeledAttributes];

for (int j = 0; j < numUnlabeledAttributes; j++) {
 tempArray[j][0] = j;
 tempArray[j][1] = combinedRanks[j];
 }

double temp;
```

```
        for(int i=0; i < numUnlabeledAttributes; i++){
         for(int j=1; j < (numUnlabeledAttributes-i); j++){
            if(combinedRanks[j-1] < combinedRanks[j]){
               //swap the elements!
               temp = combinedRanks[j-1];
               combinedRanks[j-1] = combinedRanks[j];
               combinedRanks[j] = temp;
               }
            }
         }

        for (int j = 0; j < numUnlabeledAttributes; j++) {
         for (int k = 0; k < numUnlabeledAttributes; k++) {
            if (combinedRanks[j] == tempArray[k][1]){
                attributesTitles [j] = data.attribute((int)tempArray[k][0]).toString();
               String res[] = attributesTitles [j].split("\\s+");
                     attributesTitles [j] = res[1];

               this.featureTitles.add(attributesTitles [j]);
               break;
               }
            }
          attributesScores [j] = Math.round(10000 * (combinedRanks[j]/9)) / 100d;
          attributesWeights [j] = Math.round(10000 * (combinedRanks[j]/combinedranksSum)) /
              100d;
           attributesPriorities [j] = Math.round(attributesScores [j] * attributesWeights [j]) /
              100d;
          this.featureScores.add(attributesScores [j]);
          this.featureWeights.add(attributesWeights [j]);
          this.featurePriorities.add(attributesPriorities [j]);
            }

}
```

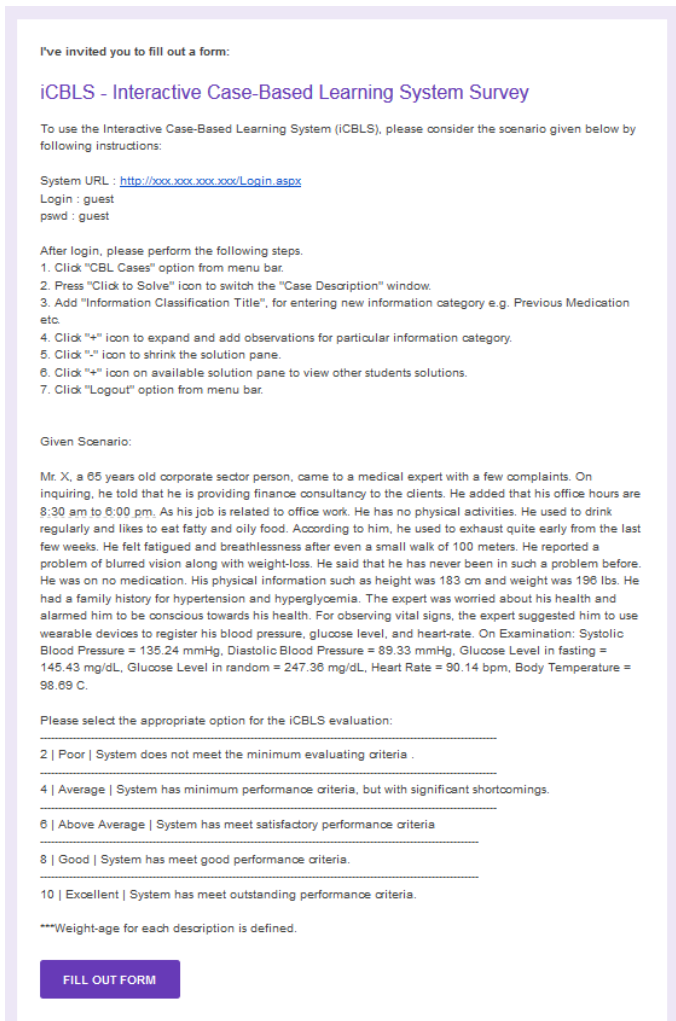
```
public ArrayList<String> getFeatureTitles () {
    return featureTitles ;
}

public void setFeatureTitles ( ArrayList<String> featureTitles ) {
    this . featureTitles = featureTitles ;
}

public ArrayList<Double> getFeatureScores() {
    return featureScores ;
}

public void setFeatureScores ( ArrayList<Double> featureScores) {
    this . featureScores = featureScores ;
}

public ArrayList<Double> getFeatureWeights() {
    return featureWeights ;
}

public void setFeatureWeights ( ArrayList<Double> featureWeights) {
    this . featureWeights = featureWeights ;
}

public ArrayList<Double> getFeaturePriorities () {
    return featurePriorities ;
}

public void setFeaturePriorities ( ArrayList<Double> featurePriorities ) {
    this . featurePriorities = featurePriorities ;
}
```

