Cloud based intelligent system for delivering health care as a service

Pankaj Deep Kaur*, Inderveer Chana

Computer Science and Engineering Department, Thapar University, Patiala, India

ABSTRACT

The promising potential of cloud computing and its convergence with technologies such as mobile computing, wireless networks, sensor technologies allows for creation and delivery of newer type of cloud services. In this paper, we advocate the use of cloud computing for the creation and management of cloud based health care services. As a representative case study, we design a Cloud Based Intelligent Health Care Service (CBIHCS) that performs real time monitoring of user health data for diagnosis of chronic illness such as diabetes. Advance body sensor components are utilized to gather user specific health data and store in cloud based storage repositories for subsequent analysis and classification. In addition, infrastructure level mechanisms are proposed to provide dynamic resource elasticity for CBIHCS. Experimental results demonstrate that classification accuracy of 92.59% is achieved with our prototype system and the predicted patterns of CPU usage offer better opportunities for adaptive resource elasticity.

© 2013 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

In recent years, technological innovations have led to the development of new computing infrastructures such as cloud computing that provides infrastructure and software on lease to end users. The "pay for use" pricing model, on-demand computing and ubiquitous network access allow cloud services to be accessible to anyone, anytime, anywhere. The inherent benefits like fast deployment, lower costs, scalability, rapid provisioning, instant elasticity, greater resiliency, rapid re-constitution of services, low-cost disaster recovery and data storage solutions promises the potentials of cloud computing [1,2].

Cloud computing can transform the way healthcare is practiced by empowering professionals to deliver better services in effective management of chronic illness such as Diabetes Mellitus, often mentioned simply as diabetes. Diabetes is a metabolic disorder characterized by high levels of blood glucose in the human body that originates from the defects in the insulin production, insulin usage or both. The insulin hormone secreted by pancreatic beta cells regulates the uptake of the glucose from the blood into most cells of human body [3]. The inability of the human body to produce or properly use the generated insulin hormone results in increased level of blood glucose which eventually leads to many health complications such as damage of heart and stroke; high blood pressure; retinopathy with severe vision loss or blindness and many more [4,5].

There are three prominent classes of diabetes eminently Type 1 diabetes or Insulin-Dependent diabetes mellitus (IDDM), Type 2 diabetes or Non-Insulin-Dependent diabetes mellitus (NIDDM) and Gestational diabetes. Type 1 diabetes results from the body failure to produce insulin and therefore requires the person to inject insulin for survival. Type 2 diabetes arises from the inability of the body to efficiently utilize
the insulin. It is the most prevalent form of diabetes in adults and accounts for about 90–95 percent of all diagnosed cases of diabetes. The third class of diabetes is Gestational diabetes, a form of glucose intolerance diagnosed during pregnancy [6].

In this paper, we advocate the use of cloud computing for the creation and management of cloud based health care services. We design a Cloud Based Intelligent Health Care Service (CBIHCS) that performs real time monitoring of user health data for effective management of chronic illness. Our system gathers gigantic data sets of users from different locations using advanced components such as glucose meters and other sensory devices. The accumulated data is stored in a cloud based storage repository for efficient retrieval, updates and quick transfers as and when required. This data is subsequently analyzed by our prototype service to identify patients with diabetes by closely monitoring their vital statistics. Furthermore, to accommodate variably high service usage demand, our resource infrastructure must automatically expand or contract with the changing load. To ensure this, we augment our system with dynamic infrastructure elasticity that scales the underlying resource pool in accordance with the predicted resource usage patterns. Thus, a stable level of application performance on service provider's infrastructure is ensured regardless of the service usage demand.

1.1. Motivation

The motivation of this work stems from the fact that although significant efforts have been done to utilize traditional Information and Communication Technology (ICT) infrastructures to provide e-health care service solutions yet these solutions become insufficient to retain patient data for long periods of time due to limited set of constrained resources [7]. Furthermore, ICTs infrastructures are usually confined to a limited user base within a geographic region and thus result in increased cost of processing, storage and energy requirements [8]. Clouds, in contrast, relieves consumers from the need to own their physical computing infrastructures and gives them the liberty to use the best technology on a rental basis. It allows for wide accessibility and provides access to huge amount of computational and storage resources as per the requirements of users limited by the available infrastructure at IaaS provider end [9].

1.2. Technical challenges

Accurate, well-timed and precise information is a prerequisite of an efficient health care service, especially if the service is hosted on large scale distributed infrastructure such as Clouds and Grids. The key challenges in designing such a cloud based health care system are outlined as follows:

- Data heterogeneity. Patients may use variety of health monitoring tools such as glucose meters, weight scales, blood pressure monitors or other wireless devices available in the market to monitor their health data. The collected data often have missing values and may arrive in different formats making it difficult to process in a uniform way.

- Data size. Monitoring, processing and transfer of user health data can go up to 24 hours a day, 7 days a week and 365 days a year depending on the individual configuration settings. These huge gigantic health data sets requires to be stored and retained for longer time periods for subsequent analysis with increased efficiency.

- Data costs. Cost reduction is an important challenge for health care service providers to increase its user base. However, the cost should not trade off with the performance of our health care web service. Performance characteristics of applications must be maintained while lowering down the costs associated with the underlying infrastructure. Lack of desired performance, however, steadily decreases the provider's profit with consumers shifting to other providers. Henceforth, efficient mechanisms are required that predict the service usage demand of users and scales the underlying resources in response to the changing load of user's access requests.

- Data security. Security is an important concern for cloud environments where the data resides within the vicinity of service providers. The problem becomes even more challenging when the infrastructure is used to host the data of personal medical nature.

