A CLOUD-BASED, PREDICTIVE AND CONTEXT-AWARE SYSTEM FOR AMBIENT ASSISTED LIVING

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy (Computer Science)

Abdur Rahim Mohammad Forkan
B.Sc. Computer Science and Engineering (CSE)
Bangladesh University of Engineering and Technology (BUET)

School of Computer Science and Information Technology
College of Science, Engineering, and Health
RMIT University
Melbourne, Australia.
June, 2016
Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Abdur Rahim Mohammad Forkan
20th June, 2016
Acknowledgement

All the praises and thanks be to God who gave me the opportunity and endurance to reach this endeavour. I would like to express my sincere gratitude and appreciation to a group of people for their kind support, guidance, advice and inspiration. This work would not have been possible without their cooperation.

First and foremost, I would like to thank my senior supervisor, A/Prof. Ibrahim Khalil, who guided me to accomplish my research goals. His effective supervision, constructive criticism, endless encouragement and continuous support has always kept me motivated and ensured a smooth candidacy for completion of this research. I would like to thank him for his patience and sincerity in helping me reach the finishing line of my PhD journey. I would also like to thank my associate supervisor, Prof. Zahir Tari, for his advice and support.

I would also like to thank my family. To my father, A. H. M. Shahjahan and my mother, Ferdousi Akhter, who always provided me mental strength and courage while being thousands of kilometres away. I am grateful for the sacrifice they made for me. Their prayers always helped me a lot. A special thanks to my beloved wife, Tasnim Nasrin, who has been all time companion on this journey. Thanks for the support, encouragement, patient, unwavering love and care that I continuously receive. I would like to thank my elder sisters for their inspiration and to my in-laws for their appreciation and prayers throughout my entire life.

I am thankful to my colleagues, Dr. Heshan, Dr. Ayman, Dr. Mahathir, Dr. Azami and Dr. Naimah, who helped me with suggestions and materials during my research. I appreciate the support of RMIT CS&IT for providing a good research environment. I would like to express my gratitude to all my friends in and outside of RMIT that directly or indirectly supported me throughout this time.

Last but not least, I would like to thank my scholarship provider, National ICT Australia (NICTA) and RMIT CS&IT for giving me the opportunity to do PhD study and develop my career. I hope this piece of work would benefit the research community in my area.
Credits

Portions of the material in this thesis have previously appeared in the following publications:


- A. R. M. Forkan and I. Khalil, “A clinical decision-making mechanism for context-aware and patient-specific remote monitoring systems using the correlations of multiple vital
signs”, *Computer Methods and Programs in Biomedicine*, DOI: 10.1016/j.cmpb.2016.10.018, October 2016. (ERA Rank A*) - Chapter 5

- “ViSiBiD: A learning model for early discovery and real-time prediction of severe clinical events using Vital Signs as Big Data”, *Computer Networks*, **Provisionally accepted with a minor revision**, October, 2016. - Chapter 6


- “Patient clustering using entropy-driven dynamic partitioning on correlated vital signs”, *IEEE Transactions on Knowledge and Data Engineering*, **Submitted**, June 2016. - Chapter 7
Contents

Declaration ii
Acknowledgement iii
Credits iv
Contents vi
List of Figures xiii
List of Tables xix
Abstract x

1 Introduction 4
  1.1 Background and Motivations .............................................. 5
  1.2 Preliminaries ................................................................. 7
  1.3 Research Challenges ...................................................... 11
  1.4 Research Questions ....................................................... 14
  1.5 Limitations of Existing Solution ........................................ 17
  1.6 Research Contributions .................................................. 19
  1.7 Thesis Organization ....................................................... 24

2 A Cloud-oriented Context-aware Middleware for Assisted Living 26
  2.1 Motivations and Contributions ........................................... 27
    2.1.1 Motivations ............................................................. 27
    2.1.2 Contributions .......................................................... 28
  2.2 Related Work ............................................................... 29
## Contents

2.3 CoCaMAAL System Overview ........................................... 31
2.4 System Description ......................................................... 33
  2.4.1 AAL System .......................................................... 33
  2.4.2 Context Providers and Aggregator Cloud ......................... 36
    2.4.2.1 Context Providers ............................................. 36
    2.4.2.2 Context Aggregator .......................................... 37
    2.4.2.3 Context Modeling ............................................. 38
  2.4.3 Service Providers Cloud .............................................. 39
  2.4.4 Context-aware Middleware (CaM) .................................... 40
    2.4.4.1 Context Management System (CMS) ......................... 41
    2.4.4.2 Context Storing .............................................. 42
    2.4.4.3 Context Retrieval ............................................ 42
    2.4.4.4 Context Manipulation ....................................... 43
    2.4.4.5 Service Mapping .............................................. 43
    2.4.4.6 Self-adaptation .............................................. 46
    2.4.4.7 Service Discovery ........................................... 47
    2.4.4.8 Security Service ............................................ 47
  2.4.5 Monitoring and Data Visualization Service ....................... 47
2.5 Case Study .............................................................. 47
2.6 Experimental Results .................................................... 51
2.7 Conclusion .............................................................. 55
2.8 Epilogue ................................................................. 55

3 A Personalized Knowledge Discovery Framework ....................... 56
  3.1 Motivations and Contributions ........................................ 57
    3.1.1 Motivations ...................................................... 57
    3.1.2 Contributions ................................................... 58
  3.2 Related Work ............................................................ 58
  3.3 Architecture ................................................................ 59
    3.3.1 Ambient Assisted Living (AAL) Systems ...................... 60
    3.3.2 Personal Cloud Servers (PCS) .................................. 60
    3.3.3 Data Collector and Forwarder (DCF) ......................... 62
    3.3.4 Context Aggregator (CA) ....................................... 62
    3.3.5 Context Providers (CP) ......................................... 62
3.3.6 Context Management System (CMS) .................................. 63
3.3.7 Service Providers (SP) ........................................... 63
3.3.8 Remote Monitoring Systems (RMS) ................................. 63
3.4 Definitions and Concepts ............................................. 64
3.5 System Functionalities .............................................. 65
  3.5.1 Context Conversion ............................................. 65
  3.5.2 Context Aggregation ........................................... 66
  3.5.3 Trend Analysis .................................................. 67
  3.5.4 Correlations Learning and Association Rule Mining .......... 68
  3.5.5 Learning using the Association Rules .......................... 71
  3.5.6 Data Mining .................................................... 71
  3.5.7 Context-aware Decision Support ............................... 71
3.6 Use case Implementation ............................................ 72
  3.6.1 Description of use case ....................................... 72
  3.6.2 Synthetic data generation ..................................... 73
  3.6.3 Cloud evaluation .............................................. 74
  3.6.4 Implementation ............................................... 75
3.7 Results and Discussion ............................................. 77
3.8 Conclusion ......................................................... 80
3.9 Epilogue ........................................................... 80

4 Long-term Change Detection and Abnormality Prediction 81
  4.1 Motivations and Contributions ................................... 82
    4.1.1 Motivations .................................................. 82
    4.1.2 Contributions ............................................... 83
  4.2 Related Works .................................................. 84
  4.3 Methodology ..................................................... 84
    4.3.1 System Description .......................................... 85
      4.3.1.1 Background ............................................ 85
      4.3.1.2 A motivating scenario ................................. 86
      4.3.1.3 Proposed fusion-based architecture .................. 86
    4.3.2 Change detection in activity and location patterns using HMM 87
      4.3.2.1 Data description ..................................... 88
      4.3.2.2 Modelling observable Markov Model ................. 88
CONTENTS

4.3.2.3 Modelling Hidden Markov Model ........................................... 89
4.3.2.4 Abnormality prediction using HMM ......................................... 91

4.3.3 Change detection in lifestyle pattern and routine behaviour ................... 92
4.3.3.1 Routine behaviour discovery ................................................. 92
4.3.3.2 Statistical model design ....................................................... 93
4.3.3.3 Abnormality classification .................................................... 94

4.3.4 Physiological change detection in different vital signs ........................ 94
4.3.4.1 Vital signs and related disease identification .............................. 95
4.3.4.2 Forecasting model development ............................................. 95

4.3.5 Information fusion and decision making using fuzzy rules .................. 97
4.4 Experimental Results ....................................................................... 99
4.4.1 Case study description .................................................................. 99
4.4.2 Implementation and validation of proposed HMM ............................. 100
4.4.3 Implementation and validation of routine behaviour change detection method .......................................................... 103
4.4.4 Implementation and validation of trend detection in vital signs .......... 104
4.4.5 Implementation of fuzzy rule-based model and wellness detection .... 105

