PERSONALIZED HEALTH INSURANCE SERVICES USING BIG DATA

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Ylli Sadikaj

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Electrical and Computer Engineering

April 2016

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Personalized Health Insurance Services using Big Data

By

Ylli Sadikaj

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Samee U. Khan
Chair

Jacob S. Glower

Ying Huang

Approved:

April 15, 2016
Date

Scott C. Smith
Department Chair

ABSTRACT

Cloud computing paradigm has significantly affected the healthcare sector like various other business domains. Persistently growing healthcare data over the Internet has called for the development of methodologies to efficiently handle the health big data. This study presents a framework that utilizes the cloud computing services to offer personalized recommendations about the most apposite health insurance plans. The users are offered implicit and explicit recommendations.

A standard ontology is presented to offer a unified representation to the health insurance plans. The plans are ranked based on: **(a)** similarities between the users' coverage requirements and the plans **(b)** priority of the cost based criteria in the users' query. The framework overcomes the issues pertaining to the long-tail in recommender systems and propose to cluster plans to reduce the number of comparisons.

Experimental results exhibit that the framework accurately identifies the appropriate health insurance plans that satisfy user's requirements and is scalable.

ACKNOWLEDGEMENTS

First and foremost thanks to GOD Who has helped me throughout the course of my studies. All of my knowledge, strength, courage, health, and abilities are His blessings upon me and there is no way to fulfill His right to thank Him.

Special thanks to Dr. Samee U. Khan, my advisor, for his help, guidance and innovative ideas. I offer my sincere and deep hearted gratitude to my advisor who always encouraged me, and persistently conveyed the spirit and guidance required for the research. Without his kind guidance and continuous efforts, this disquisition would not have been possible.

Special thanks to my committee members, Dr. Jacob S. Glower, and Dr. Ying Huang for their support, guidance and helpful recommendations. Thanks to the Electrical and Computer Engineering staff members Jeffrey Erickson, Laura D. Dallman, and Priscilla Schlenker for all the unconditional help and favor.

I would like to thank my family. Their continuous support is always a source of motivation and encouragement for me. I especially like to thank my parents, my sister, and my brother who are the only and every reason for whatever I am today and whatever I have achieved in my life. They have shown extreme patience and cooperation that resulted in successful completion of this disquisition.

Also, I am very grateful to the United States Agency for International Development (USAID), for this given opportunity to study in the US. As a scholarship recipient of USAID, I am thankful for their help, support and guidance during my MS program.

Finally, I owe my heartiest thanks to all my friends and colleagues here in the US and Kosovo, who always helped me in the time of need.

DEDICATION

I would like to dedicate this thesis to my family, especially to my parents and my siblings for all the inexplicable love, support, and motivation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Introduction to Big Data

The recent growth of use of information and communication technologies has resulted in exponential increase in data volumes over the Internet. Consequently, the need to develop tools and methodologies to search for personalized health insurance plans has increased manifolds [1]. Apart from immense data volumes, the complexity of managing concurrently originating data from multiple sources could not be done by traditional data management tools because they are limited to handle such enormous data volumes. Therefore, the necessity of big-data empowered tools and techniques are required [2]. The same trends of speedy growth of data have also been witnessing in healthcare domain besides the electronic commerce and various scientific domains [2]. The rapid growth of healthcare content has been instigated from various points of care and web-based health communities [3].

Big Data also referred to as Data Intensive Technologies, are becoming a new technology trend in science, industry and business [4]. Big Data are becoming related to almost all aspects of human daily activities starting with social media that people use daily, different sensors obtaining information about human behavior every second, and all the data related to research, problems, solutions and digital services delivery to final consumer. Current technologies such as Cloud Computing and ubiquitous network connectivity provide a platform for automation of all processes in data collection, storing, processing and visualization [5].

Big Data applies to data sets of extreme size such as terabytes, petabytes, exabytes, and zettabytes that are beyond the ability of commonly used software tools to capture, store, and compute within a tolerable timeframe [5]. Based on ability of computing extreme size of data, Big

Data is defined by the 5V Big Data properties: Volume, Velocity, Variety Value and Veracity. The volume refers to the amount of data whereas velocity refers to the speed at which data is being processed or generated [6]. The velocity refers to the speed at which data is created, processed, stored and computed by relational databases [7]. The variety refers to the type of data, mostly of the time the type of data is unstructured which means that the data can be in all shapes and forms. Also the data can be structured data which comprises of database tables and schemas whereas the unstructured data consists of text, audio, and video data. Therefore, big data can simply be considered as large volumes of continuously generated, highly dimensional, and multi-sourced data [7]. Currently, there is wide variety of Web based health related content including the clinical and hospital data, genomics driven data, and social networks data [8]. The value refers to the cost of every step of processing the data in datacenters and opportunities to find solutions to reduce the price of computing the data. The veracity refers to accuracy, reliability and security of the data in Big Data systems. Data can change constantly the meaning and helps developing the Artificial Intelligence that will be very influential in human daily activities in the future. The assumption underlying the effort of Big Data to capture, store and compute data, will be prone to the same quality problems that plague traditionally-sized data sets, characterized by accuracy, precision, completeness, consistency, timeliness, lineage, and relevance [5].

The Patient Protection and Affordable Care Act (PPACA) is the landmark health reform legislation that includes a long list of health-related provisions, which fosters the concept of insurance marketplaces to offer search support for quality health insurance plans [1]. Accordingly, the need to develop tools and methodologies to search for personalized health insurance plans has increased manifolds. Currently, there exist several Web based tools to search for the health insurance products. However, the tools are limited in offering personalized recommendations to users for comparing different health insurance plans from multiple perspectives. The reason for incompetence of existing tools is that currently there exist large numbers of health insurance plans and the existing tools make simplistic comparisons and present users a few insurance plans based on the premium [2]. Moreover, lot of information about health insurance plans is concealed far down on the websites of insurance providers that might not be indexed by the conventional search tools. Therefore, it is imperative to develop methodologies that deeply search the widely scattered and hidden information about health insurance plans and permit users to evaluate the quality of insurance plans using multiple decision criteria. In addition, there is also a need to utilize the hidden information to extract the meaningful information about plans by using the semantic Web techniques because different healthcare insurance providers do not use same terms for exact the same criteria. Thus, there is a need to unify these information and to make it simple for users to understand providers' information about healthcare insurance plans. Also it would make much easier to compare two healthcare insurance plans from different healthcare insurance providers.

1.2. Plan Recommendations and Research Contributions

In this study, we present a framework that helps users in identification of best suited health insurance plans based on user defined requirements. This work is enhancement of the previous work presented in [2] that allows users to evaluate the health insurance plans based on various coverage requirements and cost based criteria. The framework presented in this study offers implicit and explicit recommendations about health insurance plans based on the popularity of the plans and user stated requirements, respectively. The framework provides implicit recommendations to the users about the popular plans even without specifying the requirements. Likewise, explicit recommendations are provided on the basis of the users' coverage requirements and cost based criteria. Moreover, the framework uses a Web crawler to retrieve plans information about health insurance plans from the websites of insurance providers and subsequently transforms them into ontology.

The similarities between the users' coverage requirements and health insurance plans are determined and Rank Order Centroid (ROC) method is used for weight assignment to different criteria specified by the users. However, as in the framework presented in [2], comparing one user's coverage requirements with the entire list of plans for a particular category is compute-intensive and may result in increasing the response time for real-time query processing. Therefore, in this study we presented a methodology that reduces the number of comparisons. The methodology proposes to cluster the plans in a way that the plans having less ranking distance are clustered together and the user's coverage requirements are mapped to the plans of only that cluster whose ranges of premium, copay, deductibles, and out-of-pocket limit are closer to the corresponding values indicated in the user query. In fact, clustering of the plans and identification of the most appropriate cluster for comparisons with the users' queries not only contribute to obtain personalized recommendations but also minimize the number of comparisons. Therefore, we argue that the selection of clustering technique also plays vital role in achieving the desired recommendation accuracy. To this end, we appraised the performance of four clustering techniques namely: **(a)** Voronoi diagram based clustering [9], **(b)** Density-based Spatial Clustering of Applications with Noise (DBSCAN) [10], **(c)** Fuzzy C-means clustering [11], and **(d)** Bayesian Hierarchical Clustering [12]. Experimental results show that the clustering approaches based on the DBSCAN and Voronoi diagram achieved higher accuracy as compared to the other two approaches. Moreover, in this study, we also present a procedure that updates the popularities of different insurance plans based on the initial popularity and ranking score for each of the plans.

The framework utilizes the cloud computing services to deal with huge volumes of data. As the coverages and prices for plans are different across the states, the framework maintains separate plan repositories for each of the states. Because there are large numbers of insurance plans offered in each of the geographical areas, it requires high-end computing and storage services to handle the constantly growing data and to make the task of plans retrieval more efficient. Therefore, cloud computing seems quite suitable to manage the health insurance big data. Moreover, the task of implicit recommendations is precomputed in offline mode.

