ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

HEALTHCARE KNOWLEDGE DISCOVERY FRAMEWORK IN BIG DATA USING C5.0 CLASSIFICATION ALGORITHM

R.Karthikeyan [1], K.Thava Veera Murugan[2]

Karthikhonda77@gmail.com[1], murugan.thavam@gmail.com[2]

HOD, CSE Department, Mohamed Sathak Engineering College, Kilakkarai, India [1]

ME(Student), CSE Department, Mohamed Sathak Engineering College, Kilakkarai, India [2]

Abstract- **Context-awareness is considered as an enabling technology and a rich area of application such as real time personalized health care services and a rich area of big data application. This context aware monitoring has a challenge to analyze the behavior of the data. To overcome the challenge, the paper explains that, the context-aware system uses the Ambient Assisted Living (AAL) system which allows adapting the behavior of data in runtime based on analysis of data generated based on the knowledge discovery-based approach for this system called BDCaM model and stores the data in cloud repositories. Initially this process mines the trends and patterns in the data of an individual patient with the associated probabilities and utilizes that knowledge to learn proper abnormal conditions of those individual patients. The outcomes of these processes are applied in the context-aware decision-making processes for the patient. Here use case is implemented to illustrate the applicability of the framework that is used to discover the knowledge of classification which identifies the true abnormal conditions of patients having variations in blood pressure (BP) and heart rate (HR). The evaluation of our process shows a much better estimation of detecting proper anomalous situations for different types of patients. The effectiveness of our proposed model is shown in the implemented on the case study which shows the obtained accuracy and efficiency.**

Keywords— **Assisted Healthcare, Big data, Cloud Computing, Context awareness, Data Mining, Knowledge Discovery.**

I. INTRODUCTION

Context awareness is a property of mobile devices that is defined complementarily to location awareness. Whereas location may determine how certain processes in a device operate, context may be applied more flexibly with mobile users, especially with users of smart phones. Context awareness originated as a term from ubiquitous computing or as so-called pervasive computing which sought to deal with linking changes in the environment with computer systems, which are otherwise static. The term has also been applied to business theory in relation to contextual application design and business process management issues. Various categorizations of context have been proposed in the past. Dey and Abowd distinguish between the context type's location, identity, activity and time. Kaltz et al identified the categories user & role, process & task, location, time and device to cover a broad variety of mobile and web scenarios. They emphasize yet for these classical modalities that any optimal categorization depends very much on the application domain and use case. Beyond more advanced modalities may apply when not only single entities are addressed, but also clusters of entities that work in a coherence of context, as e.g. teams at work or also single bearers with a multiplicity of appliances.

Some classical understanding of context in business processes is derived from the definition of AAA applications with the following three categories:

- Authentication, which means i.e. confirmation of stated identity
- Authorization, which means i.e. allowance to accrual or access to location, function, data
- Accounting, which means i.e. the relation to order context and to accounts for applied labor, granted license, and delivered goods,

These three terms including additionally location and time as stated.

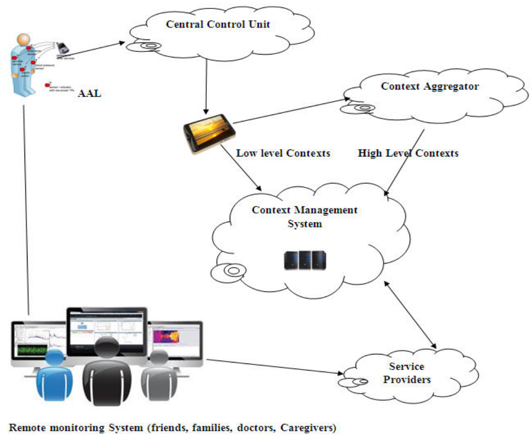

Fig.1. System Architecture

Home health care allows the elderly and medically dependent individuals live in their own home with the assistance of a private care professional. Depending on their medical condition, home health care can help seniors to live independent fuller lives and avoid the need for a nursing home. In home care can involve a variety of services from different types of health care professionals.