4.5 Conclusion ....................................................................................... 108
4.6 Epilogue ......................................................................................... 108

5  Patient-specific Future Clinical Event Prediction ................................. 109
5.1 Motivations and Contributions ......................................................... 111
5.1.1 Motivations .............................................................................. 111
5.1.2 Contributions ........................................................................... 111
5.2 Related work .................................................................................... 112
5.3 The framework for clinical decision support system (CDSS) ................. 113
5.3.1 Patient data collection ................................................................. 113
5.3.2 Correlations of vital signs for clinical inference ............................... 113
5.3.3 Data preprocessing and feature generation .................................... 115
5.3.4 Multi-label classification (MLC) and evaluation ............................ 116
5.3.5 Knowledge adaptation ................................................................. 116
5.3.6 Clinical decision support ............................................................. 117
5.4 Research methodology and implementation .................................... 117
5.4.1 Data sources .............................................................................. 117
5.4.2 Data cleaning and preprocessing ........................................ 118
5.4.3 Segmentation ............................................................... 118
5.4.4 Feature extractions and class labelling ............................... 118
5.4.5 Multi-label classification .................................................. 119
  5.4.5.1 Binary relevance (BR) ........................................... 120
  5.4.5.2 Classifier chain (CC) ........................................... 120
  5.4.5.3 Pruned Sets (PS) ................................................ 121
  5.4.5.4 Fourclass Pairwise (FW) and Ranking + Threshold (RT) ... 121
  5.4.5.5 Algorithm Adaptation method .................................. 122
  5.4.5.6 Evaluation Measure ............................................... 122
5.4.6 Abnormality prediction ................................................... 122
5.4.7 Incremental and patient-specific learning .......................... 123
5.5 Experimental evaluations and results ................................... 123
  5.5.1 Selection of window size ........................................... 123
  5.5.2 Selection of forecast period ....................................... 124
  5.5.3 Classifier performance evaluation ................................. 125
    5.5.3.1 Performance for a single patient .......................... 125
    5.5.3.2 Performance for multiple patients ......................... 127
    5.5.3.3 Performance in terms of model building time ............ 128
  5.5.4 Performance for using correlations ............................... 128
  5.5.5 Discussions ......................................................... 130
5.6 Conclusion ................................................................. 130
5.7 Epilogue ........................................................................ 131

6 Early Prediction of Abnormal Clinical Events Using Known Knowledge 132
  6.1 Motivations and Contributions ........................................ 133
    6.1.1 Motivations ......................................................... 133
    6.1.2 Contributions ..................................................... 134
  6.2 Related work .................................................................. 135
  6.3 Preliminary analysis ...................................................... 136
    6.3.1 Vital sign correlations ............................................ 136
    6.3.2 Clinical events ..................................................... 138
    6.3.3 Principal Component Analysis (PCA) .......................... 139
  6.4 System setup .................................................................. 140
6.4.1 Model description .............................................. 140
6.4.2 Dataset description ........................................... 141
6.4.3 Data formulation .............................................. 142
6.4.4 Analysis using PCA ........................................... 143
6.4.5 Training and Learning ........................................ 144
6.4.6 Learning in the cloud ....................................... 144
6.4.7 Real-time and personalised monitoring ...................... 145

6.5 The PEACE-Home model ......................................... 146
6.5.1 Mapping bio-signals to observations ......................... 147
6.5.2 Parameters of HMM ........................................... 147
6.5.3 Estimation of model parameters - Training .................. 148
6.5.4 Validation, classification and probabilistic prediction ... 149

6.6 The ViSiBiD model ................................................ 150
6.6.1 Segmentation .................................................. 150
6.6.2 Feature Extraction ............................................ 150
   6.6.2.1 Correlation coefficients ............................... 151
   6.6.2.2 Wavelet coefficients .................................. 151
   6.6.2.3 Statistical features .................................... 152
6.6.3 Training and Learning ....................................... 153
6.6.4 Feature selection ............................................. 153
6.6.5 Classification ................................................ 154

6.7 Experimental results and discussions .......................... 154
6.7.1 The PEACE-Home model ..................................... 154
   6.7.1.1 HMM modelling ......................................... 154
   6.7.1.2 Prediction performance ................................. 155
   6.7.1.3 Probability Prediction in real-time monitoring ...... 156
   6.7.1.4 Evaluation using cloud frameworks ................. 157
   6.7.1.5 Comparison with other models ....................... 158
6.7.2 The ViSiBiD model ............................................ 159
   6.7.2.1 Training and learning ................................. 159
   6.7.2.2 Clinical event prediction result ........................ 160
   6.7.2.3 Analysis of accuracy with data size .................. 161
   6.7.2.4 Analysis of model building time ...................... 161
   6.7.2.5 Scalability analysis .................................... 162
CONTENTS

6.7.2.6 Model performance using selected features 163
6.7.2.7 Prediction in continuous monitoring 164
6.7.2.8 Performance comparison 165

6.8 Conclusion 166
6.9 Epilogue 166

7 Patient Clustering by Dynamically Partitioning Patient States 167
7.1 Motivations and Contributions 169
7.1.1 Motivations 169
7.1.2 Contributions 169

7.2 Related work 170

7.3 The patient clustering method 172
7.3.1 Data normalization 173
7.3.2 Outlier detection 174
7.3.3 Unsupervised pattern discovery 175
7.3.3.1 Similarity measure 175
7.3.3.2 Transform multi-dimensional observation into single-dimensional vector 176
7.3.3.3 Dynamic partitioning 177
7.3.3.4 Cluster analysis 179
7.3.4 State feature extraction 179
7.3.5 Identify similar subgroups 179

7.4 Experimental results 181
7.4.1 Data Sets 182
7.4.2 Outlier detections 183
7.4.3 Dynamic pattern discovery 183
7.4.4 Cluster Analysis 184
7.4.5 Patient classification 188

7.5 Conclusion 191

8 Conclusion 193

Bibliography 200
# List of Figures

1.1 (a) A demonstration of wearable sensors on a human body. (b) A conceptual BSN architecture of the proposed AAL system. (c) An example of wearable textile sensors.  

1.2 There are different types of ambient device and white goods installed in AAL system including body sensors. Some sample devices are illustrated in here.  

1.3 The concept of context in our model.  

1.4 The workflow of real-time monitoring, classification, predictions and clinical decision support.  

1.5 A scalable, predictive and protective context-aware system and related research problems.  

1.6 A summary of research questions, thesis contributions and chapter organizations.  

2.1 The Generic architecture of CoCaMAAL model showing 5 major distributed cloud-based components of the proposed system.  

2.2 **Data Acquisition** The raw sensor data produced by body sensors and other devices are transmitted to a local workstation using a connected medium. The Local workstation then forwards the data to the cloud server using a cloud gateway.  

2.3 **Data flow in a single AAL:** Raw data from the AAL system is converted to context by the aggregator cloud. The service provider cloud contains the cloud storage for service rules. The context-aware middleware cloud manages incoming context and finds assistive actions. Then it responds back to the AAL system as a context-aware service.  

2.4 **Flow diagram of Context Classifier:** All the context providers perform this generic classification process to extract the high-level features. The method inside each of the square blocks can be different based on input and output data.
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td><strong>Context Model using OWL</strong> Each context entity has some attributes to describe some basic properties of the entity. Some context entities are part of parent entity such as characteristics, diseases, preference, social and health ontologies are part of person ontology. Each of those entities has some more children to describe them. The relation among different entities is also shown here.</td>
<td>38</td>
</tr>
<tr>
<td>2.6</td>
<td><strong>Dynamic scalability of distributed CMS</strong> A snapshot of how the CMS of CoCaMAAL works.</td>
<td>42</td>
</tr>
<tr>
<td>2.7</td>
<td><strong>Context Tree</strong> Every context sample and service of our system is converted to such a context tree to find the service mapping.</td>
<td>43</td>
</tr>
<tr>
<td>2.8</td>
<td><strong>Evaluation</strong> The experimental setup of simulated model for evaluating the case studies described in this section.</td>
<td>48</td>
</tr>
<tr>
<td>2.9</td>
<td>The $M/G/1$ queuing system of the CoCaMAAL model</td>
<td>53</td>
</tr>
<tr>
<td>2.10</td>
<td>(a) Using the lower value of the request arrival rate ($\lambda$). (b) $\lambda$ is high. That is, requests are arriving in high volume from a massive number of AAL systems. For both (a) and (b) the average response time ($W$) increases with the increase of request arrivals. The response time is still very fast (only 0.11 sec) even with a high request arrival rate($\lambda = 60$).</td>
<td>54</td>
</tr>
<tr>
<td>2.11</td>
<td>The average response time($W$) varies with the distribution of service time in different processing nodes for different observations. Using only a single ultra-fast server with $k$ processing slots, the average response time (using the same arrival rate) is very low for each observation. For $k \geq 5$, the response time of each observation is very similar.</td>
<td>54</td>
</tr>
<tr>
<td>3.1</td>
<td>The complete architecture of BDCaM model showing all the cloud-based components of the system.</td>
<td>60</td>
</tr>
<tr>
<td>3.2</td>
<td>Data flow between cloud components of BDCaM model.</td>
<td>61</td>
</tr>
<tr>
<td>3.3</td>
<td>Big data case for context-awareness in AAL</td>
<td>61</td>
</tr>
<tr>
<td>3.4</td>
<td>Modules of system functionalities of the BDCaM model. Data collection and context-aware actions happen inside AAL system and personal cloud. Context processing, knowledge acquisition, data mining etc. are performed in distributed cloud components.</td>
<td>66</td>
</tr>
<tr>
<td>3.5</td>
<td>The change in no of rules with minSupp value</td>
<td>76</td>
</tr>
<tr>
<td>3.6</td>
<td>Classification time for different classifiers using 1000 to 34000 context states for hypertensive patient.</td>
<td>79</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