1.3. Our approach

The primary contribution of this paper lies in a collection of techniques to identify users with chronic illness such as diabetes and proposing adaptive mechanisms for supporting infrastructure elasticity. At the SaaS layer, an automated analysis of user's health data is performed that assists professionals to identify patients with diabetes by closely monitoring their vital statistics. Specifically, data preprocessing is first accomplished that addresses the issue of data heterogeneity by transforming the variable data collected from arbitrary components into a uniform format for data analysis. Secondly, to deal with huge data sizes, attribute selection is employed that transforms the multi-dimensional user data into fewer dimensions. The feature selection technique has the potential to identify the most useful information from the data and reduce the dimensionality in such a way that the most significant aspects of the data are represented by the selected features [10]. In particular, we use Principal Component Analysis for feature selection. Two classification techniques, emminently, K-Nearest Neighbor and Naïve Bayes Classifier are introduced that classifies individual users as “Diabetic” and “Non-Diabetic”. The classification outcome thus obtained is evaluated for classification accuracy along with sensitivity and specificity measures. Finally, the results attained are sent to individual users, doctors and paramedics for final validation and clinical diagnosis.

At the infrastructure level, we implement resource elasticity mechanisms that scale the underlying resource infrastructure for optimal performance of CBIHCS and thus reducing the data costs. Prediction based techniques specifically Exponential smoothing (single, double, triple) and Regression Analysis are utilized that derive the resource usage patterns for CBIHCS for varying workload and make prediction of future resource needs. As a result, dynamic resource elasticity is provided by creating and destroying virtual machines on
2. Related work

Tremendous efforts are undertaken by the researchers from the academia and industry to provide computer assisted health care solutions. A recent fact sheet released by World Health Organization (WHO) revealed that 346 million people worldwide are suffering from diabetes. It is projected that the number of diabetes deaths will double between 2005 and 2030 [11]. United States alone has 8.3% of the population diagnosed with diabetes as stated by American Diabetes Association. In 2007, an estimated diabetes costs in United States reached $174 billion [6,12]. The results clearly indicate the urgency of efficient strategies for identification and prevention of diabetes in a cost effective manner.

2.1. Existing approaches for E-health care

ICT based health care services have the potential to provide a high quality, guideline based care with decreased medication errors [13]. Different solutions have been proposed for providing e-health care [14–16]. The solutions lack a comprehensive integration with its intended environment and needs adaptation to specific infrastructure.

The success of Web 2.0 and increasing trend toward social networking has resulted in number of health applications such as PatientsLikeMe [17], SugarStats [18], CureTogether [19], TUDiabetes [20] that allow patients to maintain their own health and seek advice. These applications are basically focused on sharing information and substantially lack medical diagnosis or disease treatment [21].

Furthermore, sedentary life style, aging population and rising medical expenditures demand an economically viable personalized health care service solution. Despite the increase in development and implementation of computer based patient health care systems, medical errors arising due to faulty process of information entry and retrieval impede its successful adoption [22]. Numerous articles and resources report successful integration of Wireless Sensor Networks (WSN) and Body Area Networks (BAN) to automate the process of individual health care by allowing continuous monitoring and analysis of health data [23–26]. These networks, however, face significant challenges in terms of inadequate storage and processing capabilities, constrained network and limited battery life.

Health care applications being real time applications require robust IT infrastructure to generate high level of responsiveness. To address such concerns, researchers have investigated the use of Grid Computing for the resource infrastructure needs. Number of biomedical research projects were initiated that exploited grid resources to perform number of parallel computations [27–30]. Grid based e-health initiatives are mainly targeted for the research community needs that require massive computational power to analyze huge data for research purposes such as development of new drugs. Although, the infrastructural requirements for the researchers are realized from the Grid resources yet the inherent limitations of Grid technology impede its successful adoption by individual patients. Cloud computing, sharing the same underlying goals of grid computing, provides on-demand computing access to resources with better usability and accessibility options for the targeted user base [31].

Although Kuo et al. [32] suggested numerous opportunities of cloud computing to improve health care services yet very few works from academia are reported on Cloud based health care. One such work is described by Pandey et al. [33] that utilized Aneka [34] framework to create an autonomic cloud environment for hosting ECG data analysis services. They proposed the use of simple heuristics to provide elastic infrastructure for ECG processing service and considered response time as the only Quality of Service (QoS) factor. Our work shares similar objectives of [33] utilizing data mining techniques for delivering intelligent service behavior and stronger focus on QoS issues including application performance, resource utilization, security and costs. The experimental results of our prototype system are conducted on a public Cloud specifically Amazon EC2 and the results clearly demonstrates the effectiveness of our approach to perform resource elasticity decisions while maintaining application performance within constrained budgets.

3. Architectural overview

Our objective is to design a Cloud Based Intelligent Health Care Service (CBIHCS) that performs real time monitoring of user health data for identification of chronic illness such as diabetes. Our system is generic enough to accommodate diagnosis and detection of multiple diseases by analyzing the user health data stored in cloud repositories. However, in this paper we focus on only one specific use case, namely, identifying users as “Diabetic” or “Non-Diabetic”. The architectural overview of our proposed system is depicted in Fig. 1. As shown, our system is composed of two subsystems: (1) User Subsystem; (2) Cloud Subsystem.

3.1. User subsystem

The user subsystem provides patients with personalized and intelligent monitoring of their health data in real time eliminating the need for patients to visit hospitals to get their vital statistics read. It integrates number of wireless health...
monitoring devices such as Blood Pressure monitors, Peak Flow meters, Glucose monitor, Body weight scale, ECG monitors etc. with a Cloud Based Intelligent Health Care Service (CBIHCS). These devices are outfitted on patient’s body that requires to be remotely monitored. The readings received from the devices are automatically forwarded to a preconfigured mobile device (Mobile phones, FDA’s, Laptops etc.) via Bluetooth through a self-formed network. It thus requires no IT expertise of the patient and allows for secure and accurate transmission of patient’s health data to our cloud based web service without the user intervention.