In the proposed framework, we develop a new architecture called BDCaM which is a knowledge discovery-based approach that allows the context-aware system to adapt its behavior in runtime by analyzing large amount of data generated in AAL system and used to store in cloud repositories.

After classifications, the CMS sends appropriate notification to the monitoring system or to the AAL system. Using the obtained general behaviour, the CMS is also capable of clustering similar groups of patients so that they can be covered under the same treatment plan we use C5.0 classification model for the context monitoring system (CMS). The CMS consists of a number of distributed cloud servers that hold the big data. It stores the context of millions of patients. Our experimental results show the system robust and efficiency of the system. And we display the graphs for each classification algorithms.

II. RELATED WORKS

An ambient assisted living (AAL) system consists of heterogeneous sensors and devices which generate huge amounts of patient-specific unstructured raw data every day. A data element can be from a few bytes of numerical value (e.g. HR = 72 bpm) to several gigabytes of video stream.
For example, if we assume a single AAL system generates 100 kilobytes data every second on average then it will become 2.93 terabytes in one year. If any system targets to support say, 5 million patients, then the data amount will be 14 exabytes per year 1. Even if a healthcare system targets to analyze only continuous ECG of cardiac patients in real-time inside the cloud environment, then it will produce around 7 PetaBytes data everyday from 3.5 million patients[8]. Including these dynamically generated continuous monitoring data, there are also huge amounts of persistent data such as patient profile, medical records, disease histories and social contacts.

III. METHODOLOGIES AND DISCUSSION

1. *Ambient Assisted Living (AAL)*

Fig.2. Ambient Assisted Living System

The Ambient Assisted Living (AAL) is defined as assistant systems for the constitution of 'intelligent environments' aim to compensate predominantly age-related functional limitations of different target groups through technological information and communication support in everyday life. At the same time they take charge of control and supervision services for an independent course of life. Every person is connected with single Individual patient.

AAL system produces raw data that contain lower level information of a patient health status. The higher level contexts are obtained from these low level data. This system is used for learning daily activities patterns of a patient.

2. *Central Control Unit (CCU)*

The number of AAL systems are connected to the Data Collector which receives the information from the AAL i.e. individual patient's data are collected and sent to the Central Control Unit (CCU). This Central Control Unit is used for recollect the data from the AAL if any path loss occurs. And this data is forwarded to Context Aggregator (CA).
The data collector module runs in the local server (e.g. mobile device), collects the raw data from an AAL system and forwards them to the CA cloud. As described, the CPs convert low level data to high level context and send them back to the CA cloud. From existing research literature we assumed that such capabilities of context conversions already exist.
To make the computation simpler, each context attribute value set Ai is converted to a numerical value. Some context attributes already have numeric values (e.g. HR, BP, room temperature). Numerical annotations are used for contexts having nominal value (e.g. activity).
The static or historical contexts that have Boolean values (e.g. symptoms) are combined in a single binary string which results a decimal value (e.g. 001100 converted to 12).

3. *Context Aggregator (CA)*

The computations like the conversion from low level data to a high level context are performed inside the cloud servers. The local vital signs such accelerometer, blood pressure, ECG Data, GPS Coordinates, RFID Status, Captured images are directly transformed by the AAL system to the CA.

Fig.3. Aggregated value System

The Context Aggregator is uses a context model to integrate all the primitive contexts in a single context state using a context model. It also determines the patient's current situation is normal or not using the past and present contexts. The job of the context aggregator (CA) is to integrate all the primitive contexts in a single context state using a context method. For Example, an increment in HR seems an abnormal condition as a single context, but if the user doing exercise; this can be a normal situation. So, using past and present contexts, it can be determined whether the current user situation is normal or not. Therefore, all the contexts need to be aggregated to classify a situation accurately.

4. ***Context Management System (CMS)***

This method consists of a number of distributed cloud servers that hold the big data and also stores the context histories of millions of patients.