4.1 The complete architecture of context-aware change detection model. 87
4.2 Markov process for activity sequences. 89
4.3 Observable Markov process. 89
4.4 Different regions and related parameters to identify abnormality in routine behaviour. The distribution is Gaussian. Normal pattern shown in solid line and a behavioural shift shown in dashed line. 93
4.5 Time-series graph of 2 vital signs (HR, BP) with 96 observations for 3 users. 96
4.6 Estimation of body temperature value for detecting fever symptoms. Here, the forecast is for 10 periods. We can see that the forecast graph follow a linear trend. 97
4.7 Model performance improves when the number of hidden states increase. But gain is low after hidden states reaches 40. So the model with 40 hidden states is considered the optimized model and that is selected as standard model for classification. 101
4.8 The Hidden Markov Model for detecting abnormal activity. 101
4.9 Observed value of $\log P(O|\lambda)$ for 20 normal and 20 abnormal samples. The line indicates the cluster boundary to distinguish between the two classes. 102

5.1 Strong correlations between a pair of vital signs. (a)-(e) examples of positive correlation (f) example of negative correlation. Here SBP, DBP and MBP refers to Systolic, Diastolic and Mean blood pressure respectively. 110
5.2 A framework of a clinical decision support system (CDSS) based on multi-label classification (MLC) using the correlations of multiple vital signs. 114
5.3 Observation, lead and prediction window time. 115
5.4 Selection of optimal window size for calculating features. Three evaluation measures from 30 days monitoring data of patient a40493n using different window sizes. Here the multi-label classifier is BR and learning algorithm is Random Tree. Window size 10 achieved best hamming score, accuracy and F1 value in compare to other window sizes. 124
5.5 Performance for single patient for different lead time using $w=10$. Three evaluation measures from 30 days monitoring data of patient a40493n using $l$ 1 to 7. Here the multi-label classifier is BR and learning algorithm is Random Tree. For all 7 cases the classification has good hamming score value ($\geq 85\%$). 125
5.6 Hamming score of 85 patients separately using the statistics from 24 classifier combinations. 126
5.7 Performance measure of different classifier combination using random 24 hours data of patient a42141n. ................................................................. 127
5.8 Average performance of different combination of multi-label classifiers and learning algorithms in terms of hamming score. These average values are calculated from random 24 hours data of each 85 patients using $w=10$ and $l=7$. Here, for most cases hamming score is above 90%. The performance of Random Tree and SMO is better than J48. ................................................................. 127
5.9 Hamming score of 24 different classifier combinations using the statistics of 85 patients ................................................................. 128
5.10 Average building time for different classifier using random 24 hours data of 85 patients. PS-Random Tree is the fastest process. SMO takes more time than Random Tree and J48. FW has very high building time. ......................... 129

6.1 High correlation between HR and BP. The top 2 plots show the time-series, changes in HR, Systolic BP (SBP), Diastolic BP (DBP) and mean BP (MBP) with time (in minutes). The bottom 2 plots show the correlation between HR and MBP from corresponding upper plots. Here strong (a) positive and (b) negative correlation is shown between HR and MBP. ................................................................. 137
6.2 Scatter plot of correlations between HR-SBP and HR-MBP pairs after combing data of many similar patients. Warmer color indicate higher density. ................ 137
6.3 Correlations among HR, MBP and RR in a single patient. Low value in colour bar indicate all vitals have high value and vice versa. Depending on correlations among 3 vitals patients can be in different clinical states. ......................... 138
6.4 Validation of clinical event separation from a large number of patients using PCA 140
6.5 The scenario of the developed system showing its components. .................. 141
6.6 Separation of 5 clinical events by PCA method ................................. 143
6.7 Separation of 5 clinical events using observed data before that event by PCA method 144
6.8 The model learning process using multiple vital sign data from several group of patients (normal and with different clinical conditions). The learning phases are performed in the cloud. The output of this learning process is multiple HMMs and these learned models are stored in the cloud storage. ......................... 145
6.9 Real-time clinical abnormality prediction for a patient with unknown clinical condition using multiple vital sign data. Here CE means a clinical event. The current observations are matched with stored HMMs and the probabilities of matching are continuously estimated. .................................................. 146

6.10 The conversion process from bio-signals to observation sequence along with HMM model parameters. The observation sequence is extracted from 1 hour observed data of the actual clinical event. .................................................. 148

6.11 The process of segmentation to compute the features from observation window and class from target window .................................................. 150

6.12 Discrete wavelet transform and decompositions ......................... 152

6.13 (a) Change of $\log P(O|\lambda)$ value, (b) model building time (in minutes) with the change of number of hidden states for 5 HMM models. In (a) Lighter colours indicate higher value in $\log P(O|\lambda)$ and vice versa. In (b) Darker colours indicate lower value in model building time. The goal is to number of states which have high $\log P(O|\lambda)$ and low model building time in all 5 models. .................. 155

6.14 The upper plot shows the continuous monitoring values of 6 bio-signals of a patient for 1000 minutes. The lower plot depicts continuous $\log P(O|\lambda)$ values in a heatmap table obtained for 5 HMMs. The warmer colours indicate higher likelihood probability value. .................................................. 157

6.15 Training time and prediction accuracy for J48 and RF with the change in training data size. .................................................. 161

6.16 Model building time (in seconds) in different EC2 instances using J48 and RF classifiers. .................................................. 162

6.17 The change in model building time (in seconds) and accuracy (%) with the change in number of datanodes in a Hadoop cluster. .................................................. 162

7.1 The workflow of the proposed patient clustering method. First, different states of multiple patients with unknown medical conditions are identified dynamically in parallel from their multiple vital signs using similarity analysis based on entropy values. Then, a hierarchical clustering is used on features of different patient’s states to identify subgroups of similar patients. .................................................. 173

7.2 The distribution of 12 clinical classes in 35 patients of Dataset-1 used in our evaluation. This is used as ground truth for the patient clustering experiment. 182
7.3 Outlier detection using average relative density measure. The top plot shows the
time-series of six vital signs of a patient for 185 minutes. In the middle plot, the
detected outliers are shown in vertical lines. The bottom plot shows the changes
in the density measure over time computed from six vital sign values of the top
plot. ................................................................. 184

7.4 Dynamic pattern discovery from six correlated vital signs. The plots in the left side
show the time-series of six correlated bio-signals for 3 patients having CHF/Pul-
monary Edema in first 24 hours. The plots in the right side show the patterns of
states identified using an entropy-driven clustering method on 3 principal compo-
nent axes. We can observe that the number of discovered states are different in 3
patients. ................................................................. 185

7.5 Visualization of state distributions for patient 4 in (a) 3D and (b) 2D using PCA.
Here PC-1, PC-2 and PC-3 are first 3 principal components computed from score
matrix of original data. ..................................................... 186

7.6 Number of discovered dynamic states of all patient samples for two experimental
datasets. ................................................................. 187

7.7 Cluster analysis for 9 patients of Dataset-1. From the labelled information we
know that 3 of these patients (6,7 and 9) belongs to the same clinical class and
others are in different classes. ............................................. 187

7.8 Transition of states in 24 hours data for a single patient. ................................. 188

7.9 Consistency validation of dynamic partitioning method using Silhouette plots of
12 patients in Dataset-1. ..................................................... 188