3.2. Cloud subsystem

The Cloud Subsystem is composed of our proposed health care web service hosted on a cloud computing based software stack. The web service consists of a user interface that allows patients to submit their personal details such as Patient name, age, height, gender, sleep time, family history of disease, habitual physical inactivity, smoking and drinking habits etc. These details are stored in a cloud based storage repository indexed by patient id which is automatically generated by our web service. Furthermore, the monitoring and analysis of data is accomplished based on the individual preferences such as every hour, after every meal or some fixed time intervals during the day. The analysis process involves several computations on stored data that primarily consist of data preprocessing, attribute selection and classification as shown in Fig. 1. The user data in the storage repository is then classified with an indication of potential health problem or disease. This information is then relayed wirelessly to doctors, paramedics and patient’s mobile devices for final validation and clinical diagnosis. In addition, since our web service may be accessed by multiple users around the globe so we augment our cloud subsystem with dynamic capabilities that derive the resource usage patterns to predict the future resource needs. Henceforth, dynamic resource elasticity is performed at the infrastructure level.

4. Detailed description

This section describes in detail the techniques utilized in the process of analyzing user health data for identification of diabetes and the corresponding mechanisms instrumental in the cloud subsystem for implementing infrastructure elasticity.

4.1. Cloud based health care web service – SaaS layer

Our Cloud based health care web service is a multi-tier SaaS application that allows real time monitoring of user health data for identification of chronic illness such as diabetes. The process flow diagram of CBIHCS is represented in Fig. 2.

4.1.1. Functional aspects of CBIHCS

The multiple functionalities provided by our CBIHCS are identified as tasks classes for the purpose of application behavior analysis. The web service allows users to create their profile once they subscribe to CBIHCS. The create profile task presents a user interface that allows patients to submit their personal details.

The task named store data is responsible for storing the patient details in a cloud based storage repository indexed by patient id (which is automatically generated by our web service). The patient may choose to observe and analyze his data every hour, after every meal or some fixed time intervals during the day. The analysis of the patient data is accomplished by analyze task.

The analysis process involves several computations on stored data that primarily consist of data preprocessing, attribute selection and classification. Once the analysis is complete the information can then be relayed wirelessly to doctors, paramedics and patient’s mobile devices for final validation and clinical diagnosis. The task named transmit data is accountable for transmitting data so that validation can be performed. After the data is approved, the doctors may recommend medication and save in user accessible cloud storage. The task Validate and store provides the said functionality.
4.1.2. Security aspects of CBIHCS

To address the security challenges in a cloud hosted health care application, we implement security mechanisms at multiple levels and provide role based access control to ensure the protection of critical medical data of patients. The two types of user roles defined by CBIHCS are (1) Owner; (2) Trusted Partner. The patient whose medical data resides in our cloud hosted web application is designated as the Owner ‘O’ of data. Additionally, at times, the owner may wish to share his personal medical data with a group of people for consultation or other purposes. We call such people as Trusted Partners (TP) of the owner. We label MngTP as the task performed by data owner to create, modify and delete trusted partners.

The three types of trusted partners defined by CBIHCS are (1) Reader, (2) Editor, (3) Anonymous. A Trusted Partner designated as Reader ‘TPR’ is allowed to access a read only copy of the patient record such as family members of patient. Trusted partner with an Editor role ‘TPE’ is allowed to read the owner’s data and also make limited modifications. For example, a physician of the owner is designated as an Editor who may view the patient’s record and prescribe new set of medications by writing to the patient’s database. Anonymous Trusted Partners ‘TPA’ includes anyone who is allowed to read patient data in a pseudonymized manner. Examples include a government body or a research firm that may need access to enormous records of patient’s medical data for development of new drug or medicine. It is mandatory that the firm is listed as a Trusted Partner for the patient and our web service ensures that the data supplied to the anonymous Trusted Partner is made de-identifiable by excluding Patient Id, Name and Address.

To implement security mechanisms, we propose to integrate the use of symmetric cryptosystems for authentication and role based access control (RBAC) mechanisms for authorization. Users of CBIHCS are identified by a unique user name and password. Rather than storing the user password in plain text on cloud based storage, we encrypt the password with a Private Key ‘Ppk’ issued by a Trusted Third Party (TTP). A TTP is an entity that assures the security support for data and communications exchanged between the relying customer parties. In addition, it ensures that only legitimate users who are registered with CBIHCS can access its functionalities while preventing fake users (who may be infrastructure owners or users with administrator privileges) to access the data of other users.

Once the users are authenticated based on the credentials, the authorization is implemented based on the user roles. The authorization roles associated with user id allows for execution of tasks and workflows. Every data owner is provided with a functionality to create number of Trusted Partners and assign different authorizations to them labeled as (‘TPR’, ‘TPE’, ‘TPA’). Further, a single key ‘Xk’ issued by TTP is shared between the Data Owner ‘O’ and all the TPs under him. This key ‘Xk’ is used to encrypt the patient data before storing it on cloud. Therefore, any user who has access to ‘Xk’ can decrypt
the patient data and access it based on the assigned authorization level (‘O’, ‘TPe’, ‘TPe’, ‘TPx’). The use of multiple level symmetric cryptosystems in this work successfully addresses the security concerns of CBHCS and we leave to explore the integration of asymmetric cryptography techniques as part of the future work.