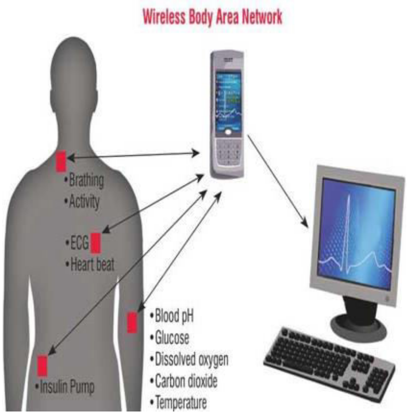

Fig.4. Context Management System

Here there are different machine learning techniques run inside the CMS which infers different personalized and generic rules for various user events. The generic rules are forwarded to the service provider's cloud. After the rule generation, the CMS runs on the data mining classification algorithm called C5.0 algorithm. When the optimized accuracy is achieved, the CMS retains the C5.0 classifier inside the model to classify any new situation. After classifications, the CMS sends appropriate notification to the monitoring system or to the AAL system.

The dataset generated in the previous phase is used to build classifiers for AAL system j and so any new context state can be classified accurately and immediately. The dataset is subdivided into training and test set. Different data mining algorithms are applied over training data and the accuracy of classification is obtained using test data. Comparing the accuracies of different classifiers, the CMS picks the best classifier for decision support. The training and classification process run in distributed clusters inside the CMS.

The CMS uses the classifier generated in the data mining step to classify forthcoming context states and make context-aware decisions. Based on the classification the CMS performs following actions.

- If a situation is normal then do nothing.
- If a situation is abnormal but not dangerous then sends a warning to the user.
- If any vital context attribute has abnormal value then send alert to doctor.
- If two or more context attributes are abnormal or anyone is extremely abnormal then notify to emergency.

A context state will not always be same. It can change due to the change of a patient's condition or low level sensor setup of the AAL system, etc. So the CMS needs to adapt its behaviour every time with these changed situations.

The CMS iteratively runs all the phases to become up-to-date with any altered conditions. Being cloud-based model, the CMS has sufficient storage and processing capability to run the whole process iteratively.

5. ***Context Provider (CPs)***

The context providers (CPs) cloud is the main source for generating contexts. The CA distributes the low level data collected from different AAL systems to multiple CPs. Each CP applies well-known techniques to obtain primitive context from the low level data. For example, in applying data mining on accelerometer data it can identify the current activity of the user using GPS it can identify the location context of the user; it can extract HR value from ECG data and, so on. CP then delivers the converted context with possible high level values to the CA.

6. ***Service Provider (SP)***

In the BDCaM model, the service providers are the cloud servers that sustain the generic medical rules to identify various types of diseases and symptoms. The rules of symptoms and anomalous behaviors are continuously updated by medical experts, doctors and other medical service providers. When any new rule is discovered in the CMS it also triggers the change in the SP cloud. The CMS uses rules of SP for data filtering and classification.

IV. SYSTEM FUNCTIONALITIES

In this section, the functionalities and algorithms of the overall framework are briefly described.

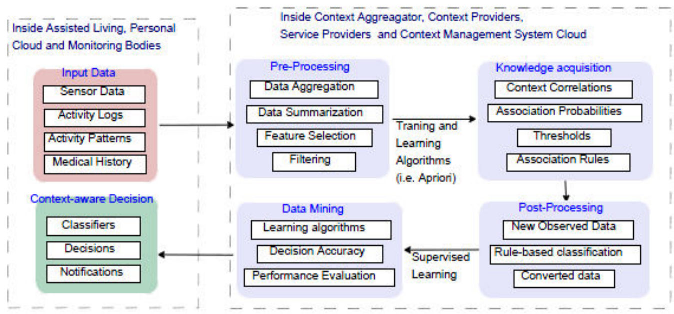

Fig.5. Modules of system functionalities of the BDCaM model

a. ***Context Conversion***

The data collector module runs in the local server (e.g. mobile device), collects the raw data from an AAL system and forwards them to the CA cloud. As described, the CPs convert low level data to high level context and send them back to the CA cloud (Figure 5). From existing research literature we assumed that such capabilities of context conversions already exist. To make the computation simpler, each context attribute value set Ai is converted to a numerical value. Some context attributes already have numeric values (e.g. HR, BP, room temperature). Numerical annotations are used for contexts having nominal value (e.g. activity). The static or historical contexts that have Boolean values (e.g. symptoms) are combined in a single binary string which results a decimal value (e.g. 001100 converted to 12).