7.10 Correlation map among all patients for 2 experimental datasets. ....................... 189

7.11 Hierarchical cluster tree for Dataset-1 containing 35 patients along X-axis. The
original distribution of patients are shown at top and identified subgroups are
shown at bottom in different colours. .................................... 190

7.12 Hierarchical cluster tree and similarity matrix for Dataset-2 containing 34 patients
along X-axis. The identified subgroups are shown at bottom in different colours. 191

7.13 Clustering result for Dataset-1 using k-means clustering ................................. 191
List of Tables

2.1 Some mentionable research works describing the framework of a Context-aware System (CaS) .......................................................... 30
2.2 Examples of some typical body sensors and their use in AAL ............... 34
2.3 Service rules of detecting possible heart attack .................................. 40
2.4 Example of some assistive services, their rules and the provider of the service ... 41
2.5 The components considered in the experimental setup of the simulated prototype. 48
2.6 Aggregated high-level context samples ............................................... 49
2.7 Components developed for simulated prototype ................................... 50
2.8 Average Service Response Time for different Context, Service Provider (SPs), and service load. **Note that in the 2nd column of item 2, 1 and Activity refers to Activity, Location and Time including Physical parameters of row 1 and so on for other rows of row column .......................... 52
2.9 General distribution of Service Time of five different sets for \( k=10 \) and \( \lambda = 2 \) sec 53
3.1 General medical rules to identify different diseases related to the variation in BP and HR .............................................................. 73
3.2 Different patient categories considered in the experiment and their average values of vital signs ................................................. 74
3.3 Domains and context attributes used in the experiment .......................... 75
3.4 Symptoms that are considered in the experiment ................................... 76
3.5 Sample aggregated data of Patient P1 for rule mining ........................... 76
3.6 Class attributes used for situation classification ................................... 77
3.7 Comparison of BDCaM model generated rule-based classifications with generic rule-based classification .............................................. 78
3.8 Comparison of association rule generation time using single and multi-core processing on same machine ................................. 78
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9</td>
<td>Anomaly detection accuracy and false positive rate for 3 different types of patient over 1 year data</td>
</tr>
<tr>
<td>4.1</td>
<td>Illustration of patient logs for our system</td>
</tr>
<tr>
<td>4.2</td>
<td>Daily activity log. Where each $x(t_j) \in X$, $X = {x_1, x_2, \ldots, x_N}$, $N$ is the number of activities considered in the design. $t_k-t_0 = 24$ hours</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of activity and location sequences</td>
</tr>
<tr>
<td>4.4</td>
<td>Normal routine behaviour of the user. Distribution is Gaussian. Mean in start time, SD in minutes. Min and Max duration is in minutes</td>
</tr>
<tr>
<td>4.5</td>
<td>Normal range of vital signs</td>
</tr>
<tr>
<td>4.6</td>
<td>Different diseases for the change in vital signs</td>
</tr>
<tr>
<td>4.7</td>
<td>Estimation of heart rate value using Holt’s method for Tachycardia prediction</td>
</tr>
<tr>
<td>4.8</td>
<td>Daily activities and location logs</td>
</tr>
<tr>
<td>4.9</td>
<td>Vital sign logs</td>
</tr>
<tr>
<td>4.10</td>
<td>Hidden state sequence using Viterbi algorithm</td>
</tr>
<tr>
<td>4.11</td>
<td>Activity sequence classification accuracy using different HMM for 10 users</td>
</tr>
<tr>
<td>4.12</td>
<td>Classification for routine behaviours</td>
</tr>
<tr>
<td>4.13</td>
<td>More than 1 anomalous behaviour cases</td>
</tr>
<tr>
<td>4.14</td>
<td>Trend detection result in different vital signs</td>
</tr>
<tr>
<td>4.15</td>
<td>Decision using fuzzy rules from detected abnormalities in different domains</td>
</tr>
<tr>
<td>4.16</td>
<td>Type of decisions from fuzzy rule model</td>
</tr>
<tr>
<td>4.17</td>
<td>Type of abnormalities for different context domains</td>
</tr>
<tr>
<td>4.18</td>
<td>Detailed accuracy by class</td>
</tr>
<tr>
<td>4.19</td>
<td>Comparison of our model with some context-aware models</td>
</tr>
<tr>
<td>5.1</td>
<td>The six vital signs and their acronym used in this study</td>
</tr>
<tr>
<td>5.2</td>
<td>Normalized class value for different ranges of six vital signs</td>
</tr>
<tr>
<td>5.3</td>
<td>Performance of different multi-label classifier using the features of random 24 hours data of patient a40215n. Here $w=10$ and $l=7$</td>
</tr>
<tr>
<td>5.4</td>
<td>A comparison of prediction accuracies for 6 vital signs as individual using 3 classification algorithms. This is using same dataset and settings as in Table 5.3. Here, WCF means with all correlated features of all vitals, OF means using Only features of corresponding vital and SF using subset of features selected by a feature selection algorithm from all features.</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Threshold values of various clinical condition (CC)</td>
<td>138</td>
</tr>
<tr>
<td>6.2</td>
<td>The occurrence of our targeted clinical events for evaluation, their acronyms and class labels.</td>
<td>139</td>
</tr>
<tr>
<td>6.3</td>
<td>The confusion matrix after performing the classification using validation set of 5 classes. Here 96.6% accuracy is obtained.</td>
<td>156</td>
</tr>
<tr>
<td>6.4</td>
<td>HMM training time (TT) for case NNNN in different Amazon EC2 instances</td>
<td>158</td>
</tr>
<tr>
<td>6.5</td>
<td>Comparison of our model with the results and features obtained in similar works</td>
<td>158</td>
</tr>
<tr>
<td>6.6</td>
<td>Classification results for different classifiers using different forecast horizons on cloud environment in terms of F-measure, Accuracy, Mean Square Error (MSE), model building and validation times. For all types of forecast horizons, the best result is obtained for Random Forest classifier which is highlighted here.</td>
<td>160</td>
</tr>
<tr>
<td>6.7</td>
<td>Accuracy and model building time for the best 3 classifiers using different selected feature sets. The optimum model in terms of accuracy, building time and features is highlighted in the table.</td>
<td>164</td>
</tr>
<tr>
<td>6.8</td>
<td>Weighted Average of FP and TP for 5 different classes using level 4 wavelet coefficients and correlation coefficients as features, Random Forest classifier, 2 hours forecast gap and 1000 minutes continuous data of 6 bio-signals for 10 different patients.</td>
<td>165</td>
</tr>
<tr>
<td>6.9</td>
<td>Comparison of ViSiBiD model with similar methods in literature</td>
<td>165</td>
</tr>
<tr>
<td>7.1</td>
<td>The occurrence of clinical event targeted for the evaluation and their sample size and index number</td>
<td>183</td>
</tr>
</tbody>
</table>
Abstract

Ambient assisted living (AAL) technology provides the opportunity for people with disabilities or chronic medical conditions to lead independent lives in their home, relying on wearable sensors and intelligent processing services for the care of their health. The concept of context-awareness is used in AAL to identify current medical conditions and potential abnormalities of a user (patient or elderly). It facilitates the decision-making tasks of health professionals in real-time through remote monitoring, thereby protecting a user from possible health-related risks. The accurate detection of user-specific anomalies, prediction of future events and real-time decision support require intelligent analysis on large biomedical data gathered from many AAL users. This necessitates the need of a scalable system with large storage and high processing capability to support multiple AAL systems simultaneously with individualized context-aware services. The work herein seeks to address such issues and develop a cloud-based context-aware system for AAL that also solves problems regarding clinical abnormalities detections and predictions by discovering personalized knowledge through using context correlations and different data mining techniques.

Chronically ill patients die of various diseases from the lack of an efficient automated system having predictive ability. Traditional healthcare solutions are limited to some specific services. While learning models have been developed, they are based on generalized observations and suffer high false alert rates when uncertainties in data increase. Considering these drawbacks this research realizes the need for a self-care, predictive and protective assisted living system where a patient’s personalized knowledge is developed by learning from large amounts of historical and contextual data.

In an AAL system, a patient is monitored using wearable sensors. These biomedical sensors lack the processing power to perform monitoring, analysis and data-aggregation tasks, necessitating data transmission and computation at central locations. The resource-constrained nature of typical wearable sensors is factored into this research, with cloud computing features
utilized to provide a real-time service. Moreover, this study focuses on the development of learning techniques to transform large biomedical data into useful knowledge using a cloud-based framework. This research is intended to explore the machine-learning methods that suit large scale data analyses and exhibit high accuracy and efficiency.