4.1.3. Diabetes identification – a detailed methodology
The Cloud Based Health care Service Delivery System allows users to automatically upload data from different meters using modern health care devices and classifies them as Diabetic or Non-Diabetic. The subtasks comprising this identification are (A) Data Pre-processing, (B) Attribute Extraction, (C) Data Classification.

4.1.3.1. Data preprocessing. Modern health care devices come equipped with variety of tools that allow patients to automatically upload data from different meters. However, such data might contain some noise components or missing samples necessitating the execution of preprocessing steps. Data preprocessing comprises of thorough examination of raw data followed by subsequent data integrations, transformations and reduction for formal analysis. Distorted or missing readings often become misleading in the formal analysis. Henceforth, sufficient number of quality samples must be collected and analyzed to allow for critical evaluation.

With a cloud based web service, the monitoring interval of health data is scheduled as per the user preference settings. Nevertheless, it is still possible that the user might require to upload his data for dynamic analysis in response to certain unforeseen complications arising due to hypoglycaemia or hyperglycaemia [35,36]. For example, excessive exercising or prolonged delays in meal reduces the blood sugar in human body and an intake of insulin in such times may worsen the patient situation causing nervousness, severe headaches, loss of consciousness and even coma in acute cases. Also, some measurement problems may result in missing values. Hence, to deal with such situations we utilize data interpolation techniques for converting infrequent or non-uniformly monitored data to a uniform sampled data. Based on the assumption that blood glucose profiles at the same time on different days are usually similar to each other, we interpolate missing values using the Shepard interpolation [37]. Missing values are computed as the weighted sum of blood glucose readings made at the same time on adjacent days. Weights of supportive readings are inversely proportional to the time elapsed between the day they were recorded and the day where missing data are to be estimated.

To account for variable data formats we resort to construct a uniform attribute array for formal analysis [38]. Let \( n \) be the number of attributes submitted by \( p \) users. To ensure the reliability of collected data, \( s \) snapshots are sampled per user. This eventually results in \( p \cdot s \) number of two dimensional arrays (called data matrices \( D^i \)) corresponding to each user \( i \), where \( (i = 1, 2, 3, ..., p) \). Thus, each data matrix \( D^i \) contains data values of attributes collected from \( p \) users and every element \( d^i_{j,k} \) represents the value of attribute \( j \) for snapshot \( k \) belonging to user \( i \), where \( 1 \leq i \leq p \), \( 1 \leq j \leq n \) and \( 1 \leq k \leq s \).

\[
\begin{array}{cccccccc}
  d^1_{1,1} & d^1_{1,2} & \cdots & d^1_{1,n} & d^2_{1,1} & d^2_{1,2} & \cdots & d^2_{1,n} \\
  d^1_{2,1} & d^1_{2,2} & \cdots & d^1_{2,n} & d^2_{2,1} & d^2_{2,2} & \cdots & d^2_{2,n} \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
  d^n_{1,1} & d^n_{1,2} & \cdots & d^n_{1,n} & d^n_{2,1} & d^n_{2,2} & \cdots & d^n_{2,n} \\
\end{array}
\]

To facilitate data analysis, each matrix \( D^i \) is reorganized into a column matrix \( d^i \) with dimensions \((n \times s)\) such that

\[
d^i = [d^i_{1,1} d^i_{1,2} \cdots d^i_{1,n}]^T, \quad d^1 = [d^1_{1,1} d^1_{1,2} \cdots d^1_{1,n}]^T, \quad d^2 = [d^2_{1,1} d^2_{1,2} \cdots d^2_{1,n}]^T, \ldots, \quad d^p = [d^p_{1,1} d^p_{1,2} \cdots d^p_{1,n}]^T
\]

where \( 1 \leq i \leq p \).

Finally, a single large matrix is constructed using (1) that contains the data values of all the attributes of users as:

\[
D_{(n \times s) \times p} = [d^1 \ d^2 \ \cdots \ d^p]
\]

(2)

The resulting conversion of a multi-way matrix to a single large matrix thus allows easy extraction of appropriate attributes for disease identification.

4.1.3.2. Attribute extraction. Real life data often records more attribute variables than are strictly necessary for the classification task. Our prototype cloud health care service is generic enough to accommodate storage of numerous attributes information specific to user in cloud repositories. It thus allows for diagnosis and detection of several diseases. However, in our specific use case, only a small number of attributes are actually utilized to identify users as “Diabetic” or “Non-Diabetic”. Considering all these attributes as relevant attributes we employ attribute extraction to transform the attribute space in a low-dimensional subspace. Specifically, we utilize Principal Component Analysis (PCA) to identify the most discriminative attribute variables so that minimum correlation exists between them. The stored data is thus transformed into new space such that the resultant data becomes easier to be separated into different classes.

PCA is a useful statistical technique for analyzing the data for determining key variables in a high dimensional data set by reducing the number of dimensions without any loss of information [39]. Mathematically, PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called
the first principal component), the second greatest variance on the second coordinate, and so on. PCA is thus capable of transforming a number of possibly correlated variables into a smaller number of uncorrelated variables called as principal components.

In our case study, multiple attributes containing user specific health information are gathered using electronic equipment or obtained using manual submission by users using our health care web service. Each recorded attribute value is expressed in different units of measurement as indicated in Table 1. For example, the HDL cholesterol of user is usually a large number less than 200 and is expressed in mg/dl, while the physical activity status is expressed as a discrete value in the scales 1–5. The value ‘1’ for the physical activity indicates no exercise at all while the value ‘5’ depicts heavy exercise level of the user. Henceforth, it becomes necessary to transform the data matrix into a uniform format. To do so, we first apply normalization that transforms the data values for the attributes present in the data matrix D to obtain newer matrix $D'$. The new values in $D'$ are obtained in a uniform scale ranging from 0.0 to 1.0.