```
    protected double[] computerMinMaxValues(double dataArr[]) throws Exception {
        // assign first element of an array to largest and smallest
        double smallest = dataArr[0];
        double largetst = dataArr[0];

        for(int i=1; i<dataArr.length; i++){
            if(dataArr[i] > largetst)
                largetst = dataArr[i];
            else if(dataArr[i] < smallest)
                smallest = dataArr[i];
            }

        double minmaxArr[] = new double[2];
        minmaxArr[0] = smallest;
        minmaxArr[1] = largetst;

        return minmaxArr;
        }
}
```

Appendix C

Survey Forms for Evaluating the iCBLS

C.1 Users Interaction Evaluation



I've invited you to fill out a form:

iCBLS - Interactive Case-Based Learning System Survey

To use the Interactive Case-Based Learning System (iCBLS), please consider the scenario given below by following instructions:

System URL : http://xxx.xxx.xxx.xxx/Login.aspx
Login : guest
pswd : guest

After login, please perform the following steps.
1. Click "CBL Cases" option from menu bar.
2. Press "Click to Solve" icon to switch the "Case Description" window.
3. Add "Information Classification Title", for entering new information category e.g. Previous Medication etc.
4. Click "+" icon to expand and add observations for particular information category.
5. Click "-" icon to shrink the solution pane.
6. Click "+" icon on available solution pane to view other students solutions.
7. Click "Logout" option from menu bar.

Given Scenario:

Mr. X, a 65 years old corporate sector person, came to a medical expert with a few complaints. On inquiring, he told that he is providing finance consultancy to the clients. He added that his office hours are 8:30 am to 6:00 pm. As his job is related to office work. He has no physical activities. He used to drink regularly and likes to eat fatty and oily food. According to him, he used to exhaust quite early from the last few weeks. He felt fatigued and breathlessness after even a small walk of 100 meters. He reported a problem of blurred vision along with weight-loss. He said that he has never been in such a problem before. He was on no medication. His physical information such as height was 183 cm and weight was 196 lbs. He had a family