b. ***Context Aggregation***

For a single AAL system, after converting all the context attributes to numerical values described above, the context information (as in Definition 2) for each of the domain are generated. Then they are converted to a context state

(Definition 3). This is the aggregated information of all context domains at a specific time of the AAL system. Before converting to context information, some processing such as the elimination of clinically insignificant values is required. Some attributes have discrete time intervals (e.g. BP measured in time interval _t) and some have time duration (e.g. activity x starts at ts and end at te). To represent everything in a single time(t) slot, a standard time interval _ is chosen. That is, a context state is sampled in _ interval. Such sampling process satisfies the velocity property of big data for our model, as each context state is generated and made available using a fixed time interval.
Inside CA, contexts of each domain are summarized in a single time slot using the following steps.

- ***Case 1***: Domain Dk (e.g. vital signs) where every context attribute (e.g. SBP, DBP, BR) are sampled in a fixed time interval (_t) - To find the context attribute value at time t pick the latest value between time t □ _t to t and add it to ItD k .
- ***Case 2:*** Domain Dk (e.g. activity domain) where identified context progress in a time duration (t = ts ! te)- Two separate domains Dx (current activity) and Dy (past activity) are created. During the sampling at time t, the ongoing context observation is added in ItD x . If there is an overlap of contexts between 2 observations (t and t + _) then the most recent observation is added in ItD y.

- ***Case 3:*** For the static context domain Ds – a context value is added in IDs at time t only when this is requested by the CMS.

Once the above processing tasks are completed for every domain Dk and ItD k is generated, the next step is to aggregate all these ItD k in a context state.
A Map Reduce process which runs in multiple clusters inside the CA cloud does this aggregation task for every AAL system.

c. *Trend Analysis*

All generated context states and context information are sent to the CMS cloud. The CMS stores those inside its cloud repository. One of the roles of the CMS is to detect the trend in the dataset. Some of the patterns are detected using statistical analysis. For example, by summing up the duration of sleeping activity it is possible to summarize how many hours the user usually sleeps in one night. Using this statistic, the daily mean of sleep hours can be measured say, from the observation of 1 month's data. So, for any new data if there is a large deviation of sleep hours from the mean, then it is considered as less sleep symptom. When any symptom is detected from the trend, the CMS acts as a CP and notifies CA.
CA then includes this in the IDs of next context state. In this way, other symptoms (e.g. smoking, weight gain, less exercise) can also be detected.

d. *Correlation Learning and Association Rule Mining*

There are two important pre-defined parameters namely minSup and minConf. In the rule mining process, the items that have support value _ minSup and the rules that have confidence value _ minConf are taken. The outcome of this process is the Decision vector Uj = (Rj ; Pj ; Thj). The overall process is described. Note that, 2 Map Reduce processes run inside the algorithm block for calculating support and confidence and filter out the infrequent situations.

Table 1 Sample aggregated data of Patient p1 for rule mining

HR	SBP	DBP	Room Temp.	Activity	Last Activity	Medication	Symptom
68	167	84	0	1	6	0	0
80	152	87	0	1	2	0	0
89	144	72	0	3	1	0	0
78	154	92	1	6	4	0	4
81	165	80	0	1	3	1	0
92	110	84	2	5	3	0	16
78	161	88	0	4	2	1	0
80	170	88	1	1	1	0	8
57	113	94	0	2	4	1	0
63	164	102	1	3	5	0	3

This works for individual AAL system independently and can run parallel in multiple clusters (e.g. Apache Hadoop, Amazon elastic MapReduce) on cloud servers. Therefore, this process can run very fast for millions of context state and can discover patient specific knowledge very quickly.