Before applying PCA, we adjust the normalized data matrix $D'$ to $D''$ such that its columns have zero mean. Zero mean ensures Mean Square Error (MSE) of data approximation in finding a principal component basis remains minimum. Each column of the resultant matrix $D''$ thus corresponds to the distinct users of our web service while the row values contain the values of the health attributes of users.

PCA is now applied on the adjusted zero mean matrix $D''$. A Covariance matrix, $C$, is computed from the transformed data matrix $D''$ using Eq. (3) that treats each column as the data point in the region $(n \times s)$ as:

$$C = \frac{1}{p} - I \cdot D'' \cdot D''^T$$

(3)

Subsequently, the eigen values $\lambda_1, \lambda_2, \ldots, \lambda_p$ and the corresponding Eigen vectors $E_1, E_2, \ldots, E_p$ are computed from the covariance matrix $C$. The computed eigen values are sorted in the decreasing order as $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_{n \times s}$. The ith principal component corresponding to eigen value $\lambda_i$ is computed using Eq. (4) by projecting each data point of $d_i \epsilon \text{region (n x s)}$ into a data point $y_i \epsilon \text{region u}$

$$y_i = E_i^T \cdot d_i \quad \text{where} \quad i = 1, 2, 3, \ldots, p.$$  

(4)

Since most of the variation in the data is concentrated in the first few principal components, so we need to calculate only first few eigen vectors. The number of eigen vectors that satisfy the Eq. (5) are chosen:

$$k = \frac{\lambda_1 + \lambda_2 + \lambda_3 + \ldots + \lambda_u}{\lambda_1 + \lambda_2 + \lambda_3 + \ldots + \lambda_{n \times s}} \times 100\%$$

(5)

where $u \leq n \times s$ and $k \geq \text{TH} \quad \text{(a certain predefined threshold value).}$

4.1.3.3. User health status classification. The final step is to classify the health status of users based on the extracted data of the principal components. In our work we investigate two well-known classification techniques (i) K-Nearest Neighbor (K-NN) and (ii) Naïve Bayes Classifier. These classifiers are utilized by CBHCs to determine the categorical class labels of users. In the following subsections, a brief description of the techniques utilized in our experiments is presented.

(i) K-Nearest Neighbor. K-Nearest Neighbor (KNN) is the most commonly used instance based supervised machine learning technique. It utilizes training data to classify unknown instances [41, 42]. The instances included in the training data set include all possible cases with similar properties and known class labels.

**Algorithm 1. K-NN outline**

Learning:

- Construct the set of training instances ‘T’.

Classification:

- For each unknown instance $y_i$
  
  1. Find $y_{a_1}, y_{a_2}, \ldots, y_{a_k}$, the k most nearest instances from T nearest to $y_i$, using the distance metric where $y_{a_1}, y_{a_2}, \ldots, y_{a_k}$ are the data points in the region u.
  
  2. Set class label = most frequent class label of the k nearest instances.
  
  3. Return class label.

To classify an unknown instance, the distance (using some distance measure) is calculated from that instance to every other training instance. The description of both the classifiers K Nearest Neighbor and the k smallest distances are identified, and the most represented class in these k classes is considered as the output class label. To break the ties the value of k is usually chosen as an odd number. The pseudo-code of the K-NN technique is described in Algorithm 1. K-NN does not have any training phase and all the training data is utilized during the testing phase. It is therefore considered as a lazy learning algorithm as it defers computation until classification is performed.

(ii) Naïve Bayes classifier. Naïve Bayes classifier [43, 44] is considered to be one of the most powerful probabilistic classification technique that classifies high dimensional input data. The method strongly assumes independence of attributes to each other and henceforth is named naïve. It utilizes Bayes theorem to calculate the probability that an unknown instance Y is classified as class ‘C’ among a set of possible outcomes $C \in \{ C_1, C_2, C_3, \ldots, C_{n}\}$. This probability is known as posterior probability and is expressed using the Bayes rule as in Eq. (6):

$$P(C|Y) = \frac{P(Y|C) \cdot P(C)}{P(Y)}$$

(6)

Since naïve bayes assumes the conditional probabilities of the independent variables are statistically independent the likelihood $P(Y|C)$ can be decomposed to a product of terms as in Eq. (7):

$$P(Y|C) = P(y_1|C) \times P(y_2|C) \times \cdots \times P(y_n|C)$$

(7)

where $y_1, y_2, \ldots, y_n$ are the data points in the region u corresponding to n features with s snapshots of each. Therefore,
the posterior probability of Eq. (6) can be rewritten using Eq. (7) as:

\[ P(C|Y) = P(C) \cdot \prod_{i=1}^{m} P(y_i|C) \]  

An unknown instance Y can thus be labeled using Eq. (8) as belonging to class \( C_k \) that achieves the highest posterior probability. The pseudo-code of the Naïve Bayes (NB) classifier is described in Algorithm 2.

Algorithm 2. NB outline

Learning:
Given the set of training instances '\( T \)',
1. For each target value of class \( c_i \) (\( c_1, c_2, \ldots, c_l \)), Estimate \( P(C = c_i) \) with examples in \( T \).
2. For every data value \( y_{jk} \) (\( j = 1, 2, \ldots, n; k = 1, 2, \ldots, s \)) of the data point \( Y \), Estimate \( P(Y = y_{jk}|C = c_i) \) with examples in \( T \).