history for hypertension and hyperglycemia. The expert was worried about his health and alarmed him to be conscious towards his health. For observing vital signs, the expert suggested him to use wearable devices to register his blood pressure, glucose level, and heart-rate. On Examination: Systolic Blood Pressure = 135.24 mmHg, Diastolic Blood Pressure = 89.33 mmHg, Glucose Level in fasting = 145.43 mg/dL, Glucose Level in random = 247.36 mg/dL, Heart Rate = 90.14 bpm, Body Temperature = 98.69 C.

Please select the appropriate option for the iCBLS evaluation:

2 | Poor | System does not meet the minimum evaluating criteria .

4 | Average | System has minimum performance criteria, but with significant shortcomings.

6 | Above Average | System has meet satisfactory performance criteria

8 | Good | System has meet good performance criteria.

10 | Excellent | System has meet outstanding performance criteria.

***Weight-age for each description is defined.

FILL OUT FORM

Figure C.1: Instructions on how to use and evaluate the iCBLS.

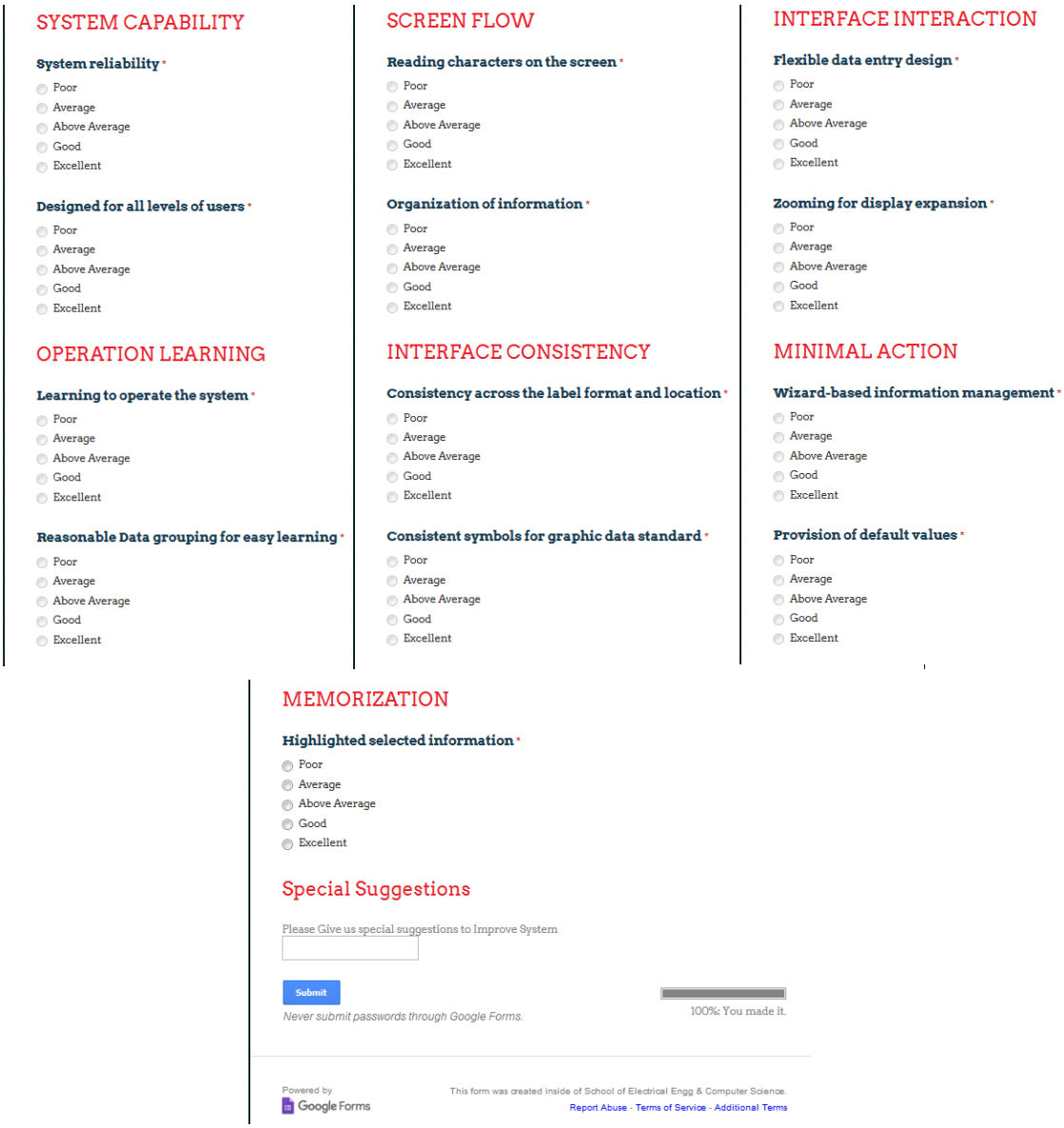


Figure C.2: Users interaction survey form.

C.2 Learning Effectiveness Evaluation

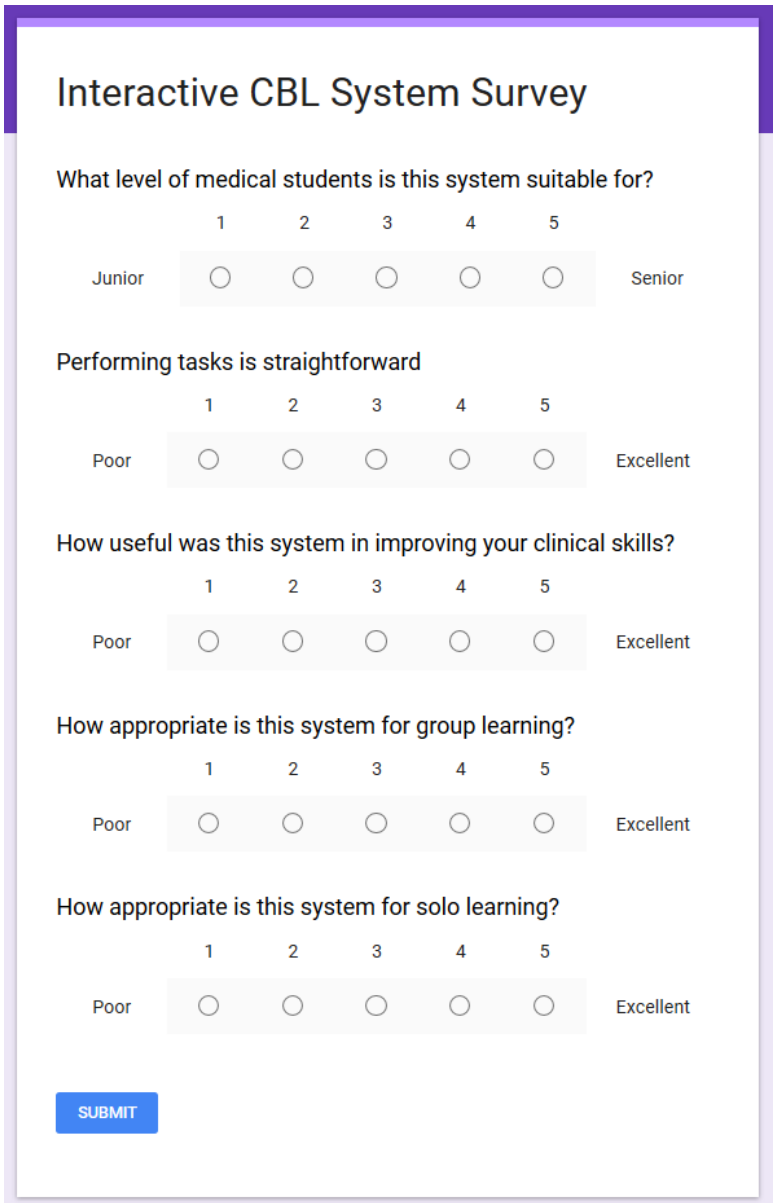


Figure C.3: Learning effectiveness survey form.

Appendix D

List of Publications

D.1 International Journal Papers [8]

1 **Maqbool Ali**, Soyeon Caren Han, Hafiz Syed Muhammad Bilal, Sungyoung Lee, Matthew Jee Yun Kang, Byeong Ho Kang, Muhammad Asif Razzaq, and Muhammad Bilal Amin, "iCBLS: An Interactive Case-Based Learning System for Medical Education", *International Journal of Medical Informatics* (SCI, IF:3.287), Vol.109, pp.55–69, 2018.

2 **Maqbool Ali**, Rahman Ali, Wajahat Ali Khan, Soyeon Caren Han, Jaehun Bang, Taeho Hur, Dohyeong Kim, Sungyoung Lee, and Byeong Ho Kang, "A Data-Driven Knowledge Acquisition System: An End-to-End Knowledge Engineering Process for Generating Production Rules", *IEEE Access* (SCIE, IF:3.244), Vol.6, pp.15587–15607, 2018.

3 **Maqbool Ali**, Hafiz Syed Muhammad Bilal, Muhammad Asif Razzaq, Jawad Khan, Sungyoung Lee, Muhammad Idris, Mohammad Aazam, Taebong Choi, Soyeon Caren Han, and Byeong Ho Kang, "IoTFLiP: IoT-based Flip Learning Platform for Medical Education", *Digital Communications and Networks* (ESCI, Elsevier), Vol.3, pp.188–194, 2017.