The major contributions of this study are as follows:

- A cloud based framework is presented to help users evaluate different health insurance plans according to four different criteria, such as premium, copay, deductibles, and out-of-pocket limit.
- The framework offers implicit and explicit recommendations about health insurance plans. Moreover, a standardized representation for health insurance plans is presented.
- A methodology to calculate the initial rank of each of the plans is presented. Initial ranking scores are needed to offer implicit recommendations in the start when there are no users in the system.
- We propose to cluster the health insurance plans to minimize the number of comparisons between the users' queries and the actual plans offered by health insurance providers.
- A ranking approach is presented that utilizes the similarity scores, weights assigned to the user defined criteria, and satisfiability measure to determine the rank of each of the plans.
- A procedure to avoid the long-tail issue of the recommender systems is presented. The popularities of the plans are updated frequently to help newly introduced plans emerge as the popular plans.

- Cloud computing services are utilized to simultaneously process health insurance plans data in different geographical areas. Jobs are executed in parallel manner to handle the huge data volumes and to support simultaneous real-time queries by multiple users.
- We also present the scalability analysis of the framework and evaluate the performance by increasing the workload and resources, such as the number of processors.

The remainder of the study is organized as follows. Motivation for the proposed works is presented in Chapter 2. Chapter 3 presents the architecture of the proposed system and also discusses the implicit and explicit recommendation methodology. Chapter 4 presents the prototype of framework implementation. Discussion, results, and related work are presented in Chapter 5 whereas Chapter 6 concludes the study.

2. BACKGROUND AND MOTIVATION

2.1. Big Data on Healthcare

In recent years, substantial technological advancements have been witnessed in the healthcare sector that have led to creation and exchange of large volumes of healthcare data over the Internet. Huge volumes of healthcare content is being generated every day from multiple sources, such as hospitals, clinics, clinical laboratories, health insurance providers, and pharmacies [13]. Consequently, the data originated from several sources evolves as the health big data.

Typically, the term big data has three defining properties namely, volume, velocity, value, veracity and variety. Further growth of business data over the Internet in general and healthcare data in particular is expected in coming years. Moreover, conventional methodologies used to store and process huge data volumes seem ineffective and therefore, the need for development of tools and models capable of supporting parallel execution of multiple tasks becomes more obvious. Cloud computing paradigm is among one of the models that due to its key characteristics, such as cost-effectiveness, scalability, agility, and on-demand service provisioning has the ability to manage and process the big data [14]. Besides various business and scientific domains, the healthcare sector has also started using the cloud computing services.

2.2. Heterogeneity of Healthcare Data

The inclination of the healthcare organizations towards the cloud computing is due to the fact that the model liberates the organizations of the tasks of infrastructure management and development [15]. Moreover, the cloud computing enables various participating organizations, such as hospitals, clinics, pharmacies, and insurance companies to exchange electronic health data in a convenient way [14].

Nonetheless, there is a need to enhance collaboration among the participating entities in a way that evolves a monolithic health ecosystem. Particularly, the role of health insurance providers needs to be extended beyond the claims processing so that they could emerge as the key players of the cloud based e-health systems [2]. In this study, we present a framework that offers implicit and explicit recommendations about health insurance plans based on the popularities of the plans and user stated requirements, respectively. The framework retrieves information about different health insurance plans from the webpages of different health insurance providers. However, the retrieved information is highly heterogeneous both in terms of semantics and syntax. Semantic heterogeneity arises when the interpretation of the same concept though represented differently across the systems is similar [16]. In the health insurance scenario, semantic heterogeneity refers to different terminologies used by different healthcare providers. For example, across one provider for different categories of drugs, such as general, brand formulary, brand non-formulary, and specialty drugs, the equivalent terms used by other insurance provider may be Tier 1, Tier 2, Tier 3, and Tier 4 drugs, respectively [17]. Consequently, unification is desirable for semantically related data such that a standardized representation for different health insurance terminologies and plans is offered. Syntactic heterogeneity means that the health insurance data available on the Web is stored in different formats. The semantic or structural heterogeneity can be overcome through a standardized ontology [18]. The standardized health insurance ontology can offer a uniform representation to all of the health insurance plans being offered by several providers.

3. PERSONALIZED HEALTH INSURANCE RECOMMENDATION SERVICES USING CLOUD COMPUTING[1]

3.1. Introduction to System Architecture of the Framework

The architecture of the proposed system comprises of the following modules: **(a)** plans retrieval and ontological transformation module and **(b)** health insurance plans recommendation module that further comprises of implicit recommendation and explicit recommendation modules. Fig. 1 illustrates the architecture of the proposed system.

3.2. Plans Retrieval and Ontological Transformation Module

As stated earlier that currently there are large numbers of medical and dental insurance plans that have been shortlisted as the qualified plans under the insurance marketplaces. As an example, over 78,000 medical plans and around 45,000 dental insurance plans [2] have been identified for the insurance marketplaces. In addition, there are also other plans that are being offered by several other insurance providers. Moreover, the aforementioned numbers are expected to increase after the complete implementation of the PPACA. Consequently, it is indeed a challenge for the contemporary comparison tools to offer personalized recommendations according to the

[1] This paper is submitted to *IEEE Transactions on Big Data (TBD)* and is in the second round of interview. The material in this chapter was co-authored by Assad Abbas, Usman Shahid Khan, Aziyati Yusoff, Ylli Sadikaj, Jamin Ashley, and Samee U. Khan. Ylli Sadikaj had primary responsibility for developing the standardized representation of the health insurance plans and computing the ranking score by developing algorithm. Ylli Sadikaj also contributed on drafts and revisions of this paper.

diversified requirements of users. The reason is that lot of information is hidden and unindexed and therefore, the search engines are not able to locate such information.

In this study, we used a Web crawler that crawls through the webpages of the insurance providers and retrieves information about health insurance plans. However, the retrieved pages comprise of large volumes of unstructured information, which is not directly usable. Moreover, it is difficult to deduce the meaningful information from the aforementioned data. Consequently, it requires certain means to represent the information in a standardized way. Ontology and semantic Web tools allow the development of standardized vocabularies and uniform structural representation of heterogonous data. Semantic Web aims to extend the capabilities of the current Web by giving well-defined meanings and correct interpretation of the information through expressive rules [19]. From the health insurance perspective, the data is unstructured and heterogeneous both in terms of interpretation of the concepts and the underlying knowledge representation. In this regard, we have developed ontology to unify the semantically related data from multiple sources. Ontology is defined as a specification of the conceptualization [20]. In fact, the ontology comprises of a standardized vocabulary and defines a precise view of a domain. Considering the health insurance as a complete domain, ontology exhibits potential to overcome the structural and syntactic heterogeneity [15]. As there are large numbers of plans for diverse categories of users with different coverages, the relationships can effectively be represented through the ontology.

Currently, there is no global ontology for health insurance terms. Therefore, we propose a generic ontology encompassing rich insurance terms. Standard health insurance ontology will not only be beneficial for the providers in offering a standardized representation of plans but will also be useful for the masses to have the unified comparative information about multiple plans readily available at a single point. We used Web Ontology Language (OWL) to develop the ontology [21]. The OWL enables greater machine interpretability of Web content as compared to the XML, RDF, and RDF Schema (RDF-S) through additional vocabulary besides the formal semantics [21]. Fig. 2 shows the asserted model of the health insurance ontology using the Protégé 5.0. The classes and subclasses presented in the ontology depict that there exist complex relationships between the classes that vary on the basis of age, family size, geographical area, and various other factors. The proposed framework also ensures the provision of most recent and updated information at all the times and considering that the plan coverages and other supplemental benefits change over time, periodic jobs are executed to retrieve information from the webpages of providers.