e. *Algorithm*

1: ***Input***: CjT and Smj for all AAL system j
2: ***Output***: Decision vector Uj = (Rj; Pj; Thj) for all AAL system j
3: **for** each AAL system j in parallel **do**
4: q- Number of context attributes in CjT
5: for each s in Smj , find _(s) in CjT using Map Reduce and prune Smj to Spj by eliminating every s that has _(s) < minSup
6: **for** each combination (x; y) in Spj , find association rule x) y using MapReduce over temporal support of x and (x; y). Here, x are vital sign attributes having the form ai_[vy; vz] and y are other context attributes
7: **calculate** confidence(x) y)
8: **if** confidence(x) y) _ minConf **then**
9: Add [vy; vz] in Thj
10: Add confidence(x) y) in Pj
11: Add assciation rule x) y in Rj
12: **end if**
13: return Uj = (Rj ; Pj ; Thj)
14: **end for**

Algorithm 1: Association Rule Mining

All these learning processes run inside the CMS cloud. When Uj for an AAL system is obtained, the CMS sends this to the PCS of that AAL system where the rules and thresholds are stored.

f. *Data Mining*

The dataset generated in the previous phase is used to build classifiers for AAL system j and so any new context state can be classified accurately and immediately. The dataset is subdivided into training and test set. Different data mining algorithms (e.g. Multi Layer Perceptron, Decision Table, J48 Decision Tree, Radial Basis function, Bayes Network) are applied over training data and the accuracy of classification is obtained using test data. Comparing the accuracies of different classifiers, the CMS picks the best classifier for decision support. The training and classification process run in distributed clusters inside the CMS.

g. *C5.0 Classification Algorithm*

The C5.0 algorithm is a new generation of Machine Learning Algorithms (MLAs) based on decision trees. It means that the decision trees are built from list of possible attributes and set of training cases, and then the trees can be used to classify subsequent sets of test cases.

C5.0 was developed as an improved version of well-known and widely used C4.5 classifier and it has several important advantages over its ancestor. The generated rules are more accurate and the time used to generate them is lower.

In C5.0 several new techniques were introduced:

- ***Boosting***: several decision trees are generated and combined to improve the predictions.
- ***Variable misclassification costs:*** it makes it possible to avoid errors which can result in harm.
- ***New attributes***: dates, times, timestamps, ordered discrete attributes.
- Values can be marked as missing or not applicable for particular cases.
- Supports sampling and cross-validation.

The C5.0 classifier contains a simple command-line interface, which was used by us to generate the decision trees, rules and finally test the classifier. C4.5 is a widely-used free data mining tool that is descended from an earlier system called ID3 and is followed in turn by See5/C5.0. To demonstrate the advances in this new generation, we will compareSee5/C5.0 Release 2.09 with C4.5 Release 8 using three sizable datasets:

- Sleep stage scoring data (***sleep***, 105,908 cases). Every case in this monitoring application is described by six numeric-valued attributes and belongs to one of six classes. C5.0 and C4.5 use 52,954 cases to construct classifiers that are tested on the remaining 52,954 cases.
- Census income data (***income***, 199,523 cases). The goal of this application is to predict whether a person's income is above or below $50,000 using seven numeric and 33 discrete (nominal) attributes. The data are divided into a training set of 99,762 cases and a test set of 99,761.
- Forest cover type data (***forest***, 581,012 cases); also from UCI. This application has seven classes (possible types of forest cover), and the cases are described in terms of 12 numeric and two multi-valued discrete attributes. As before, half of the data -- 290,506 cases -- are used for training and the remainder for testing the learned classifiers.

Since C4.5 is a Unix-based system, results for the Linux version C5.0 are presented to facilitate comparison. Both were compiled using the Intel C compiler with the same optimization settings. Times are elapsed seconds for an unloaded Intel Core i7 (3.4GHz) with 4GB of RAM running 64-bit Fedora Core. Both C4.5 and C5.0 can produce classifiers expressed either as decision trees or rule sets. In many applications, rule sets are preferred because they are simpler and easier to understand than decision trees, but C4.5's rules set methods are slow and memory-hungry.