Classification:
For each unknown instance \( y_i \)
1. Traverse probability tables to assign class \( c^* \) to \( Y_i \) if
   \[
   \frac{P(y_{i1}|c^*) \ldots P(y_{is}|c^*)} {P(y_{i1}|c_1) \ldots P(y_{is}|c_l)} P(c^*) > \frac{P(y_{i1}|c_1) \ldots P(y_{is}|c_l)} P(c)
   \]
   where \( c \neq c^* \), \( c = \{c_1, c_2, \ldots, c_l\} \)
2. Return \( c^* \) as the class label

4.1.4. Infrastructure elasticity
One of the most innovative part of our prototype system is to offer resource elasticity at the infrastructure level. In particular, we utilize state-of-the-art statistical prediction techniques [45,46] that forecast the future resource usage needs that help to determine the amount of underlying resources required for the application. As CPU is considered as the most bottleneck resource, so we primarily take into account three types of usage patterns for CPU utilization by our prototype application as indicated in Table 2.

We divide the timeline into \( n \) number of fixed time intervals. Let ‘\( T \)’ be the current time at which the CPU usage prediction is performed; ‘\( V_T \)’ is the data associated with time interval ‘\( T \)’; \( d \) represents the number of intervals in advance after which the forecast has to be calculated; ‘\( F_{T+d} \)’ indicates the forecasted data value at time ‘\( T + d \)’. We assume that the prediction is performed one period in advance i.e. \( d = 1 \).

Case1 (Steady Pattern). This case considers the steady rate of CPU utilization over a specific time frame. We consider three approaches for forecasting the CPU usage value.

(a) Preliminary technique: The preliminary technique assumes the value at the next time interval is equal to the value at the current time interval as in Eq. (9):

\[ F_{T+1} = V_T \]  

(b) Moving average: The moving average technique predicts the future values by taking the average over the last \( k \) intervals of the available data values as expressed in Eq. (10):

\[ F_{T+1} = \frac{\sum_{k=0}^{m-1} (V_{T-k})}{m} \]  

(c) Exponential smoothing: This technique predicts the future value by taking into account all past values rather than just \( k \) past values as in Moving Average. However, only the most recent forecasted value needs to be stored to calculate the next value. The expression of exponential smoothing is given in Eq. (11):

\[ F_{T+1} = a V_T + (1-a) F_T \]
where $V_T$ is the most recent observation, $F_T$ is the last forecast and $\alpha$ is the smoothing factor, $0 < \alpha < 1$, usually 0.1 or 0.2.

**Case 2 (Linear Pattern).** This case assumes that the rate of CPU utilization increases linearly over a specific time range. Considering $S_T$ as the current estimate of the intercept and $G_T$ as the current estimate of the slope; forecast for time $d$ into the future $F_T(d)$ can generally be expressed as in Eq. (12):

$$F_T(d) = S_T + G_T \cdot d \quad \text{where} \quad d = 1, 2$$

We consider four approaches for computing $S_T$ and $G_T$ and henceforth calculating the forecasted value of the user access request are described below:

(a) **Preliminary technique:** Using this technique, the values of $a_T$ and $b_T$ are computed as in Eq. (13):

$$S_T = V_T; \quad G_T = V_T - V_{T-1}$$

(b) **Linear regression:** This technique computes $G_T$ and $S_T$ using Eqs. (14) and (15) respectively assuming $m$ last data values as described below:

$$G_T = \frac{-m}{2m-1} \sum_{k=0}^{m-1} V_{T-k} + \sum_{k=0}^{m-1} k \cdot V_{T-k}$$

$$S_T = \frac{\sum_{k=0}^{m-1} V_{T-k} + G_T \cdot \frac{m(m-1)}{2}}{m}$$

(c) **Moving average:** The moving average computes $\gamma_T$, $\delta_T$ using Eq. (16) to obtain the values of $S_T$ and $G_T$ as in Eq. (17):

$$\gamma_T = \frac{\sum_{k=0}^{m-1} V_{T-k}}{m}; \quad \delta_T = \frac{\sum_{k=0}^{m-1} \gamma_T \cdot k}{m}$$

$$S_T = 2 \cdot \gamma_T - \delta_T; \quad G_T = \frac{2}{m-1} (\gamma_T - \delta_T)$$

(d) **Double exponential smoothing technique:** The technique is so called as it requires separate smoothing constants for slope and intercept. The advantage is that once we begin building a forecast model, we can quickly revise the slope and signal constituents with the separate smoothing coefficients. The values of $S_T$ and $G_T$ are computed by setting $S_1 = V_1$ and $G_1 = 0$ as in Eq. (18):

$$S_T = \alpha \cdot V_T + (1 - \alpha) (S_{T-1} + G_{T-1}); \quad G_T = \beta (S_T - S_{T-1})$$

$$+ (1 - \beta) G_{T-1}$$

**Case 3 (Repetitive Pattern).** This case assumes that a pattern of CPU utilization repeats itself periodically. Let ‘R’ be the length of period for which the pattern of CPU utilization repeats itself. For the repetitive patterns, we consider the preliminary technique as well as the Triple Exponential method as described below:

(a) **Preliminary technique:** This technique assumes that the forecast for time $d$ into the future $F_T(d)$ with ‘R’ as the length of period can generally be expressed as in Eq. (19):

$$F_T(d + k \cdot R) = V_{T+d-R} \quad \text{where} \quad d = 1, 2, \ldots, R; \quad k = 1, 2, \ldots$$

(b) **Triple exponential smoothing technique:** The method is so called as it requires three smoothing constants; first for the signal, second for the trend and third for seasonal factors. It is commonly known as Holt’s-Winter’s method after the name of its inventors and is expressed as in Eq. (20):

$$F_T(d + k \cdot R) = S_T + G_T \times d + C_T \cdot d \cdot R$$

where $d = 1, 2, \ldots, R; \ S_T$ is the ‘intercept’ of the demand; $G_T$ is the trend slope component of the demand; $C_T \cdot d \cdot R$ is seasonal component for time of interest.