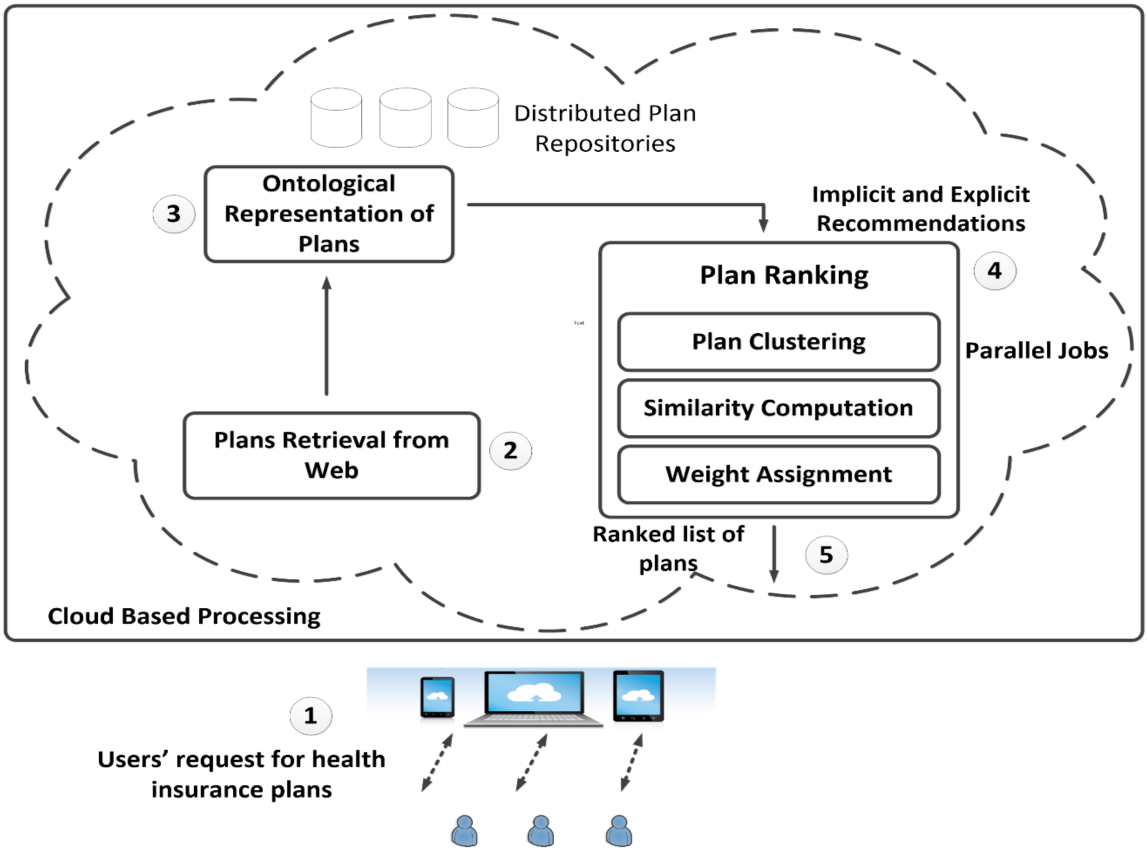

Figure 1. Architecture of proposed cloud based framework.

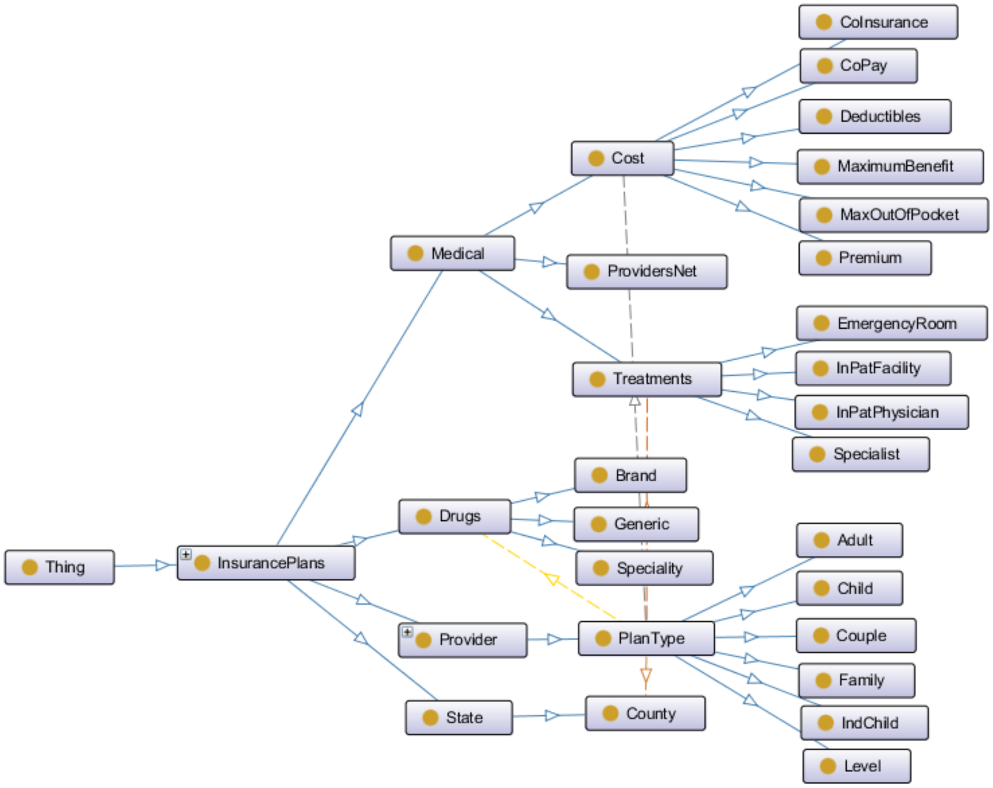

Figure 2. Asserted model for the health insurance plan ontology.

3.3. Health Insurance Plans Recommendation Module

The health insurance plan recommendation module offers implicit and explicit recommendations about health insurance plans. Implicit recommendations are offered based on the popularity of the plans whereas explicit recommendations are offered on the basis of users' requirements in terms of cost and coverage. As the coverage and prices of the plans are different across the states. Therefore, our proposed framework maintains separate plan repositories for different types of plans being offered in different geographical areas. Besides the coverage requirements, the users also indicate the priorities for the four cost based attributes or criteria over which the recommendations are made. The cost criteria include premium, copay, deductibles, and maximum out-of-pocket limit of a plan. The procedures of implicit and explicit recommendation are presented below. Table I presents the definitions of the symbols used throughout the study.

Table 1. Symbols and Definitions

Symbol	Meaning	Symbol	Meaning
C	Cost based criteria	S	cluster
P_r	Premium	T	Set of trees
D_r	Deductibles	E	Set of edges
CP_r	Copay	σ	Root node
OP_r	Out-of-pocket limit	ϕ	Labeling function
$\tilde{\mathrm{I}}$	Initial rank	Ꞧ	Requirements tree
w_i	Weight assigned to each attribute	Ƥ	Plan tree
I_a	Combined popularity	δ_{r_i}	Satisfiability measure
γ	Cluster size	μ	Desired value
ρ	Actual value		

3.3.1. Implicit Recommendation Module

The new users are recommended health insurance plans both implicitly and explicitly. We consider those recommendations as the implicit recommendations that are provided to users whenever they first interact with the system even without stating the coverage and cost requirements. Implicit recommendations are offered based on the popularity of plans. It is worth mentioning that initially the system does not contain users. Therefore, the users who access the system in the very start may not be able to obtain the implicit recommendations. Moreover, the coverage requirements for this module cannot be obtained because at first there are no users who can specify the requirements. Therefore, to overcome the cold start issue, we first computed the initial popularity of insurance plans by computing all of the possible combinations for the four decision criteria or attributes and then assigned weights according to the importance of each of the specified criteria. For implicit recommendations, we only consider the cost based attributes. Consequently, the framework determines the cost based requirements and computes the initial popularity by assigning weights to different cost based criteria. The procedure to calculate initial ranking is explained below.

We denote a set $C = \{P_r, D_r, CP_r, OP_r\}$ that represents various cost based criteria. The higher the importance of the criteria or attribute, the more weight is assigned. Because we have four attributes in C, there can be 24 different ways in which cost based attributes can be arranged with different priorities. For weight assignment, we used the ROC method. In the ROC method, the weights to attributes are assigned on the basis of their relative importance as compared to the other attributes. Suppos$e$ $\acute{\mathrm{M}} = \{m_1, m_2, \dots, m_n\}$ represents a set of plans initially stored in the plan repositories. For each $m \in \acute{\mathrm{M}}$, we select an elements of C as the cost requirement, if it has minimum value amongst all of plans. For each particular plan according to predefined criteria the initial rank $\tilde{\mathrm{I}}$ is computed as below:

$$\tilde{\mathrm{I}}_i = \sum(C_i \times w_i) \qquad (1)$$

Where C_i represents a particular cost based criteria and w_i denotes the weight assigned to each of the criteria according to the ROC method. The weights are calculated according to the following equation:

$$w_i = \left(\frac{1}{k}\right)\sum_{n=i}^{k}\left(\frac{1}{n}\right) \qquad (2)$$

Where k represents the total number of decision attribute or criteria and w_i is the weight assigned to i-th attribute. Consequently, the popularity scores obtained from Eq. 1 are considered as the initial rank of a particular plan in the absence of explicitly stated coverage requirements in the start. However, once the users start using the system frequently, the popularities are updated such that the scores are based on both the ranking score of the plan and the previous popularity of the plan. Algorithm 1 lists the steps to determine the initial rank of each of the plans in the absence of explicit coverage requirements stated by the users. Line 1 of Algorithm 1 computes all of the possible combinations in which the elements of C can be arranged. Line 3 computes the minimum values for each of the plans with respect to all of the possible combinations of the elements of C. Line 4 assigns weights to the elements of C using the ROC method for each $\mu_i \in \mu$. Line 5 calculates the initial popularity based on line 3 and line 4. Line 6 clusters the plans and line 7 calculates the ranges of each element of C for each cluster. Line 9 returns the combined popularity of a plan.