4 **Maqbool Ali**, Syed Imran Ali, Dohyeong Kim, Taeho Hur, Jaehun Bang, Sungyoung Lee, Byeong Ho Kang, and Maqbool Hussain, "An Efficient and Comprehensive Ensemble-based Feature Selection Methodology to Select Informative Features from an Input Dataset", *PLOS ONE* (SCIE, IF:2.806), 2018 (Revision submitted).

5 Rahman Ali, Muhammad Afzal, Maqbool Hussain, **Maqbool Ali**, Muhammad Hameed Siddiqi, Sungyoung Lee, and Byeong Ho Kang, "Multimodal Hybrid Reasoning Methodology for Personalized Wellbeing Services", *Computers in Biology and Medicine* (SCI, IF:1.836),

Vol.69, pp.10–28, 2016.

6 Muhammad Hameed Siddiqi, **Maqbool Ali**, Mohamed Elsayed Abdel rahman Eldib, Asfandyar Khan, Oresti Banos, Adil Mehmood Khan, Sungyoung Lee and Hyunseung Choo, "Evaluating Real-life Performance of the State-of-the-art in Facial Expression Recognition using a Novel YouTube-based Datasets", *Multimedia Tools and Applications* (SCIE, IF:1.331), Vol.77, pp.917–937, 2018.

7 Muhammad Asif Razzaq, Claudia Villalonga, Sungyoung Lee, Usman Akhtar, **Maqbool Ali**, Eun-Soo Kim, Asad Masood Khattak, Hyonwoo Seung, Taeho Hur, Jaehun Bang, Dohyeong Kim and Wajahat Ali Khan, "mlCAF: Multi-Level Cross-Domain Semantic Context Fusioning for Behavior Identification", *Sensors* (SCIE, IF:2.677), Vol.17, pp.2433, 2017.

8 Muhammad Idris, Shujaat Hussain, **Maqbool Ali**, Arsen Abdulali, Muhammad Hameed Siddiqi, Byeong Ho Kang, and Sungyoung Lee, "Context-aware scheduling in MapReduce: A Compact Review", *Concurrency and Computation: Practice and Experience* (SCIE, IF: 0.997), Vol.27, pp.5332–5349, 2015.

D.2 Domestic Journal Paper [1]

1 Muhammad Bilal Amin, Muhammad Sadiq, **Maqbool Ali**, and Jaehun Bang, "Curating Big Data for Health and Wellness in Cloud-centric IoT", *The Journal of The Korean Institute of Communication Sciences*, Vol.35, pp.42–57, 2018.

D.3 International Conference Papers [10]

1 **Maqbool Ali**, Sungyoung Lee, and Byeong Ho Kang, "KEM-DT: A Knowledge Engineering Methodology to Produce an Integrated Rules Set using Decision Tree Classifiers", In *12th International Conference on Ubiquitous Information Management and Communication* (IMCOM '18), ACM, 2018. https://doi.org/10.1145/3164541.3164640