The values for $S_T$, $G_T$ and $C_T$ can be computed by setting the initial values as in Eq. (21) and substituting in Eq. (22):

$$S_R = V_R; \quad G_R = \frac{V_{R-1} - V_R}{R}; \quad C_d = V_d - (V_1 + G_R (d - 1))$$

$$S_T = a (V_T - C_T \cdot k) + (1 - a) (S_{T-1} + G_{T-1}); \quad G_T = \beta (S_T - S_{T-1}) + (1 - \beta) G_{T-1}$$

$$C_T = \gamma (V_T - S_T) + (1 - \gamma) C_{T-1}$$

In a dynamic cloud environment, the access requests of users vary dynamically. As such, it becomes extremely difficult to maintain the performance of the application in response to the changing user request load. So we use a simple heuristic that observes the past pattern of CPU utilization and selects the best available technique for the specific pattern to calculate the future value. Based on the calculated future value, computing resources are provisioned in advance while maintaining the performance characteristics of the application. The steps that implement our solution are summarized in the flowchart as given in Fig. 3.
5. Experiments

Our Cloud Based Intelligent Health Care Service (CBIHCS) is a three-tier web application that makes use of Apache at the web server tier, Tomcat at the application tier with MySQL as the database server. WEKA [47] toolkit is used for the data mining operations. In our implementation, we first wrap our application into a set of service components according to the Web Service Resource Framework (WSRF) [48] standards. This adaption allows our application to be easily deployed on a third party cloud computing environment such as Amazon EC2. Amazon EC2 is an IaaS cloud provider that offers numerous types of machine instances. We use Amazon EC2 default instance type ‘m1.small’ with Amazon Machine Image (AMI) running CentOS 5.3 with a Linux 2.6.18 Xen kernel. We specify 3 and 75 as the minimum and the maximum number of Amazon Machine Image (AMI) instances respectively. Also, us-east-1a is selected as the availability zone for launching instances in our experiments.

5.1. Data acquisition

In our diabetic case study, we classified user health status based on the training data collected from 65 subjects including 35 males and 30 females. The subjects range in the age of 18–85 years and the mean age is 52 years with a standard deviation of 7.5. User’s personal data is recorded one time by our web service interface and subsequent health data values are recorded continuously every hour for a period of 7 days using electronic health devices. After the seventh day, the recorded values are saved to a file and utilized in our experiments. Out of the total 65 subjects, 40 subjects are “Diabetic” and 25 are “Non-Diabetic”. Data of 23 diabetic subjects is selected for
training and remaining 17 subjects are used as test set. For Non-diabetic use case, data of 14 subjects is utilized for training while the remaining 11 subjects are chosen for testing. The true diagnosis is verified under the supervision of expert endocrinologist who has special training and experience in treating people with diabetes. Table 3 details the statistical analysis of the electronically recorded attributes along with the mean and standard deviations.

5.2. Preprocessing

After dealing with noisy and missing data values, we first normalize the attribute data values to ensure that all the values lie in the interval [0.0–1.0]. Thereafter, we use the PCA algorithm to reduce the dimension of the user health attributes. Specifically, the number of attributes is reduced from 17 to 7 using PCA. Further, in application of our system, we use Euclidean distance as the distance metric between the two data points \( y_a \) in region \( s \) and \( y_b \) in region \( s \) as:

\[
D(y_a, y_b) = \sqrt{\sum_{i=1}^{n}(y_{a,i} - y_{b,i})^2}
\]

5.3. Evaluation criteria

The objectives of CBHCS are two-fold. Firstly, it needs to correctly identify the patients as diabetic (Type-1 or Type-2) or non-diabetic and secondly, it intends to identify the access patterns of user requests so as to forecast the future resource needs. Henceforth, different measures are used to calculate the performance of classifier and the predictor.

5.3.1. Classifier performance

We use classification accuracy along with sensitivity and specificity measures as metrics for evaluating the performance of our classifier [49].

(a) Classification accuracy: The classification accuracy of a classifier is measured as the ratio of the number of subjects correctly classified to the total number of subjects. It is evaluated by the formula:

\[
CA = \frac{t_c}{n}.100
\]

where \( t_c \) is the number of correctly classified subjects and \( n \) is the total number of subjects cases.

(b) Sensitivity and specificity: These two measures are widely used to evaluate the performance of a two-class classifier.

Sensitivity specifies the proportion of actual diabetic users, which are correctly classified whereas Specificity is the proportion of non-diabetic users which are correctly identified. The relation among the terms is depicted in Table 4. The measures are calculated using the formulas as given below:

Sensitivity = \( \frac{TP}{TP + FN} \)

Specificity = \( \frac{TN}{FP + TN} \)

where

False positives (FP): users labeled as diabetic but diagnosed as non-diabetic by the expert.

False negatives (FN): users labeled as non-diabetic but diagnosed as diabetic by the expert.

True positives (TP): users labeled as diabetic and also diagnosed as diabetic by the expert.

True negatives (TN): users labeled as non-diabetic and diagnosed as non-diabetic by the expert.

5.3.2. Resource usage prediction accuracy

In our experiments Mean Absolute Percentage Error (MAPE) is utilized for evaluating the prediction accuracy [46,47]. MAPE expresses accuracy as percentage and is given by the formula:

\[
MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|
\]

where \( A_t \) is the actual value, \( F_t \) is the forecast value and \( n \) is the number of observations in the dataset for which prediction is made.