Algorithm 1: Initial rank calculation

Input: Set of criteria *C*, cluster size
Output: Combined popularity I_a
Definitions: C= set of criteria, γ= cluster size, I_a= combined popularity

```
1. $\mu \leftarrow computeCombinations(C)$
2. PARFOR each $m \in \mu$
3. $X \leftarrow computePlanMin()$
4. W ←assignWeight(X)
5. $\bar{\iota}_{pop} \leftarrow InitialPopularity(X, W)$
6. ¥ ← $Clusters\ (m, \gamma)$
7. $Q \leftarrow calculteClusterRange$(¥)
8. end PARFOR
9. Return $combinedPop\ (I_a\ )$
```

3.3.2. Explicit Recommendation Module

Another important module of the framework is plan recommendation against the explicitly stated requirements of users. The users specify their desired cost and coverage requirements and the framework generates recommendations that best suit the users. The initial popularity scores computed for the plans serve as the basis for the explicit recommendation module. As there are large numbers of health insurance plans being offered in each of the geographical areas. Therefore, it requires reasonably large number of comparisons between the user requirements and the plans offered by health insurance providers. Comparing the explicitly stated requirements indicated by each user with all other plans can be computationally expensive.

Therefore, we need to reduce the comparison space such that the users' stated requirements are satisfied to the maximum level. To this end, our methodology at first utilizes the initial popularity scores of the plans and clusters the plans according to their distance from each other while considering the values indicated in the user query for each $c_i \in C$. Subsequently, the plans matching with the user defined coverage requirements and cost based criteria are compared with the plans from the cluster that best matches with the user defined criteria. The clusters are periodically updated because the popularity rankings may change frequently after the user rankings are generated. The process of identifying the appropriate cluster is explained below.

Based on the initial popularity of each of the plans for 24 different combinations, the plans are clustered together such that each cluster contains the plans that are closer to each other in terms of their ranking score. There are several clustering algorithms that can be used to cluster the plans. However, to demonstrate the efficacy of clustering the plans in reducing the number of comparisons, we used the clustering algorithms namely: **(a)** Voronoi diagram based clustering, **(b)**

Density-based spatial clustering of applications with noise (DBSCAN), **(c)** Fuzzy C-means clustering, and **(d)** Bayesian Hierarchical clustering.

Each cluster contains several plans based on the ranking score for each of the combinations. Therefore, mapping the user's coverage requirements to each of the plans in all of the clusters is extremely compute-intensive and cannot be considered realistic for real-time queries where nominal response time is expected. To evade the overhead, our approach compares user's requirements with the plans of the identified cluster only. In each cluster $s_i \in S$, the ranges of each of the elements of C are calculated for all of the possible combinations. As already stated that the users indicate both the priority of each element of C and the coverage requirements. Therefore, we first used the cost based criteria indicated by the user to identify the most suitable cluster for comparison with the user's coverage requirements. The values for each of the $P_r, D_r, CP_r,\ and\ OP_r$ in the user query are compared with the values of clustered plans for the corresponding attributes. If the values of the elements of C are within the user defined criteria, then s_i has the potential of being selected as the appropriate cluster.

However, there is possibility that not all of the cost based criteria indicated by the user are satisfied in user's query. Therefore, we defined the criteria for selection of the cluster. According to the selection criteria, for any combination of C specified in the user query, if there exists any cluster $s_i \in S$ whose plans range within the values of each element of C specified in user query, then s_i is selected as the appropriate cluster. However, the situation where the entire user stated criteria are satisfied by the plans of a particular cluster may not occur all of the times. Therefore, we specify the cluster selection criteria where at least three of the attributes of C stated in the user query are satisfied. For example, $C = \{c_1, c_2, c_3, c_4\}$ represents the priorities for different cost based criteria indicated by a certain user, where c_1 has the highest priority and c_4 has the least priority. If the values of any three elements of C are within the range of the plan values of a particular cluster then that cluster is considered as the appropriate cluster for subsequent comparisons. However, in this case we make selection decision based on the importance of attributes in the user query. Therefore, the cluster whose plans satisfy the criteria $c_1, c_2,$ and c_4 is selected as the appropriate cluster because this cluster satisfies the two top most user specified criteria $c_1$and c_2. Likewise, clusters satisfying $\{c_1, c_3, c_4\}$ and $\{c_2, c_3, c_4\}$, respectively are the next potential candidates of being selected as the appropriate clusters. However, if the plans in a cluster satisfy only two criteria then that cluster is not selected.

a) Tree Matching and Plan Ranking

Once the appropriate cluster has been identified, the user's coverage requirements are mapped to the plans of the selected cluster to compute the similarities. The similarity scores are subsequently used to calculate the ranking for each plan. The procedure for similarity computation and ranking the plans is discussed below.

Both the health insurance plans and the user requirements are represented as XML schemas to determine similarities between the user's coverage requirements and the coverage benefits offered by the plans. The XML documents are thereafter represented in the form of labeled trees. Using the XML, a whole document can be represented as a root node of the tree [22]. The nodes in a traditional Document Object Model (DOM) represent the XML elements that are labeled with the corresponding tags. The elements are represented in the same order in the tree as they are represented in the corresponding XML documents. The preliminary concepts and definitions for labeled trees in context of the proposed scenario are presented below.

A tree for exact matching is defined as $T = \{N, E, \sigma, \phi\}$, where N is a set of finite set of tree nodes represented as $N = \{n_1, n_2, \dots, n_k\}$, and $E = \{e_1, e_2, \dots, e_k\}$ is a set of edges between the nodes of the labeled tree. The root node of the tree is symbolized as σ and ϕ is labeling function that is used to map each node to a set of labels $L = \{l_1, l_2, \dots, l_k\}$.

As stated earlier that the framework permits the users to specify their coverage requirements in addition to the prioritized criteria namely the premium, copay, deductibles, and maximum out-of-pocket limit. Therefore, the first step to compute the ranking score for each plan in the proposed framework is similarity computation between the user coverage requirements and the actual plan coverage offered by different providers. For similarity computation, we adopt the same approach as was used in the previous approach presented in [2]. Suppose Ʀ, and Ƥ represent the user requirements and plan trees, respectively. Ƥ_k Represents each single plan in Ƥ. We calculate the similarities between Ʀ and Ƥ through exact tree matching based approach. The exact tree matching compares two trees while preserving the ancestry such that if the label of node in tree Ʀ, is matched with the label of node at the corresponding level in Ƥ, only then the descendants of Ʀ will be compared to the descendants of Ƥ. It is important to mention that the tree matching algorithm computes only the structural similarities between the requirement trees and plans trees because the requirement tree only contains coverage requirements. The proposed tree matching algorithm compares each node in Ʀ to every node in Ƥ at the same level under the same parent irrespective of the order of the nodes in Ƥ. The structural similarities between Ʀ and Ƥ are computed as below:

$$Sim\,(\text{Ʀ}\,,\text{Ƥ})_i = \frac{\text{Ǘ}_i}{\text{Ţ}_i} \tag{3}$$

Where Ǘ_i represents the number of coverage requirements that are fulfilled by a plan and Ţ_i is the total number of requirements specified by the user. The maximum value for the similarity function $Sim\,(\text{Ʀ},,\text{Ƥ})_i$ for a user i is equal to 1. Computing only the structural similarities between

Ɽ and Ƥ does not guarantee that the users would be returned the most appropriate health insurance plans. The reason is that there might be several plans that offer coverage for the requirements indicated by the user. However, the costs, for example premium, copay, deductible, and maximum out-of-pocket limit may be significantly high from the users' cost expectations. Therefore, the users should be offered the choice to specify the priority or importance of each of the cost based criteria.