2 **Maqbool Ali**, Sungyoung Lee, and Byeong Ho Kang, "An IoT-based CBL Methodology to Create Real-world Clinical Cases for Medical Education", In *8th International Conference*

on Information and Communication Technology Convergence (ICTC 2017), pp.1037–1040, IEEE, 2017.

3 **Maqbool Ali**, Maqbool Hussain, Sungyoung Lee, and Byeong Ho Kang, "SaKEM: A Semi-automatic Knowledge engineering methodology for building rule-based knowledgebase", In *16th International Symposium on Perception, Action, and Cognitive Systems* (PACS 2016), pp.63–64, 2016.

4 **Maqbool Ali**, Jamil Hussain, Sungyoung Lee, and Byeong Ho Kang, "X-UDeKAM: An Intelligent Method for Acquiring Declarative Structured Knowledge using Chatterbot", In *16th International Symposium on Perception, Action, and Cognitive Systems* (PACS 2016), pp.65–66, 2016.

5 **Maqbool Ali**, Sungyoung Lee, and Byeong Ho Kang, "UDeKAM: A Methodology for Acquiring Declarative Structured Knowledge from Unstructured Knowledge Resources", In *2016 International Conference on Machine Learning and Cybernetics* (ICMLC 2016), Vol.1, pp.177–182, IEEE, 2016.

6 **Maqbool Ali**, Hafiz Syed Muhammad Bilal, Jamil Hussain, Sungyoung Lee, and Byeong Ho Kang, "An Interactive Case-Based Flip Learning Tool for Medical Education", In *13th International Conference On Smart homes and health Telematics* (ICOST 2015), pp.355–360, Springer, 2015.

7 Muhammad Hameed Siddiqi, **Maqbool Ali**, Muhammad Idris, Oresti Banos, Sungyoung Lee and Hyunseung Choo, "A Novel Dataset for Real-Life Evaluation of Facial Expression Recognition Methodologies", In *29th Canadian Conference on Artificial Intelligence* (AI 2016), pp.89–95, Springer, 2016.

8 Wajahat Ali Khan, Muhammad Bilal Amin, Oresti Banos, Taqdir Ali, Maqbool Hussain, Muhammad Afzal, Shujaat Hussain, Jamil Hussain, Rahman Ali, **Maqbool Ali**, Dongwook Kang, Jaehun Bang, Tae Ho Hur, Bilal Ali, Muhammad Idris, Asif Razzaq, Sungyoung Lee and Byeong Ho Kang, "Mining Minds: Journey of Evolutionary Platform for Ubiquitous

Wellness", In *12th International Conference on Ubiquitous Healthcare* (u-Healthcare 2015), pp.1–3, 2015.