We begin by designing a cloud-oriented context-aware middleware ("CoCaMAAL") which has the ability to support patients of multiple AAL systems in parallel and independently. Here we introduce how different tasks such as data processing, learning, and context aggregation can be distributed in multiple cloud components by efficiently utilizing cloud resources. We then extend the capability of the model to capture individualized knowledge from big data. We realize that context-aware data of AAL systems satisfy the characteristics of big data (i.e. volume, variety and velocity) and develop a learning model using MapReduce Apriori to discover personalized knowledge by relying on historical big data and context correlations of a patient. The model is further used to find patient-specific abnormalities in the current context, acknowledging that the level of abnormality varies among individuals and in different situations. However, this model lacks the capability of predicting future changes. Therefore in the next step, we build pattern recognition models for predicting health-related changes and behavioural trends in a patient. The proposed techniques are evaluated for different patient scenarios.

Moving beyond the feature of future abnormality estimation, we limited our focus only to some clinical event predictions based on the correlations of multiple vital signs (e.g. heart rate, blood pressure) considering the context as an interrelation among vital signs. We target to solve two types of problem: firstly, prediction of patient’s future clinical behaviour using only his/her own data, and secondly, prediction of different clinical events of a patient using available known knowledge involving many similar patients. We aim to build these learning models with continuous prediction capability. We have used patient data from publicly available database. All the experiments are conducted using cloud platforms.

The purpose of the first problem is to predict the clinical behaviour of a patient in the near future by continuously developing the knowledge from that individual’s recent past data. This problem is addressed by developing a learning engine using multi-label classification. The model is patient-specific, adaptive and continuous; it can achieve multiple targets simultaneously. To resolve the second problem, given that a large number of labelled samples from many patients are available beforehand, we developed a probabilistic model using Hidden Markov Model and a static predictor model using data mining algorithms. The probabilistic model predicts approaching clinical events of a patient using temporal behaviour of six bio-signals. This
model does not have any forecast margin and uses a few samples for model development. The static predictor model uses a large number of samples for training, varieties of features (e.g. wavelets, correlation coefficients and short-term statistics) and a long forecast gap. However, the prior knowledge about similar clinical conditions may not be always available. Therefore, we developed a new patient clustering method to identify similar patients based on the correlations in clinical data. We developed an unsupervised patient-specific pattern discovery method where multiple vital signs are characterized to a number of dynamic states using entropy criteria. A similarity analysis is then performed over features extracted from dynamic states of multiple patients using hierarchical clustering which can estimate similar patient subgroups. The experimental results indicate that the proposed algorithm can identify different patient subgroups with high accuracy.

In a nutshell, this thesis contains problem-specific efficient learning algorithms that can be employed for context-aware sensing in an AAL system by utilizing cloud platforms. This research mostly utilized existing data mining models to solve problems on abnormality detection, prediction and knowledge discovery. The solutions addressed the scalability issues and can work on big biomedical data effectively and accurately. Therefore, the research contributions in this thesis present a scalable model to provide versatile and reliable context-aware services using proper machine-learning models. We believe that this research is a big step towards building a generic model for the AAL community and the results can inspire and be used for the development of efficient learning models for home-based monitoring.
Chapter 1

Introduction

Together with an explosive growth of body sensors [1, 2], smart devices, mobile applications, and the emergence of the cloud computing technologies [3, 4], context-aware systems [5, 6, 7] for ambient assisted living (AAL) [8, 9, 10] have been introduced as providing potential technology for remote healthcare [11, 12]. Context-aware monitoring [13] is the key to an AAL system, as it should comprehend the situational context of the collected data and provide real-time personalized healthcare services tailored to meet user needs [14]. Moreover, it is a rich application area of modern big data analytics [15, 16, 17].

Prediction of future abnormalities using long-term medical data of a patient through context-aware monitoring is also becoming a prominent technology to detect progressive changes in human health and behavioural patterns [18, 19]. A failure to detect symptoms early can cause severe disease or even death. Strong correlations exist among different contexts and biomedical data gathered from a patient of an AAL system. By utilizing such individualized correlations and data mining techniques [20, 21], it is possible to learn the behavioural, health-related and clinical changes of a patient which further used to predict future abnormalities. The early detection of anomalous situations can avoid potential health-related risks for a patient and simplify the jobs of healthcare professionals by reducing unnecessary false alerts to the monitoring systems.

The physiological data of a patient varies with different activities and medical conditions. The dynamism in readings, collected over varying periods of time, results in massive context-spaces for AAL systems. Applications need a complete knowledge and must remain context-sensitive in order to satisfy different behaviour profiles based on an individual’s specialized needs. The amount of context that can be gathered from a personalized assisted living system
Background and Motivations

is so massive that it is almost impossible to store and manipulate them in a local server or mobile device. Furthermore, the growing number of chronic diseases and rise in elderly population is increasing the demand of a common platform which is capable of handling these big data challenges. The need for a large computing space, and ready availability of cloud services, has led us to envision and develop a cloud-based, predictive and context-aware system.

1.1 Background and Motivations

Throughout the world, aging populations and higher disability rates have intensified the pressure on already burdened healthcare infrastructures. Coupled with higher occurrences of chronic lifelong diseases such as cardiovascular-related illness and diabetes, this phenomenon has resulted in urgent interest in AAL research. Chronic diseases [22] are major illnesses especially in older adults, and is increasingly becoming a part of life. These diseases are the main cause of many deaths in Australia and other western countries. The main reasons for chronic illness include are: lifestyle, improper diet, and congenital genetic problems [23]. According to US Centers of Disease Control (CDC) [24] as of 2012, 117 million people (about half of the adults) have one or more chronic health conditions. Seven of top 10 causes of death in 2010 were chronic illness [25]. Only heart disease and cancer together accounted for nearly 48% of all deaths [24]. Moreover irregular lifestyle and diet cause about 78 million people obese in 2010. Many people are also suffering diabetes which is a leading cause of kidney failure and blindness. The expenditure for treating these diseases in hospital is also very high. In 2010, the total costs of heart disease and stroke were estimated to be $315.4 billion and of this amount $193.4 billion was for direct medical expenses, not including costs of hospital [24].

The challenge of providing comprehensive care to an aging population is also a global concern where we are now seeing the effect of declining fertility rates combined with the aging and the retirement of the baby boomer generation. In 1901, the average life expectancy in Australia was 47 years (Inter-generational Report 2010). By 2025 it will be over 80 years. At present 2.8 million Australians (13% of the population) are aged over 65 years and this is predicted to almost triple in the next 30 years (6.4 million) which will represent 30% of the total population. With the rapid population aging that is currently occurring throughout Australia and across the world, the need for aged and health care services will increase. Given that Australia has comparatively small population of 22 million, in near future there will be huge lack of human and institutional resource to support this large population. Another issue is the economic cost to strained healthcare systems, with increasing numbers of people requiring
extended stays in hospitals and other nursing facilities. This places a burden on the government and more importantly, the families of the patients, who must bear the emotional and economic cost. Therefore, smart technology-based healthcare services are needed to be adopted.

Traditional hospital-focused model of care neglects monitoring and treating diseases at home that could prevent and minimize risk. Unfortunately, recent reports indicate that hospitals are getting overcrowded and are having difficulties in treating the patient even in emergency situation due to increasing population. A recent news shows that, patients (including elderly people aged over 80) in Western Sydney hospitals waited more than two days in the emergency department and had to leave without treatment [26]. According to that report, there was up to 40 hours waiting time on average to be admitted to the hospital. The capacity in hospitals is inadequate to make place for all patients. Moreover, if chronically ill patients wait long for their next hospital visit, which is eventually a costly procedure, then a constant threat of uncertainty can make the chronic conditions even intense. The situation will become worse for the patients as well as for the healthcare providers unless a personalized, scalable, real-time, prognostic, context-aware system is adopted. If patients are monitored real-time in home with early prediction capability, then they would have peace of mind and a higher chance of avoiding next chronic episode. Thus, a proactive prevention system is the challenge of modern healthcare technology.