To this end, our framework permits users to specify the importance of cost based requirements and to evaluate the health insurance plans from multiple aspects of cost. We used Multi-attribute Utility Theory (MAUT) [23] to help users specify the importance of different decision criteria. The MAUT is an approach that involves users in decision making on the basis of multiple objectives that are independent of each other. The final ranking of a particular plan is calculated as below:

$$Rank_i = ((Sim\ (\text{Ɽ}\,, \text{Ƥ})_i) \times (\sum(W_i\ \times\ \delta_{r_i})\ )) \qquad (4)$$

Where W_i represents the weight assigned to each attribute and δ_{r_i} is the measure used to determine the satisfiability of a particular cost based requirement. The measure is defined as follows:

$$\delta_{r_i} = \frac{\mu}{\rho} \qquad (5)$$

Where μ and ρ respectively are the desired value of a particular cost based criteria and the actual value of that attribute in the plan offered by the health insurance provider. The measure is important to be considered because most of the times it may not be possible that the health insurance plan offered by the provider contains the same values as indicated in the user's query. The measure has maximum value equal to 1, when the values indicated in the user query and the plan are the same. However, if the requested value of any of the elements of C is less than the actual value for that attribute in the health insurance plan, δ_{r_i} is still considered as 1. The algorithm to compute the similarity and ranking is presented as Algorithm 2.

Algorithm 2: Trees similarity calculation and plan ranking

Input: user requirement tree Ʀ, plan tree Ƥ, and set of criteria C
Output: Ranking score for each $p \in$ Ƥ
Definitions: ή=matching nodes count, ℵ=non-matching nodes count, W=weight assigned to each element of C, δ_{r_i}= satisfiability of a user defined criteria, $Rank$=Rank of a plan, l=node lablel.

```
1: Procedure Sim (Ʀ,, Ƥ)
2: ή ← 0
3: ℵ ← 0
4:     if (Ʀ. l==Ƥ. l) then
5:         ή ← ή + 1
       end If
6: PARFOR each Ʀ_child, Ƥ_child of Ʀ and Ƥ
7:     x ← Sim (Ʀ_child, Ƥ_child)
8:       if (x>0) then
9:           ή ← ή + x
10:    else
11:          ℵ ← ℵ + 1
12:      end if
13: end PARFOR
14: ₲ ← getSimScore (ή, ℵ)
15: Rank = (₲)*( W *δ_r_i)
16: Return Rank
```

Algorithm 2 computes the ranking of plans based on user's explicitly stated requirements. Line 1 of Algorithm 2 provides the requirement tree and plan tree to the algorithm as input. Line 4—line 13 compute the matching nodes between the user requirements tree and the plan tree. For each node in the user requirement tree, the plans tree is exhaustively searched at the corresponding level and if a match between the nodes of two trees is found, subsequent levels are matched. Line 6—line 9 compute the matching nodes of each of the child Ʀ and Ƥ. Line 11 computes the non-matching nodes. Line 14 calculates the total similarity score based on the matching and non-matching nodes. Line 15 calculates the ranking score for the plan based on the similarity score, weights assigned to the criteria, and the satisfiability measure. Line 16 returns the final ranking score.

3.3.3. Plan Popularity Calculation

As stated earlier that the framework offers implicit recommendations to the users about the popular plans. The popularity scores of the plans are updated frequently. The popularities are updated because there is possibility that the plans identified as popular since start may always be recommended to the users while there exist certain other plans that are not as better as the popular plans but still are substantially competitive. Moreover, there might be certain other newly offered plans that obviously have low popularity in the start and may not be recommended to the users due to existence of popular plans. The aforementioned problem in recommender system is termed as the long-tail problem where popular items are recommended frequently and the unpopular items are ignored. Overcoming the long-tail problem is of significant importance because in e-commerce the long-tail items result in higher profits due to the fact that popular items bring very less profits because of the competitive environment. For instance, Amazon earns most of the profit not from the best-selling products, but from the long tail items [24]. In the scenario of health insurance plans recommendation, we solve the issue by considering both the popularity of a plan and the ranking score of that plan. For each user query, the popularity is updated such that the popularity of plans with low ranking scores also improves with time. The algorithm to update popularity score is presented as Algorithm 3.

Algorithm 3: Popularity calculations

Input: Set of plans = P, Rank R, and Initial popularity *InitPop*
Output: Updated popularity score for each of plan
Definitions: *P* = Set of plans, R = Rank of each plan , *InitPop* = initial popularity,
TempRank = Temporary rank

```
1.  Procedure  PopCalc( P)
2.  PARFOR each $p \in P$
3.     If ($InitPop_p$==$null$) then
4.        $InitPop_p \leftarrow 1$
5.     end if
6.     $TempRank_p \leftarrow (R * InitPop_p)$
7.  end PARFOR
8.  $A \leftarrow selectTopK(TempRank)$
9.  $Y^c \leftarrow \{A \in TempRank | A \notin Y\}$
10. PARFOR each p ∈ P
11.     If ($p \in A$)
12.        $InitPop_{p_1} = InitPop_{p_0} + 1$
13.     else
14.        $InitPop_{p_1} = InitPop_{p_0} + 1/2$
15.     end if
16. $UpdatePop_p \leftarrow (\frac{1}{n}\sum_{j=1}^{n} InitPop_{p_j})$ , where n = 2
17.   end PARFOR
```

From line 3—line 5 of Algorithm 3, it is determined whether a plan has initial popularity or not. If the initial popularity is null, then the plan is assigned initial popularity equal to 1. In line 6, temporary rank of each of the plans is computed based on the rank of the plan and the initial popularity. Line 8 identifies *Top-K* plans with the highest temporary rank whereas the remaining plans are identified in line 9. In line 11 it is determined whether a plan is included in the list of *Top-k* plans. If the plan is present then the initial popularity score is incremented by 1 in line 12, otherwise it is not incremented. Line 16 updates the popularity scores and the rank of the new user is multiplied with the updated popularity score. Consequently, popularity scores are updated for each plan and therefore, the implicit recommendation process cannot identify a few typical plans as the popular plans. Instead the popularities of the plans are frequently updated and even the newly entering plans emerge as the popular plans.

4. PROTOTYPE FRAMEWORK IMPLEMENTATION

4.1. Implementation of Cloud Computing Framework

We implemented the prototype of the proposed framework to help users provide personalized recommendations about the health insurance plans. A Software as a Service (SaaS) implementation of the framework enables to effectively handle large volumes of health insurance plans data. The SaaS model enables the software to be hosted as a service where the users access the services through a browser [25].

The framework performs offline processing to maintain distributed repositories of initially ranked health insurance plans offered in different geographical areas. Periodic jobs are executed to fetch information from the webpages of the providers and to subsequently transform the retrieved information to ontology. The real-time user requirements are mapped to plan trees to determine the similarities between the users' elicited requirements and actual plans. The initial ranking process is performed offline because the initial ranks of plans in the absence of explicit user requirements are preprocessed. Moreover, there are 24 different combinations for the four cost based criteria, which is difficult to compute in real-time. Therefore, computing the initial rank scores offline reduces the overheads of real-time query processing.

We conducted the experiments on our local cloud computing setup equipped with Supermicro SuperServer SYS-7047GR-TRF systems. Fig. 3 presents the mapping of the proposed framework cloud environment. As we can see in Fig. 3 there are two modules, one is realized online mode and another one can be done offline mode. The steps which are computed in offline mode are computation of possible combinations, calculation of plan minimum, and initial ranking computation. These steps are computed in offline mode because there is not necessary to have an interaction from the user to do the initial calculations of the health insurance plans. The initial calculations are done based on comparison of the cost of each health insurance plan. To be able to give recommendations is necessary to have an interaction with the user, which is done when user specifies his/her requirements. Based on users' requirements, the steps which are calculated in online mode are plan clustering, defining a cluster ranges, mapping cost based criteria on clusters, identifying the appropriate cluster, and similarity computation. After being able to compute a plan ranking module, then each plan has a ranking score based on steps which are done in online and offline mode. The ranking score is used to give explicit recommendations and also it is used to update popularity of the plans. Based on popularity of the plans, the framework is able to offer implicit recommendations.

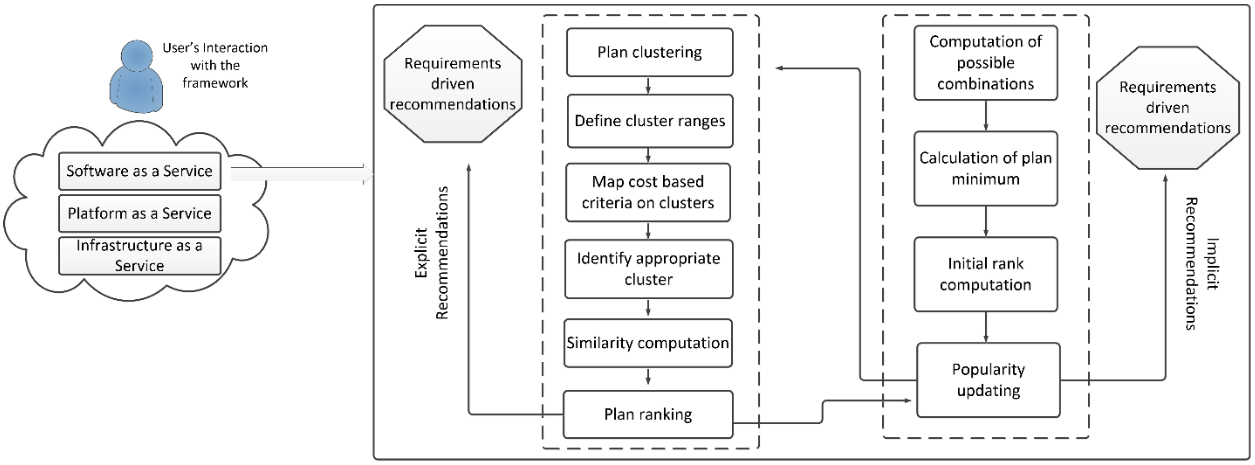
User's Interaction with the framework
Software as a Service
Platform as a Service
Infrastructure as a Service
Requirements driven recommendations
Explicit Recommendations
Plan clustering
Define cluster ranges
Map cost based criteria on clusters
Identify appropriate cluster
Similarity computation
Plan ranking
Computation of possible combinations
Calculation of plan minimum
Initial rank computation
Popularity updating
Requirements driven recommendations
Implicit Recommendations

Figure 3. Cloud service mapping of the proposed framework.