Table 2 Performance Evaluation of Different Decision Trees

Performance metrics/Decision Trees	C4.5	C5.0	CART
Accuracy in %	80.78	91.11	79.45
Size of the tree(nodes)	2111	772	951
Area Under Curve(AUC)	0.92	0.88	0.94

C5.0 embodies new algorithms for generating rule sets, and the improvement is substantial.

- ***Accuracy:*** The C5.0 rules sets have noticeably lower error rates on unseen cases for the ***sleep*** and ***forest*** datasets. The C4.5 and C5.0 rule sets have the same predictive accuracy for the ***income*** dataset, but the C5.0 rule set is smaller.
- ***Speed:*** C5.0 is much faster; it uses different algorithms and is highly optimized. For instance, C4.5 required nine hours to find the rule set for ***forest***, but C5.0 completed the task in 73 seconds.
- ***Memory:*** C5.0 commonly uses an order of magnitude less memory than C4.5 during rule set construction. For the ***forest*** dataset, C4.5 needs more than 3GB (the job would not complete on earlier 32-bit systems), but C5.0 requires less than 200MB.

Boosting

Based on the research of Freund and Schapire, this is an exciting new development that has no counterpart in C4.5. Boosting is a technique for generating and combining multiple classifiers to improve predictive accuracy.

The graphs above show what happens in 10-trial boosting where ten separate decision trees or rule sets are combined to make predictions. The error rate on unseen cases is reduced for all three datasets, substantially so in the case of forest for which the error rate of boosted classifiers is about half that of the corresponding C4.5 classifier. Unfortunately, boosting doesn't always help -- when the training cases are noisy, boosting can actually reduce classification accuracy. C5.0 uses a novel variant of boosting that is less affected by noise, thereby partly overcoming this limitation.

C5.0 supports boosting with any number of trials. Naturally, it takes longer to produce boosted classifiers, but the results can justify the additional computation! Boosting should always be tried when peak predictive accuracy is required, especially when unboosted classifiers are already quite accurate.

New functionality

C5.0 incorporates several new facilities such as variable misclassification costs. In C4.5, all errors are treated as equal, but in practical applications some classification errors are more serious than others. C5.0 allows a separate cost to be defined for each predicted/actual class pair; if this option is used, C5.0 then constructs classifiers to minimize expected misclassification costs rather than error rates.

The cases themselves may also be of unequal importance. In an application that classifies individuals as likely or not likely to "churn," for example, the importance of each case may vary with the size of the account. C5.0 has provision for a case weight attribute that quantifies the importance of each case; if this appears, C5.0 attempts to minimize the weighted predictive error rate.

C5.0 has several new data types in addition to those available in C4.5, including dates, times, timestamps, ordered discrete attributes, and case labels. In addition to missing values, C5.0 allows values to be noted as not applicable. Further, C5.0 provides facilities for defining new attributes as functions of other attributes.

Some recent data mining applications are characterized by very high dimensionality, with hundreds or even thousands of attributes. C5.0 can automatically winnow the attributes before a classifier is constructed, discarding those that appear to be only marginally relevant.

C5.0 is also easier to use. Options have been simplified and extended -- to support sampling and cross-validation, for instance -- and C4.5's programs for generating decision trees and rule sets have been merged into a single program. The Windows version, See5, has a user-friendly graphic interface and adds a couple of interesting facilities.

V. EXPERIMENTAL RESULTS

A. *Output Design*

The next consideration in the output design is the presentation involved with an information system. The presentation of an output is regarded as an important feature of output design. The presentation of an output represented either in tabular or graphical form or in both forms. A tabular format is preferred in the following conditions:

- When the details dominate the content of the output
- When the contents of the output are classified in groups.
- When the output design are to be compared.

A tabular format is also preferred for the detailed reports. Graphical representation is used to improve the effectiveness of the output because some users prefer to view information in graphic form rather than in rows and columns of the tables. The tabular and graphical formats may be combined together to enhance the presentation of the output.