An AAL system consists of heterogeneous sensors and devices which generate huge amounts of patient-specific unstructured raw data everyday [27, 28]. Due to diversity of sensors and devices, the captured data also have wide variations. A data element can be a few bytes of numerical value (e.g. HR = 72 bpm) to several gigabytes of video stream [29]. 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 analyse only continuous ECG of cardiac patients in real-time inside the cloud environment [30, 31], then it will produce around 7 PetaBytes data everyday from 3.5 million patients \(^2\). The advance in wearable sensors technology has also made it possible to monitor multiple vital signs of a patient anytime, anywhere. When multiple vital signs from large number of patients are accumulated they evolve also into big data, which is an essential part of daily monitoring and disease prevention. Vital signs such as heart rate (HR), blood pressure (BP),

\[^1\]100 \times 24 \times 60 \times 60 \times 365 \times 5 \times 10^6 = 14 \text{ Exabytes}

\[^2\]If we consider a 12-lead ECG signal from a single patient that have 1024 samples per second with 16 bits resolution, then the amount of data in a day will be, \(12 \times 1024 \times 16 \times 24 \times 60 \times 60 = 1.98 \text{ GB}\). So, for 3.5 million cardiac patients the daily amount of data will be, \(3.5 \times 10^6 \times 2.1 \text{ GigaBytes} = \text{nearly 7 PetaBytes}\).
respiratory rate (RR), oxygen saturation (SPO$_{2}$) and body temperature are crucial part of big medical data [33]. If we assume the numerical value of each vital sign contains 4 bytes and the frequency of data collection is 1 minute, then for 6 vital signs total 24 bytes data gathered per minute, which is equivalent to 33.75 KB per day, or 12 MB per year $^3$. If such data are gathered from 5 million patients, then the data amount will be 57.3 PB per year $^4$. This statistic is only for 6 vital signs data. Including these dynamically generated continuous monitoring data, there are also huge amount of persistent data such as patient profile, medical records and disease histories. If we want to store all these data and patient histories to predict any future abnormality accurately, then the representation of data will be in zetabytes in a few years. Such concerns motivated us to the development of a cloud-based assisted healthcare infrastructure [34].

Context-aware AAL system has gained significance in recent years, combining aspects of intelligent platform design, assisted living solutions and ambient intelligence technologies [35] into a coherent system. This drives numerous research efforts dealing with biomedical data analysis, health-related abnormality detection, future clinical event predictions and patient clustering methods. The idea here is to build a personalized AAL system that features situational awareness and real-time decision-making capabilities [36], with adequate flexibility at the architectural, algorithmic, and human-interface level. To maintain quality healthcare services [37], it is essential to have an intelligent, highly resourced AAL system that is efficient, responsive, and most importantly, adequately secures patient health. Therefore, our main objective of this study is to develop a scalable context-aware system for assisted living with intelligent decision support capability that satisfies the strategic objective of global research.

1.2 Preliminaries

Before proceeding to the technical details of our analysis, this section affords the reader a basic familiarity with underlying concepts, terminologies, building blocks and workflow of our research.

Ambient Assisted Living (AAL) is a technology-based approach from information and communication technology (ICT) to support patient or elderly people at home. The goal of AAL is to secure the health of its users so they can live independent in their own home. It simplifies the activities of daily living with home automation and monitor and care for the user with

$^324 \times 60 \times 24 \times 365 = 12$ Megabytes

$^412.03 \times 5 \times 10^6 = 57.36$ Petabytes
intelligent solutions. Assisted living technologies available at present age range from very simple medication reminder to high-tech solutions such as remote monitoring system with early prediction capability. This technology is economical and saves medical resources.

In an AAL system, a typical architecture involves body sensor network (BSN), ambient and smart sensors, wireless networks, ubiquitous devices, actuators and software services that collect data from a target user who lives alone and has some kind of disability (e.g. elderly person or patient) [27]. Wearable and implantable data collection devices [27] (e.g. Shimmer), available in market at a very low cost, can provide real-time physical and medical information of the user. These sensors have some key configurations and infrastructure that make them easily implantable or wearable on the human body (as shown in Figure 1.1). Some sensors can be implanted in the garment of the user; these are known as wearable textile sensors. These sensors have low power, are capable of communicating wirelessly, and monitor the health status (e.g. vital signs, ECG) and activity of the target user. Likewise, knowledge pertaining to the environment is obtained from environmental devices, ambient sensors, smart sensors, and other communication devices (as shown in Figure 1.2) that focus on ambient readings (e.g. temperature, humidity). Along with a local processing point where data are initially transmitted (e.g. a workstation, smart phone or tablet), all these sensors and devices form the overall data-collection components of an AAL system.

Figure 1.1: (a) A demonstration of wearable sensors on a human body. (b) A conceptual BSN architecture of the proposed AAL system. (c) An example of wearable textile sensors.
The data of an AAL system are collected continuously during different times of the day and also in some cases on a demand basis. The data can be physiological (e.g. heart rate, blood pressure, ECG), environmental conditions (e.g. humidity, room temperature) that affects a patient’s health condition and activities related to a patient’s behaviour (e.g. sleeping, eating, toileting) [38] that can be inferred by processing data from sensors, cameras, RFIDs etc. Some data can be captured from persistence storage such as a patient profile (e.g. the patient has heart disease), recognized patterns (e.g. patient wakes up between 7-7:30 a.m., smokes 5 times per day on average), historical (e.g. patient had a heart attack 2 years ago) and medical records (e.g. last tested white blood cell count 7746).

The terminology ‘context’ varies according to the purpose of application domain. Normally context is any information that can be used to characterize the situation of an entity [39]. The concept of context-awareness is widely used in mobile and ubiquitous computing to infer the user’s objective and relevant environmental feature. In our research, context means any high level user-specific information obtained directly or inferred from raw sensor data (as shown in Figure 1.3).
Generally, from AAL systems the aggregated contexts are sent to a remote monitoring centre (e.g. doctor, nurse, hospital) for decision-making about patient’s condition. In our vision, we take this one step further by incorporating patient-specific intelligence that constantly learns from collected data and interprets new incoming data using that gained knowledge just as a medical expert would. This also allows doctors to make decisions with greater knowledge, to monitor chronic deterioration in a patient’s condition, or to assess the patient’s response to a treatment. The prior identification of a patient’s abnormal condition can warn the patient by activating a local device (e.g. medication reminder), or send an emergency message to the monitoring centre.

In modern remote healthcare monitoring, the data from different biological signals of a patient are obtained continuously using wearable sensors and analysed in the cloud. The wireless communication (bluetooth, wifi, zigbee) ability of wearable sensors simplify the data transmission process to the cloud repositories via a mobile device having a high speed internet connection. The cloud has large distributed storage and high processing capability. The distributed resources of the cloud simplify the knowledge build up process from large biomedical data using computationally intensive machine learning methods [20]. The flexibility of using low-cost cloud platforms such as Amazon web service (AWS) [40], Windows Azure [41] and
Google cloud [42] has added greater advantage to learn abnormal situations and future events of a patient by storing and analysing every past and current events. For example, by applying data mining techniques inside the cloud environment over continuous batches of collected data of multiple vital signs it is possible to induce logical models which can infer the future values of those parameters. The predicted values of these vital signs and their interactions are used to advise a patient about the medical consequences or to recommend patient’s doctors for subsequent clinical decisions. This is also known as context-aware actions by the system. Such a workflow is presented in Figure 1.4.

Figure 1.4: The workflow of real-time monitoring, classification, predictions and clinical decision support.

Overall, this research opens up a new branch of knowledge by combining the concept of some key technologies such as assisted living, context-awareness, cloud computing, big data, remote health monitoring, knowledge discovery, data mining, decision support system, anomaly detection and clinical event predictions. We believe that, this work will encourage healthcare service providers to use cloud-based solutions.

1.3 Research Challenges

We aim to design a model where data analysis and context processing related tasks can be distributed in different cloud components. The model should have capability to support patients of multiple AAL systems in parallel and independently. With the myriad of distributed AAL systems at play, each with unique requirements and eccentricities, the challenge lies in the need to service these disparate systems with a middleware layer that is both coherent and flexible. There is significant complexity in the management of sensor data and the derivation of contextual information, as well as in the monitoring of user health and activities and in locating appropriate situational services.
AAL systems typically consist of multitudes of sensors that generate large amounts of contexts in terms of physiological data, activities, locations, environmental conditions, disease histories etc. Due to the diversity in such low-level sensors and patient’s situations, the range of data observed from AAL systems can vary widely. Accordingly, the conversion of the data to structured interchangeable formats and subsequent aggregation and management represents a significant challenge. Successful implementations rely not only on solid hardware and software foundations, but also on remote computing facilities [43]. Thus, the adoption of cloud computing paradigms will play a vital role here. Efficient processing of large volume of data using computational power of cloud infrastructure, extraction of right context information [44], finding the correlations among different contexts for inferring knowledge, and prediction of a state using those inferred observations to deliver proper situation-aware services, are some primary challenges in the development of context-aware monitoring applications.