5. RESULTS AND DISCUSSION

5.1. Evaluation Process of Cloud Computing Framework

The effectiveness of the approach was evaluated in terms of cluster identification and scalability of the framework to handle variable workloads. The evaluation results are discussed below.

5.2. Evaluation of Cluster Identification Process

In the proposed framework, identification of appropriate cluster for comparison is among one of the key tasks that affects the recommendation accuracy of health insurance plans. As stated earlier in Chapter 3 that based on the priorities of four cost based criteria laid down by the users, we identify the cluster that best matches the user defined cost based criteria. Once the cluster is identified, the user's coverage requirements are compared with each of the plans present in that cluster. Because there are large numbers of plans offered in each geographical area, therefore, it requires enormous computational resources to compare one user's coverage requirements with multiple plans. However, clustering plans with respect to their closeness to the other plans and defining ranges for each of the cost based criteria, such as premium, copay, deductibles, and out-of-pocket limit reduces the number of comparisons. The user's coverage requirements are compared with only those plans in a particular cluster that are most relevant to the cost criteria specified in the user query. Consequently, correct identification of cluster significantly impacts the recommendation accuracy. Therefore, we compared the accuracy of the proposed framework with the approach presented in [2]. The previous approach makes exhaustive comparisons of one user's requirements with multiple plans.

To evaluate the effectiveness of clustering based approach, we compared the following clustering techniques: **(a)** Voronoi clustering, **(b)** Density-based spatial clustering of applications with noise (DBSCAN), **(c)** Fuzzy C-means (FCM), and **(d)** Bayesian hierarchical clustering (BClust). There are several other clustering approaches that can be used to cluster the plans. However, in this study we intend to demonstrate that clustering not only maintains the accuracy sufficiently but also reduces the number of comparisons between the users' requirements and the actual health insurance plans offered by the providers. Brief description of each of the compared techniques is presented below.

The Voronoi diagram partitions a plane into different regions or cells based on the distance from some particular points [9]. In the proposed scenario, the Voronoi diagram clusters the health insurance plans into separate regions such that the plans having least distance from each other are clustered together.

The DBSCAN is a density based algorithm that constitutes arbitrary shape clusters in spatial databases. The algorithm defines a cluster based on the number of density connected points. The performance of the DBSCAN degrades for highly dimensional data. In other words, if there are data with high differences in density then algorithm can cluster the data effectively [10].

The Fuzzy C-means clustering is also a popular clustering methodology used in pattern recognition and various other domains. However, the methodology randomly selects the center points and due to that reason it is easily trapped into local minimum [11].

The Bayesian hierarchical clustering is a probabilistic algorithm that utilizes the marginal likelihood to merge the clusters and to avoid overfitting. Each data point is initialized in its own cluster and pairs of clusters ae merged iteratively [12]. However, the approach has several limitations including the greediness and the quadratic time complexity.

Common model evaluation metrics namely, the precision, recall, and F-measure [8] were used to evaluate the performance in terms of accuracy. Precision in the presented scenario is the ratio of correctly identified (True Positives) health insurance plans in a cluster to the total plans (True Positive (TP) + False Positive (FP)), given as:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

Recall is the identification probability of being selected for a plan from the entire training set and is given as:

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

F-measure combines the precision and recall and is the harmonic mean of both the precision and recall.

$$F - measure = 2 * \frac{\text{precision}*\text{recall}}{\text{precison}+\text{recall}} \tag{8}$$

The information about the health insurance plans was retrieved through the crawler and was transformed into a standardized representation with a common vocabulary for each of the plans. The plans were subsequently stored as XML trees. After calculating the ranking score for each of the plans, the plans were clustered based on their distance from each other. It is important to mention that calculation of ranking score and subsequent clustering were performed according to 24 different combinations of the four elements of C. The plans clustered in each cluster were compared with the plans retrieved using the approach in [2] for the user query. Approximately 200 health insurance plans being offered in one geographical area were used to evaluate the accuracy of the clustering based approach and the approach in [2]. The accuracy was determined on the basis of occurrence of clustered plans in the list of plans retrieved through the exhaustive search approach. The precision, recall, and F-measure scores for each of the four clustering methodologies are presented in Fig.4—Fig.6, respectively.

Experimental results exhibited that clustering the plans not only achieved reasonably high accuracy but also reduced the number of comparisons. Moreover, it can be observed from Fig. 4—Fig. 6 that the cluster size affects the accuracy. The accuracy was low for smaller cluster size and the reason was that the cluster selection methodology is based on the ranges for each of the premium copay, deductibles, and out-of-pocket limit. Consequently, the likelihood of the selection of clusters with fewer plans is reduced under such criteria, which eventually affects the accuracy for clusters of small sizes. However, with the larger cluster sizes, significant improvements in accuracy were observed. Interestingly the accuracy results for each of the precision, recall, and F-measure were observed significantly high for the clustering schemes based on DBSCAN and Voronoi diagram. On the other hand, Fuzzy C-means and Bayesian clustering turned consistently low in terms of accuracy. Overall the experimental results reveal that the plan clustering not only yields sufficient level of accuracy but also minimizes the number of comparisons that eventually results in reduced response time for real-time user queries.

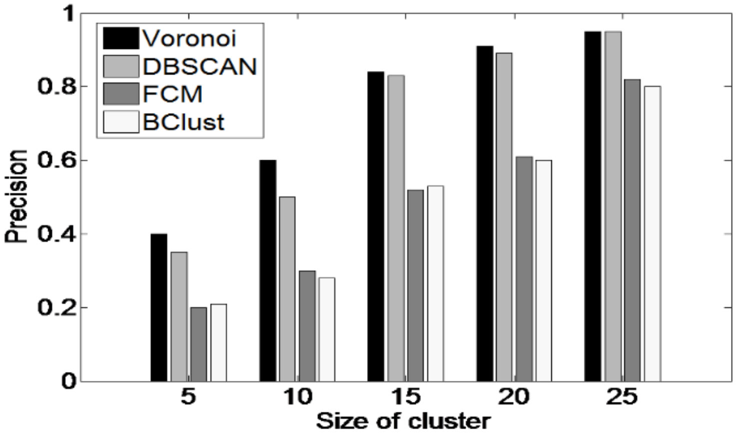

Figure 4. Precision scores for the compared clustering techniques.

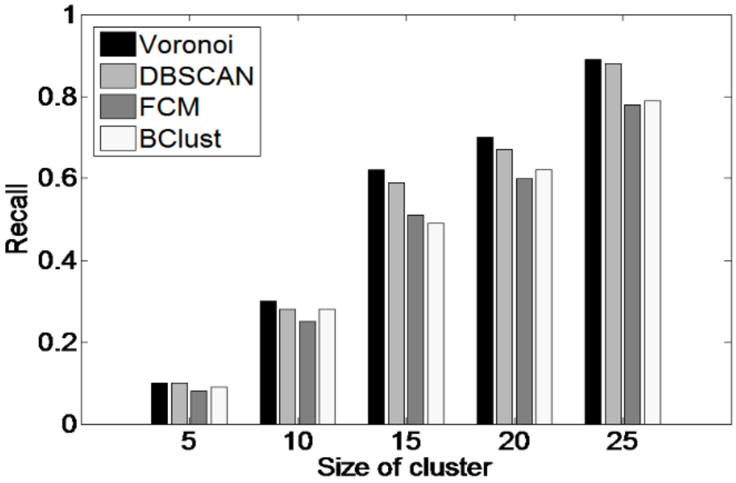

Figure 5. Recall scores for the compared clustering techniques.