6. Results and discussion

We conduct two sets of experiments: (1) to classify individuals as Diabetic or Non-diabetic and (2) to predict the future value of underlying resources so as to perform dynamic resource provisioning. For each experiment, we conduct multiple runs and the results obtained are shown thereafter.

6.1. User health status classification

In our experiments, we performed classification of the individuals using K-NN and Naive Bayes classifier. At first, we repeated our experiments for different values of \( k \).

The selection of the \( k \) value is tricky and application dependent. We classified our prototype system by altering the
value of k = 3, 5, 7, 9, 11. At k = 5 we got the maximum correct classification accuracy as shown in Table 5.

Further, the classification accuracy and the sensitivity and specificity measures of the two classifiers, K-NN and NB testing results are given in Table 6. Two diabetic patients were classified incorrectly by the K-NN, whereas three diabetic patients were classified incorrectly by the Naive Bayes. The classification accuracy of 92.59% is obtained for K-NN whereas for NB, the classification accuracy is 85.71%. Also, the sensitivity and specificity measures of K-NN ensure better performance of K-NN as compared to NB.

6.2. Resource usage predictor and elastic scaling

We measure the CPU utilization of our prototype application for varied user demand. In specific, we utilize Amazon CloudWatch to collect the CPU utilization samples. At the beginning of the experiment, a steady load of 80 emulated users is submitted and CPU usage is monitored. The user requests were then increased progressively up to 1200 emulated users and lastly the user request demand is reduced symmetrically to the original 80 users. For each of the user demand scenario, the CPU usage of the nodes is gathered.

The information is utilized by our Best Pattern Policy Selector to determine the specific pattern of CPU utilization and calculate the future value. The predicted future value is then compared with a CPU usage threshold value. The maximum and the minimum limits for CPU usage thresholds are predefined. If the predicted value of future CPU usage is above the maximum threshold, then it indicates that the virtual machines are overloaded and a new virtual machine needs to be created. However, if the predicted value falls below the minimum CPU threshold indicating underutilization of the virtual machines, an instance of the virtual machine is destroyed. The performance of the various approaches for specific CPU usage patterns is shown in Table 7 and the CPU usage trace against the workload is depicted in Figs. 4–7.

Thus, predicting the future CPU usage our prototype system is able to maintain the performance of our CBIHCS application in response to the changing user request load.

| Table 5 – Error (%) for different values of k in K-NN. |
|---|---|---|
| Value of k | Correct (%) | Error (%) |
| 1 | 85.93 | 14.07 |
| 3 | 87.23 | 12.77 |
| 5 | 92.59 | 7.41 |
| 7 | 90.12 | 9.88 |
| 9 | 87.89 | 12.11 |

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Approach</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steady</td>
<td>Preliminary technique</td>
<td>9.94</td>
</tr>
<tr>
<td>Linear</td>
<td>Moving average</td>
<td>10.45</td>
</tr>
<tr>
<td>Repetitive</td>
<td>Preliminary technique</td>
<td>21.34</td>
</tr>
<tr>
<td></td>
<td>Double exponential smoothing</td>
<td>18.56</td>
</tr>
</tbody>
</table>

Fig. 4 – CPU usage trace of CBIHCS for varying workload.

Fig. 5 – CPU usage trace of CBIHCS for steady workload.

| Table 6 – Classification accuracy and sensitivity and specificity measures of the classifiers. |
|---|---|---|---|---|
| Classifier | Diabetic | Non-diabetic | Total accuracy | % | Sensitivity/specificity |
| K-NN | TP = 15 | FP = 2 | 17 | 88.23 |
| | FN = 0 | TN = 11 | 11 | 100.0 |
| Total | 15 | 13 | 28 | 92.59 |
| NB | TP = 14 | FP = 3 | 17 | 82.35 |
| | FN = 1 | TN = 10 | 11 | 90.90 |
| Total | 15 | 13 | 28 | 85.71 |
The forecasted value of CPU usage allows VMs to be dynamically created and destroyed in response to the changing user load.

7. Conclusion

In this paper, we have presented a Cloud based Intelligent Health Care Service (CBIHCS) that performs real time monitoring of user health data collected from various wireless sensory health care equipments. It applies Principal Component Analysis for attribute selection and K-NN and Naive Bayes for user health status classification. In our work, we classified the user as diabetic and Non-diabetic. K-NN achieves better classification accuracy and greater sensitivity and specificity measures indicating that it has better potential for disease identification in real-life scenarios. Apart from that, we utilized standard statistical prediction techniques that derive the resource usage patterns for CBIHCS and propose simple heuristics to perform dynamic infrastructure elasticity. Experimental results conducted on Amazon EC2 clearly demonstrate the effectiveness of our approach in ensuring a stable level of application performance. A cost effective, globally accessible and a highly converged health care solution thus become achievable with CBIHCS.

Conflict of interest statement

The authors have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers’ bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

REFERENCES

computer reminders on processes and outcomes of care, Cochrane Database of Systematic Reviews (2009), CD001096.


[18] https://sugarstats.com/


[27] RadiotherapyGrid: www.beinggrid.eu/radiotherapygrid.html


[38] Z. Lan, Z. Zheng, Y. Li, Toward automated anomaly identification in large-scale systems, IEEE Transactions on Parallel and Distributed Systems 21 (February (2)) (2010).


[40] A.A. Miranda, Y. Borgne, A. Le, G. Bontempi, New routes from minimal approximation error to principal components, Neural Processing Letters 27 (Number 3/June) (2008), Springer.