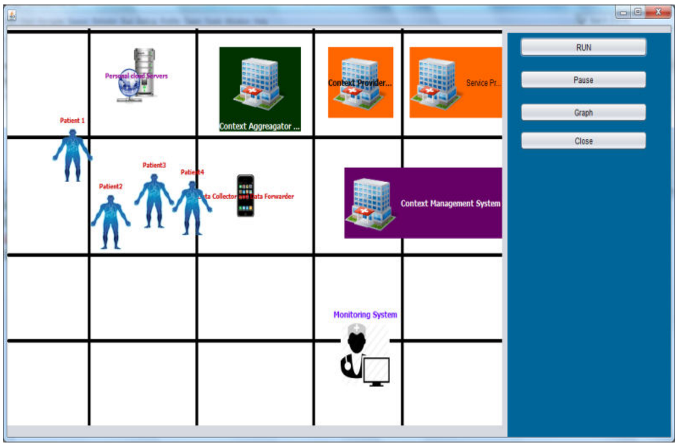

Fig.6. Home Page

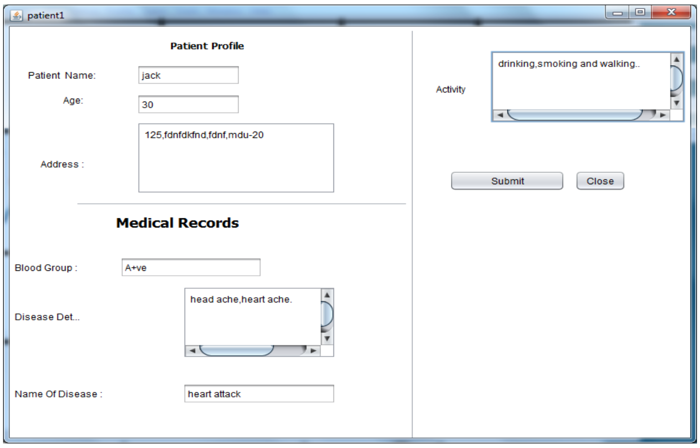

Fig.7. Patient Details

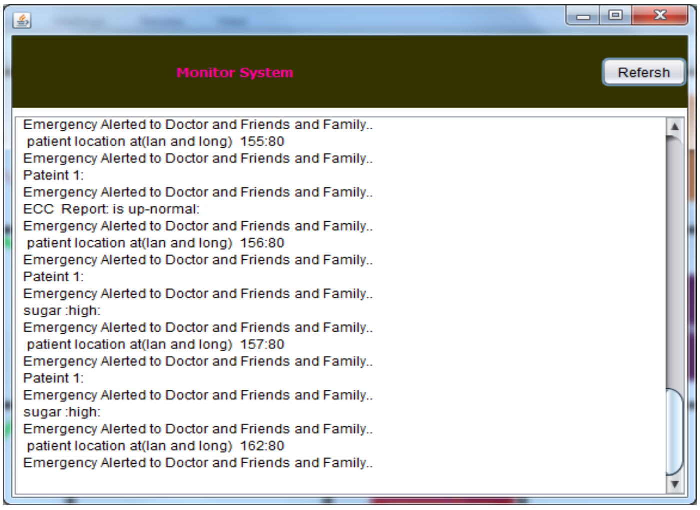

Fig.8. Monitoring Patient Details

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

B. *Graph Representation*

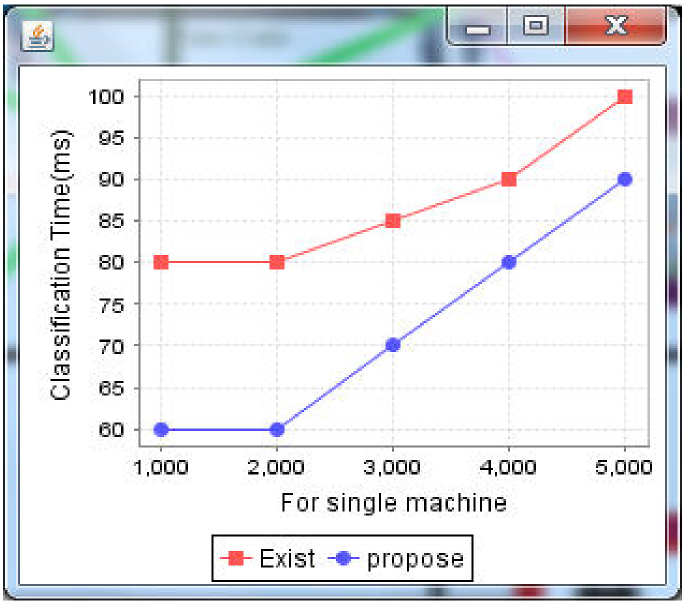

Fig.9.Time Variation of Existing and Proposed System

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a generalized framework for personalized healthcare that leverages the advantages of context aware monitoring, remote monitoring, machine learning and big data. The solution from the framework provides a support the fast growing communities of people with chronic illness who live alone and require assisted care. It easily identifies the emergencies from normal conditions and the stronger relationship between the vital signs and contextual information will make the generated data more consistent and the model will be more accurate for validation. The experiment results show that our system predict the abnormal conditions of patients within the short time.

In our future work, we develop a new framework called BDCaM which is a knowledge discovery-based approach that allows the context-aware system to adapt its behavior in runtime by analyzing large amount of data generated in AAL system and used to store in cloud repositories. We use decision tree induction algorithm for the context management system (CMS).

ACKNOWLEDGMENT

It is my privilege to remember here the grace of GOD ALMIGHTY and all those people who have contributed directly or indirectly in the successful execution of my project. I express my sincere thanks and gratitude to our principal **Dr.J.Mohamed Jahabar BE., ME., PhD (IIT-Bombay), FIE, MISTE, MIEEE, MASME,** for his whole hearted support and help in completing this project. I am taking this opportunity to express my heartfelt gratitude to **Prof. R.Karthikeyan M.E., (Ph.D.,), Head of the Department** for his timely guidance and providing all necessary facilities at the right time for completing this project.

REFERENCES