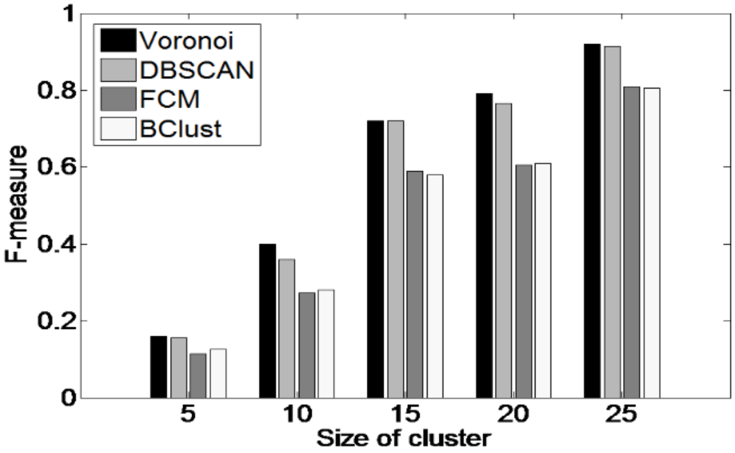

Figure 6. F-measure scores for the compared clustering techniques.

5.3. Scalability Analysis

We also evaluated the scalability of the proposed framework for health insurance plans recommendation. Scalability is a common problem faced by the systems based on centralized computing models [8]. In fact, scalability is the ability of a system to handle huge data volumes effectively. Cloud computing based implementation of the systems ensures the scalable and efficient access to handle massive volumes of health insurance plans data. However, scalability also is critical issue for the parallel algorithms and requires that the performance of a parallel algorithm should not degrade significantly with the increase of workload and the number of processors [26]. In other words, there should be a balance between the number of processors and the size of data to maintain a consistent performance. Increasing the number of processors for a constant workload may result in decrease in efficiency because now the same work has to be performed by more processors and the possible reasons for the decreased efficiency are the overheads in terms of processor startup and communication time [26]. Therefore, for an algorithm to be scalable and efficient, the computational resources should be increased in the same proportion to the workload. To overcome the scalability issue, we leveraged cloud computing services because the cloud computing enables consumers to procure processing and storage resources on demand. A parallel implementation of the algorithm was performed and the performance was evaluated by increasing the workload (number of plans) and the resources (number) of processors.

In Fig. 7, we show the scalability analysis of the proposed framework. To show the effects of increasing the workload on time consumption, the plans were replicated. It can be observed from Fig. 7 that by increasing the number of plans twice using only processor, the execution time significantly increased. However, introducing the additional number of processors resulted in significant decrease in processing time. Experimental results show that by increasing the number of plans twice resulted in an average increase of 72. 57% in total processing time whereas introducing one additional processor resulted in an average decrease of 14.25%. It is important to mention that the execution time depicted in Fig. 7 are the combined execution times for multiple modules including the plans retrieval and transformation, implicit recommendation, explicit recommendation, and clustering. Therefore, the processing times in Fig. 7 are sufficiently reasonable to offer personalized recommendation about health insurance plans from a huge corpus of plans. Fig. 7 also shows that when the number of processors was increased over ten, considerably small decreases in processing time were observed. The reason for this is that with the increase in number of processors, the overhead including time processor startup time and communication time also add in the total processing time. Moreover, increasing the number of processors beyond a certain limit can degrade the performance significantly.

In conclusion, the experimental results reveal that the presented framework maintains the performance to a sufficient level with the increase in workload and the number of processors. Therefore, the framework can be considered feasible to efficiently handle and process large volumes of data.

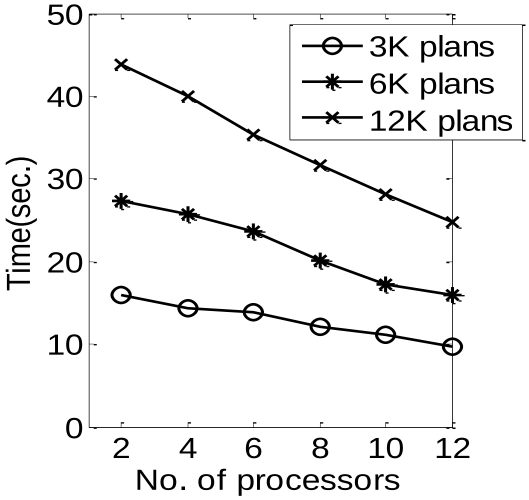

Figure 7. Processing time analysis by increasing the processors and the number of plans.

5.4. Complexity Analysis

Algorithm 1 presents the steps to compute initial ranking of each of the plans. The total time complexity of Algorithm 1 becomes $\boldsymbol{O}\left(\boldsymbol{C}! \times ((\boldsymbol{n} \times \boldsymbol{C}) + \boldsymbol{n}^{\boldsymbol{2}})\right)$, where $\boldsymbol{C}$ is the set of criteria and $\boldsymbol{n}$ is the number of plans. Algorithm 2 computes the similarities between the user's requirements and the plan trees and subsequently computes the ranking score for each of the plans. Since, there are $\boldsymbol{n}$ plan trees with each having $\boldsymbol{c}$ child. Therefore, the total time complexity of the algorithm becomes $\boldsymbol{O}(\boldsymbol{n} \times \boldsymbol{c})$. Likewise, Algorithm 3 takes $\boldsymbol{O}(\boldsymbol{P})$ to calculate the popularity of $\boldsymbol{P}$ plans.

5.5. Related Work

In this part we present the related work to the proposed framework in terms of semantic Web techniques, tree matching, and the multi-criteria decision support.

An ontology based approach to unify the data from distributed repositories in emergency scenario is presented by Li *et al.* [27]. The approach integrates the information retrieved from the data sources into local ontologies that are subsequently represented as RDF schemas. The approach maps XML schemas to ontology to identify the meaningful information. On the other hand, the proposed framework uses a crawler to retrieve dispersed information from the webpages of providers. Subsequently, the information is transformed into ontology to offer a standardized representation. The work presented in [2] utilizes an ontology based approach where each of the insurance providers maintains ontology and updates ontologies and the ontologies are retrieved through Data as a Service (DaaS). The proposed framework instead of developing local ontologies, maintains a generic ontology that is capable of identifying relationships among the coverages offered by different plans. In addition, the proposed framework also offers implicit recommendations to users based on the popularity of the plans.

Another aspect of the proposed work that is related to various other approaches is matching the trees. Aouicha *et al.* [28] presented an exact tree matching approach for XML retrieval. The approach uses the tree structure and calculates the final matching scores. The authors in [29] used a fuzzy tree structure to compute conceptual similarities between two trees. The methodology uses edit distance mapping to identify the tree parts that are similar to each other. The Edit distance is a method to quantify the dissimilarities between two strings. In fact tree edit distance methodologies are suitable when the intent is to find approximate matching between the tree structures. However, in our case we are interested in matching the node labels exactly. Therefore, we used exact matching based approach.

Similarly, the authors in [30] and [31] used sequential tree matching to first decompose the query and subsequently performed a transformation from the paths from root to leaf. On the other hand, our framework uses exact tree matching approach to calculate the structural similarities between the user requirements tree and the plan trees. For each node in the requirement tree, the nodes in the corresponding tree are compared while preserving the ancestry. If the nodes in two trees at the subsequent level match only then the next level of the tree is compared. Just like the approach in [30], our approach sequentially compares the nodes of the two trees and determines similarities regardless of the order of the nodes on the tree. Moreover, the presented framework performs clustering to reduce the number of comparisons between the users' coverage requirements and health insurance plans.

Another important aspect of the proposed work is plan ranking using the MAUT that allows ranking of different health insurance plans based on the importance of four cost based criteria. The authors in [23] used Simple Multi- Attribute Rating Technique (SMART) to aid decision support for an e-commerce recommender system. The weights of the attributes are assigned according to the importance of preferences. Moreover, some other recommendation approaches that are based on the Analytical Hierarchy Process (AHP) for making decisions on the basis of multiple criteria are presented in [32] and [33]. The previous methodology presented in [2] also used the MAUT for ranking decisions based on explicitly stated requirements of the users. On the other hand, the presented framework offers both the implicit and explicit recommendations to the users. Implicit recommendations are offered based on the popularity of different insurance plans. Explicit recommendations are generated based on explicitly stated coverage and cost requirements of users.

Also, in the proposed framework we introduce a methodology to overcome the cold start problem that occurs due to absence of any type of user requirements at the start of system. In addition, we also employ a clustering methodology that clusters the health insurance plans according to their ranking distance from each other. The authors in [34] used the utility theory to determine the matching degree of the products with the satisfaction level of the consumers. The authors transformed the recommendation problem into the problem of constraints satisfaction. The proposed methodology also makes ranking decision based on the prioritized criteria laid down by the user. However, our methodology considers the similarity scores between the users' requirements and the plans in addition to the requirements satisfaction measure to rank the plans. Another important aspect of the proposed framework that is related to the contemporary recommender systems is the ability to handle the long-tail problem. The authors in [35] presented a methodology to overcome the long-tail problem in product recommendation. However, the recommender system only follows the popularity rule and recommends the bestselling items.