9 Oresti Banos, Muhammad Bilal Amin, Wajahat Ali Khan, Muhammad Afzel, Mahmood Ahmad, **Maqbool Ali**, Taqdir Ali, Rahman Ali, Muhammad Bilal, Manhyung Han, Jamil Hussain, Maqbool Hussain, Shujaat Hussain, Tae Ho Hur, Jae Hun Bang, Thien Huynh-The, Muhammad Idris, Dong Wook Kang, Sang Beom Park, Hameed Siddiqui, Le-Ba Vui, Muhammad Fahim, Asad Masood Khattak, Byeong Ho Kang, and Sungyoung Lee, "An Innovative Platform for Person-Centric Health and Wellness Support", In *International Conference on Bioinformatics and Biomedical Engineering* (IWBBIO 2015), pp.131–140, Springer, 2015.

10 Jamil Hussain, **Maqbool Ali**, Hafiz Syed Muhammad Bilal, Muhammad Afzal, Hafiz Farooq Ahmad, Oresti Banos, and Sungyoung Lee, "SNS Based Predictive Model for Depression", In *13th International Conference On Smart homes and health Telematics* (ICOST 2015), pp.349–354, Springer, 2015.

D.4 Domestic Conference Papers [5]

1 **Maqbool Ali**, Muhammad Sadiq, Sungyoung Lee, and Byeong Ho Kang, "A Descriptive Knowledge Acquisition Approach to Construct the Machine-Readable Domain Knowledge", In *Korea Computer Congress*, 2018. (accepted)

2 **Maqbool Ali** and Sungyoung Lee, "A Multi-Level Approach for Question Answering to Construct a Structured Declarative Knowledgebase", In *Korea Computer Congress*, pp.844–846, 2017.

3 **Maqbool Ali**, Sungyoung Lee, and Byeong Ho Kang, "An Approach for Classifying Declarative Knowledge Resource to Construct Valid Knowledge base", In *Korea Computer Congress*, pp. 992–994, 2016.

4 **Maqbool Ali** and Sungyoung Lee, "Knowledge Acquisition through Wellness Model for Sedentary Lifestyle", In *Korea Computer Congress*, pp. 975–977, 2015.

5 Sangbeom Park, **Maqbool Ali**, and Sungyoung Lee, "Data collection and Data transfer module for MiningMinds Platform", In *Korea Computer Congress*, pp.455–457, 2015.

D.5 Patents [3]

1 Sungyoung Lee, **Maqbool Ali**, and Byeong Ho Kang, "An IoT-based learning methodology for medical students' education", *Korean Intellectual Property Office*, Registration No. 1018088360000, Date: 2017.12.07.

2 Sungyoung Lee, **Maqbool Ali**, and Byeong Ho Kang, "A Knowledge Engineering Methodology For Decision Trees", *Korean Intellectual Property Office*, Apply: 2016-12-02.

3 **Maqbool Ali**, Sungyoung Lee, and Byeong Ho Kang, "A Univariate Ensemble-based Methodology for Selecting Informative Features", *Korean Intellectual Property Office*, Apply: 2018-05-07.