An essential context-aware service of an AAL system is to identify the abnormal conditions of a patient accurately and to send appropriate alerts to the care givers. The system should have capability of distinguishing abnormal situations from normal ones. Proper context affords a keener understanding of the state of patients, places, or objects and how they relate to the interaction between users and the system. A patient’s physiological data vary with different activities (e.g., pulse during resting versus exercising) and indeed at different locations (e.g., body heat when running outside vs running on a treadmill at home). The data also changes as the patient ages and quite obviously, varies with the individual. Therefore, it is an extremely important issue to capture such individualized nature using proper learning scheme.

Prediction of future abnormalities is another crucial feature in the field of healthcare monitoring [45, 46] and smart-home technology [47]. The availability of a system capable of automatically detecting future behaviour or abnormality is remarkably attractive for many applications. The long-term historical information of human physical activities [48], daily behaviours and physiological data are valuable for long-term assessment of behaviour and health. Failing to detect symptoms early can result in severe disease or even death. The early anticipation of anomalies also can improve the rate of disease prevention. Existing solutions only detect abnormalities in a current context but ignore prior prognosis to alert patients about future risks. Considering huge diversity in context information, a major challenge to such context-aware technology is to convert this huge amount of information into a knowledge base using appropriate machine learning models so that the system becomes competent enough to accurately sense eventual change.

In remote health monitoring systems a large number of patients are continuously monitored.
It is not possible to manually analyse data of every patient. Moreover, the amount of data that is accumulated per patient is enormous and growing. Such data are not only massive in size but also heterogeneous in nature due to the clinical essence and medical history of the patient. In such a hostile situation, it is extremely difficult for clinicians to estimate the likelihood of a particular clinical event with bare-eye analysis even with sufficient amount of time. Some health data such as multi-parameter clinical time series contain interesting patterns for various clinical events. To ensure correct diagnosis the proper characterization of such patterns is extremely vital.

The change of patient’s context is also observed only within vital signs [49]. The trends in different vital parameters are highly correlated and patient-specific. For example, the mean blood pressure (MBP) value is always high for a hypertensive patient (i.e. patient with high blood pressure). The MBP value does not contain any anomaly for this patient as a single vital sign unless it raises above a specific threshold. For a normal patient, the threshold of abnormality is different i.e. the same MBP value can be abnormal for a normal patient but deemed as normal for a hypertensive patient. Thus for many health parameters, normal is actually a relative value for the patient. Many factors can affect changes in one or more vital parameters such as age, patient’s activity, medications, stress and surrounding conditions such as environmental temperature. The progressions of many diseases are reason of abnormal changes in multiple vitals. Certain patterns of irregularity develop through a strong correlations among various vital signs. Therefore, each vital sign and related interactions must be interpreted in the context of a patient. Hence, this necessitates the demand of a personalized model for clinical decision support that can independently discover patient-specific anomalies by employing a common learning technique.

Identifying similar patients from the patterns of their historical data is another crucial challenge of remote health monitoring. Grouping similar patients in a single category can enable better treatment and more efficient care. Moreover, an unknown patient can be categorized easily based on the prior knowledge about the representatives of similar cases. Generally, doctors use their years of experience and knowledge to find identical patterns in patients. However, such process is extremely difficult and costly for a large number of patients. Thus, a long standing goal in the area of health informatics is to automate this process by developing an unsupervised technique of similar pattern discovery in patients.

Therefore, considering the above challenges in context-aware system for assisted healthcare we can summarize the following core challenges.
• Summarization of large data from heterogeneous sources to a meaningful knowledge by utilizing scalable resources.

• Reliable and patient-specific context-aware services depend on proper personalized knowledge discovery from long-term histories.

• Protecting patients from future risks require the capability of detecting long-term changes and future abnormalities.

• The correlations among clinical data are important for appropriate estimations of abnormalities and future values.

• The continuous context-aware support with quality services demands an adaptive and real-time system.

• The complexity of gathering similar patient information and lack of availability requiring an unsupervised patient clustering process.

1.4 Research Questions

The main research objective of this thesis is to develop a scalable context-aware model for AAL that also solves problems regarding clinical abnormalities detections and predictions from large biomedical data by discovering user-specific knowledge using context correlations and different machine-learning techniques. To achieve this goal according to the background study and motivations the main research problems of this thesis are summarized in Figure 1.5.

In order to overcome the aforementioned research challenges the following research questions are defined with the aim of achieving a scalable, predictive and protective context-aware system for AAL.

The model

RQ-1. How to support large number of patients by user-specific context-aware services using a common model? How to manage, process and convert large biomedical data from heterogeneous sources of many patients to useful knowledge using that model?

This question addresses the issue of scalability by developing a cloud-oriented distributed
Figure 1.5: A scalable, predictive and protective context-aware system and related research problems.

The middleware. The significance of addressing this issue is primarily to ensure that we can support a large number of patients in parallel through context-aware services by using a common framework.

Knowledge discovery, abnormality and change detection

**RQ-2.** How to detect health-related abnormalities in a current situation and long-term behavioural changes in a patient by discovering personalized knowledge and correlating the changes in different contextual information?

From the solution of the *first research question*, we have a framework to process big data generated from patients of many assisted living systems. In the *second research question* we have utilized this framework to develop learning models for personalized knowledge discovery
and, using that knowledge, detect physiological abnormalities and future changes in a patient. The interrelations between various contexts are utilized for these models’ development. The solutions ensure that abnormalities and changes are detected accurately and patients can be supported by user-specific context-aware services.

**Future behaviour prediction**

**RQ-3.** How to predict patient-specific future abnormalities continuously by utilizing the correlations of his/her present and past biomedical data as the context for providing intelligent decision support?

This research question addresses the issues regarding prediction of future states. The same model that is developed as a solution to the first research question is utilized to learn from patient-specific past data for predicting future values. This is a ‘one-to-one’ solution where the system learned from contextual data of a patient and predicts only for him/her using that learned information. This continuous learning solution enables continuous decision support for a patient.

**Patient similarity**

**RQ-4.** How to predict clinical abnormalities and future behaviours of an unknown patient using the prior knowledge from a large group of similar patients? How to identify those similar patients?

Using the cloud-based model developed for Research Question 1, this research question addresses the issues regarding similarity in patients’ context. This provides a ‘many-to-one’ solution where learning models are developed from the past data of many patients which are then used for real-time prediction for a new patient. This research question also solves the issue relating to clustering similar patients. The solutions provide an unsupervised approach in identifying similar patients and early prediction capability of abnormal clinical situations of new patients using prior knowledge of those similar patients.
1.5 Limitations of Existing Solution

A wide range of context-aware middleware solutions exist for AAL technology where research focus is restricted to specific services such as activity monitoring [50, 51, 52], fall detection [53], caregiver management [54, 55, 56] and emergency monitoring [57]. Most of these systems proposed different context modelling approach according to their problem domains. No precise standard is found in literature for context modelling specially for home-based assistive care.

Existing systems for personalized AAL systems depend on standalone applications that run on a local server or a mobile device. These applications are developed to provide limited services [58]. However, the volume of data that is gathered from each personalized AAL system is so large that it is impossible to store and process them for knowledge-discovery in small memory of a hand-held device. Moreover, the physiological data of a patient varies with different activities, locations, medical histories and for many other reasons. Therefore, performing resource-hungry operations such as context data processing and learning on this big data with large diversity from centralized middleware servers would result in rigid, failure-prone, and non-scalable systems.

Conventional context management solutions are incapable of handling good number of AAL systems in parallel. Most of the system target to serve a specific patient category (e.g. cardiac disease) [18]. As they are based on centralized hospital server, it is impossible to serve a large number of patients with various chronic diseases using such centralized management facilities. There is a huge lack in the coordination between patients and healthcare professionals when the number of patients increases.

In traditional systems, patient situations are classified by generalized medical rules [59] or fuzzy rules [60] which are not always applicable for every kind of patient. These systems cannot sense the future at an early stage. In some monitoring systems, when a patient feels unwell he/she needs to press a wearable panic button to notify a response centre about the emergency. Some systems try to understand patient’s discomfort level and the seriousness of the condition by asking automated sets of questions [61]. Such systems can generate many false alarms to the monitoring centres; which is not desirable.