Consequently, the new items do not get opportunity to emerge as the popular items. The authors in [36] also claimed to be overcoming the popularity bias in a collaborative filtering recommender system. However, the proposed system utilizes the item ratings that eventually can make the recommendation about items having good ratings. On the other hand, we propose an approach that considers both the popularity of the health insurance plans and the ranking score of each plan to determine the final popularity of a plan. The popularity scores for each of the plans are updated with each user request and therefore, the newly introduced health insurance plans gain opportunity to emerge as the popular plans.

6. CONCLUSIONS

In this study, we presented a cloud based framework that helps users in identifying the health insurance plans based on their predefined criteria in terms of cost and coverage. The framework offers both the implicit and explicit recommendations. The framework effectively resolves the issue arising due to the new system by generating initial set of requirements and subsequently determining the popularity of plans. Explicit recommendations are provided to users based on the specified requirements. A plan ranking methodology is also presented that uses the similarity scores and weights for different cost based criteria specified by the users. We also proposed to cluster the plans using any clustering technique. The clusters are subsequently used to minimize the number of comparisons between the users' requirements and the health insurance plans. Consequently, the users' coverage requirements are matched with the plans included in the identified cluster only and therefore, unnecessary comparisons with other plans are avoided. We evaluated four clustering algorithms and observed that Voronoi diagram based clustering and DBSCAN clustering approaches achieved high accuracy as compared to the Fuzzy C-means and Bayesian hierarchical clustering approaches. We also presented a mechanism that frequently updates the popularity of different plans such that the long-tail issue is overcome.

The scalability issue is addressed using the cloud computing services. The scalability analysis shows that the performance of the framework is sufficiently preserved with the increase in workload and the number of processors. We are optimistic that the framework will be a useful resource for the researchers interested in pursuing research in health insurance recommendation systems.

7. REFERENCES

[1] S. Haeder, and D.L. Weimer, "You can’t make me do it: state implementation of insurance exchanges under the affordable care act," *Public Administration Review*, pp. S34-S47, 2013.

[2] A. Abbas, K. Bilal, L. Zhang, and S. U. Khan, "A Cloud Based Health Insurance Plan Recommendation System: A User Centered Approach," *Future Generation Computer Systems*, Vols. 43-44, pp. 99-109, 2015.

[3] H. Chen, R.H.L. Chiang, V.C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS Q. 36 (4),* pp. 1165-1188, 2012.

[4] Y. Demchenko, C. de Laat and P. Membrey, "Defining Architecture Components of the Big Data Ecosystem," *Collaboration Technologies and Systems (CTS), International Conference on,* pp. 104-112, 2014.

[5] D. Becker, T.D. King, and B. McMullen, "Big data, big data quality problem," *Big Data (Big Data), 2015 IEEE International Conference on,* pp. 2644-2653, 2015.

[6] M. A. Barrett, O. Humblet, R. A. Hiatt, and N. E. Adler, "Big data and disease prevention: From quantified self to quantified communities," *Big Data 1,* vol. 3, pp. 168-175., 2013.

[7] S. Sagiroglu, and D. Sinanc, "Big data: A review," *IEEE International Conference on Collaboration Technol-ogies and Systems (CTS),* pp. 42-47, 2013.

[8] A. Abbas, M. U. S. Khan, M. Ali, S. U. Khan, and L. T. Yang, "A Cloud Based Framework for Identification of Influential Health Experts from Twitter," *15th International Conference on Scalable Computing and Communications (ScalCom),* Aug 2015.

[9] D. Reddy, and P. K. Jana, "A new clustering algorithm based on Voronoi diagram," *International Journal of Data Mining, Modelling and Management 6,* vol. 1, pp. 49-64, 2014.

[10] I. Cordova and T. S. Moh, "DBSCAN on Resilient Distributed Datasets," *High Performance Computing & Simulation (HPCS), 2015 International Conference on,* pp. 531-540, 2015.

[11] H. Izakian, and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem," *Expert Systems with Applications 38,,* vol. 3, pp. .1835-1838., 2011.

[12] K. A. Heller, and Z. Ghahramani , "Bayesian hierarchical clustering," *Proceedings of the 22nd ACM international conference on Machine learning* , pp. 297-304, 2005.

[13] K. Mille, "Big Data Analytics in Biomedical Research," *Biomedical Computation Review,* pp. 14-21, 2012.

[14] A. Abbas and S. U. Khan, "e-Health Cloud: Privacy Concerns and Mitigation Strategies" in Medical Data Privacy Handbook, A. Gkoulalas-Divanis and G. Loukides, Eds., New York, USA: Springer-Verlag, 2016.

[15] A. Abbas and S. U. Khan, "A Review on the State-of-the-Art Privacy Pre-serving Approaches in E-Health Clouds," *IEEE Journal of Biomedical and Health Informatics,* vol. 18, pp. 1431-1441, 2014.

[16] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neu-mann, and S.Hübner, "Ontology-based integration of information-a survey of existing approaches," *IJCAI-01 workshop: ontologies and information sharing,,* pp. 108-117, 2001.

[17] "Blue Cross Blue Shield of North Dakota Qualified Health Plan Drug Formulary".

[18] P. Shvaiko, and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering,* p. 158–176, 2012.

[19] T. B. -Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american 284,* pp. 28-37, 2001.

[20] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International journal of human-computer studies 43,* no. 5, p. 907–928, 1995.

[21] "OWL Web Ontology Language," 2 June 2015.

[22] M.A. Tahraoui, K.P. Sauvagnat, C. Laitang, M. Boughanem, H. Kheddouci, and L. Ning, "A survey on tree matching and XML retrieval," *Computer Science Review,* p. 1–23, 2013.

[23] S. -L. Huang, "Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods," *Electronic Com-merce Research and Applications 10,* no. 4, p. 398–407, 2011.

[24] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, "Challenging the long tail recommendation," *Proceedings of the VLDB Endowment 5,* no. 9, pp. 896-900, 2012.

[25] J. Y. Lee, J.W. Lee, and S.D. Kim, "A quality model for evaluating software-as-a service in cloud computing," *7th ACIS International Conference on Software Engineering Research, Management and Applications,* pp. 261-266, 2009.

[26] M. Ahmed, I. Ahmad, and S. U. Khan, "A Theoretical Analysis of Scalability of the Parallel Genome Assembly Algorithms," *IEEE/EMB/ESEM/BMES International Conference on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS,* pp. 243-237, 2011.

[27] J. Li, Q. Li, C. Liu, S. U. Khan, and N. Ghani, "Community-Based Collabo-rative Information System for Emergency Management," *Computers & Op-erations Research,* vol. 42, pp. 116-124, 2012.

[28] M. B. Aouicha, M. Tmar, M. Boughanem, and M. Abid, "XML information retrieval based on tree matching," *IEEE International Conference on Engineering of Computer Based Systems (ECBS),* pp. 499-500, 2008.

[29] D. Wu, G. Zhang, and J. Lu , "A fuzzy tree matching-based personalized e-learning recommender system,," *IEEE International Conference on Fuzzy Sys-tems (FUZZ-IEEE),* pp. 1898-1904, 2014.

[30] P. Zezula, F. Mandreoli, and R. Martoglia, "Tree signatures and unordered XML pattern matching,”.," *30th Conference on Current Trends in Theory and Practice of Computer Science.,* p. 122–139, 2004.

[31] P. Zezula, G. Amato, F. Debole, and F. Rabitti, "Tree signatures for XML querying and navigation," *Database and XML Technologies,* pp. 149-163, 2003.

[32] M. F. Frimpon, "A Multi-Criteria Decision Analytic Model to Determine the Best Candidate for Executive Leadership," *Journal of Politics and Law 6,* no. 1, pp. 1-1, 2013.

[33] Z. Hua, B. Gong, and X. Xu, "A DS–AHP approach for multi-attribute decision making problem with incomplete information," *Expert Systems with Applications 34,* no. 3, pp. 2221-2227, 2008.

[34] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, "An integrated environment for the development of knowledge-based recommender applications," *International Journal of Electronic Commerce 11,* p. 11–34, 2006.

[35] D. M. Fleder, and K. Hosanagar, "Blockbuster Cultures Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity," *Management Science,* vol. 55 , no. 5, pp. 697-712, 2009.

[36] K. Lee, and K. Lee, " “Escaping your comfort zone: A graph-based recom-mender system for finding novel recommendations among relevant items," *Expert Systems with Applications 42,* no. 10, pp. 4851-4858, 2015.