[1] Abdur Rahim Mohammad Forkan, Ibrahim Khalil, Member, IEEE, Ayman Ibaida, Member, IEEE and Zahir Tari Member, IEEE, "BDCaM: Big Data for Context-aware Monitoring – A Personalized Knowledge Discovery Framework for Assisted Healthcare" 2015.

[2] D. Lymberopoulos, A. Bamis, and A. Savvides, "Extracting spatiotemporal human activity patterns in assisted living using a home sensor network," Universal Access in the Information Society, vol. 10, no. 2, pp. 125–138, 2011.

[3] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, and M. Marchetti, "Cas-mine: providing personalized services in context-aware applications by means of generalized rules," Knowledge and information systems, vol. 28, no. 2, pp. 283–310, 2011.

[4] G. Wu, H. Zhang, M. Qiu, Z. Ming, J. Li, and X. Qin, "A decentralized approach for mining event correlations in distributed system monitoring," Journal of Parallel and Distributed Computing, 2012.

[5] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, vol. 25, no. 6, pp. 599–616, 2009.

[6] R. H. Fagard and V. A. Cornelissen, "Effect of exercise on blood pressure control in hypertensive patients," European Journal of Cardiovascular Prevention & Rehabilitation, vol. 14, no. 1, pp. 12– 17, 2007.

[7] T. Tamura, I. Mizukura, M. Sekine, and Y. Kimura, "Monitoring and evaluation of blood pressure changes with a home healthcare system," IEEE Transactions on Information Technology in Biomedicine, vol. 15, no. 4, pp. 602–607, 2011.

[8] V. Dastjerdi and R. Buyya, "Compatibility-aware cloud service composition under fuzzy preferences of users," IEEE Transactions on Cloud Computing, 2014.

[9] Y. Wang and W. Shi, "Budget-driven scheduling algorithms for batches of map reduce jobs in heterogeneous clouds," IEEE Transactions on Cloud Computing, 2014.

Thava Veera Murugan.K received the B.E degree in Information Technology from Maharaja Engineering College, Tamilnadu, India, in 2013. Currently, He is post graduate student in Computer Science and Engineering, Mohamed Sathak Engineering College (MSEC), Tamilnadu, India. Him current research area includes Data Mining and Big Data.