Existing home-based context-aware remote monitoring systems use different techniques such as rule-based reasoning [13, 62, 63], probabilistic model [64], data mining [65, 66], sensor-network based decision support [67] and video-based monitoring [68] which can detect abnormalities only when they occur in an ongoing situation. There is no option to store and reuse context histories which can be utilized for modelling each user state individually and predict
future anomalies ahead of time. That is, most of the system are not capable of accurately
discovering personalized knowledge for the shortage of preserving long-term histories. These
systems have shortcomings of intelligent detection of disease symptoms and can not provide
accurate recommendations to the users of AAL systems.

In case of future change detection, much research has been done in the areas of understand-
ing human behaviour for daily activities [45, 50, 69, 70], detecting anomalies in daily
behaviour, disease prediction etc. But little work is found in the literature that focuses on
the development of an integrated and intelligent change detecting model that combine the
information from multiple context sources.

A number of intelligent monitoring systems exist for clinical abnormalities prediction for
the patients who are confined to hospital beds [60, 71, 72], but a very few attempts [73, 74] have
been made to develop a predictive system for home that could prevent and minimize health-
related risk at early stage. The monitoring systems those exist suffer from high false alert rates
and depend on manual observations of clinical data by the medical experts after an anomaly
is detected. Some of the developed models can forecast the changes in a specific physiological
parameter with good accuracy [19, 75], however, it remains a challenge to build these models
to monitor and correlate multiple bio-signals while maintaining the interpretability of the
discovered knowledge because biomedical data can evolve and vary over the lifetime of the
system.

There have been also limited research that attempts to forecast various clinical events
using multi-parameter data of a large number of patients [76]. Most of the prior studies use a
small sample of data (i.e. few megabytes) from a small group of patients, short-length forecast
window (mostly an hour) and work only on a single parameter such as blood pressure [73], ECG
[77] or PPG. Those models utilize small number of features for training and only predict the
future behaviours of some special clinical events [19, 78]. Such systems experience higher rate
of misclassification when uncertainty of data goes high with the increase in patient populations.

Considering these limitations, this thesis focuses on the development of cloud-based solu-
tion to provide reliable context-aware services to the users of ambient assisted living systems.
The analysis in this work improves the recognition and prediction ability of abnormalities by
considering large scale patient histories. In addition, this work utilizes personalized knowledge
to improve the accuracy of recognition.
1.6 Research Contributions

To address the research questions defined previously and to overcome the limitations discussed in last section, extensive studies have been performed using cloud platforms and different data mining techniques to verify our proposed solutions. A summary of the organization of the novel contributions corresponding to the aforementioned research questions is presented in Figure 1.6.

![Figure 1.6: A summary of research questions, thesis contributions and chapter organizations.](image)

In general, the contributions of this research are summarized as follows.

1. **Analysis and development of context-aware middleware**

   A distributed and scalable architecture is an important consideration, especially when we aim to build a model that can provide context-aware support patients with various chronic conditions. In relation to the first research question, middleware is developed
utilizing cloud platforms that is capable of storing and processing heterogeneous data from a large number of patients. Moreover, a unified context modelling approach is proposed to aggregate data from different sensors of an AAL system as a single meaningful context. Context-awareness in AAL systems is achieved by disseminating different responsibilities to distributed cloud components. The major computational tasks, such as raw data to high-level context conversion, context aggregation and context-aware service formation are separated and assigned to different cloud processing servers. The model also minimizes the communication gap between a patient in an AAL system and a doctor in the remote service provider. The framework can support a large number of patients simultaneously and provide patient-specific real-time context-aware services. All the past contexts of every AAL system are stored in cloud repositories. This enables faster learning with greater knowledge. The solution emphasises the concept of service-oriented architecture [79] that performs context modelling from raw data, context data management and manipulation, context-aware service mapping, service distribution and service discovery. The model is prototyped and tested by implementing some user scenarios to detect abnormalities using generalized medical rules in current situational information and to identify appropriate context-aware services. The performance of the model is evaluated using M/G/1 queuing model [80] and good service recognition accuracy is obtained in a short response time. Overall, this innovative model enables the recognition and delivery of reliable context-aware services within a short time and ensures qualitative remote monitoring support using the advantage of cloud computing. Chapter 2 contains the detailed description of this middleware and its main contributions correspond to [81].

2. Learning models for personalized knowledge discovery, abnormality and change detection

The biomedical data and other contextual information obtained from a patient in an AAL system are very specific to that particular patient and may not work accurately for detecting abnormalities in other patients. Moreover, the system cannot distinguish between normal and abnormal situations properly if information from multiple context sources is correlated. In such cases, the system suffers from misclassification of normal situations and creates false alerts. To overcome these challenges with regard to the second research question, different learning models are developed that can work on a distributed cloud model. The major contributions regarding this are described below.

- As our system learns from large historical data, we have used the MapReduce pro-
gramming model \cite{82} that can efficiently learn from big data \cite{15} in a distributed cloud environment. To discover personalized rules \cite{83} for detecting user-specific anomalies, MapReduce Apriori algorithm \cite{84} is applied over large context histories of multiple patients in parallel. The algorithm generates a set of association rules using the trends and patterns in the data of an individual patient which become a personalized knowledge for that patient. The rules act as high-level information of patient-specific correlations. Such an individualized rule-mining process also promotes the inference of more generalized medical rules for a particular patient category. In the next step, data mining techniques \cite{20} are used to obtain a personalized classifier model from the outcomes generated from personalized rules. The classifiers are then used for patient-specific situation classification and context-aware decision-making. For this model, higher recognition accuracies are obtained compared to general rule-based model. The proposed learning model can make a potential contribution in the area of big data analytics for healthcare. The model, described in Chapter 3, is named as BDCaM and its main contributions correspond to \cite{85}.

- Long-term historical data contain useful information to detect progressive changes in human health and behavioural patterns. We have developed pattern recognition models to detect any physiological and behavioural related changes in patients. The interrelations of those changes are used to predict future behaviours, physiological states, clinical situations and disease symptoms of the patient. The developed change detection framework works together with our generalized context-aware framework for all AAL users and is based on multiple recognition models. Firstly, a Hidden Markov Model (HMM) based approach \cite{86} is implemented for detecting abnormalities in daily activities. Secondly, a process is implemented to identify irregularities in routine behaviours from statistical histories. Thirdly, an exponential smoothing technique \cite{87} is developed to predict future trends in various vital signs. Finally, the outcomes of these three models are used using a fuzzy rule-based model \cite{60} for making the final prediction for future behaviours. Good prediction accuracies are found in comparison with other related context-aware models. Using this novel technique, it is possible to predict any future abnormality or change in a patient’s health in various situations. This system simplifies the tasks of healthcare professionals by assessing the causes of any anomalous situation at an early stage.
This solution thus can make a valuable contribution to reduce chronic disease-related deaths and to lessen hospitalization. All these are described in Chapter 4 and the main contributions correspond to [88].

3. **Learning model for patient-specific clinical event predictions**

To identify different clinical episodes, doctors always look for abnormalities in multiple health parameters. The symptomatic patients are likely to have several abnormal vital signs. That is, to assess the actual abnormality for a single vital sign, the context of other vital signs should be taken into consideration. This observation is used in addressing the *third research question* and a completely new approach of patient-specific vital sign prediction system is developed using the correlations among multiple vital signs. A multi-label classification-based [89] prediction model is implemented that can detect the future trends in multiple vital signs at the same time using their correlated features. It is possible to make better predictions if the changes of multiple vital signs are considered as a whole rather estimating them individually. A multi-label classification is extensively applied to detect such correlations and predict the future values of vital signs in advance. The model is adaptive and capable of making continuous predictions. Multi-label classification evaluation measures such as hamming score, accuracy, exact match and F1 micro average are used for interpreting prediction performances of patient-specific situation classifications. The results are compared with single-label classification algorithms and without considering the correlations among the vital signs. The evaluation results reveal that a multi-label classification process using the correlated features of the vital signs is an effective technique for early prognosis of vital sign values. Moreover, by utilizing our cloud-based model it is possible to build such multi-label classifiers for many patients concurrently. The developed technique, described in Chapter 5, is a useful contribution to clinical decision support as it can guide healthcare professionals in diagnostic decision-making.

4. **Analysis on patient similarity**

In our design consideration we assume that patients’ clinical data are populated continuously in cloud storage. Once we have sufficient data we can build prior knowledge from similar patient groups to develop a more accurate prediction model for new patients. Sometimes data are labelled with the clinical classes of the patients which can easily be utilized to build prediction models using supervised learning. The models can be used to
Research Contributions

decide about clinical situations of an unknown patient. However, if data is not labelled then we need unsupervised learning to label them first. Based on these investigations the main contributions to the fourth research question are illustrated below.

- A probabilistic prediction model is developed that can determine the clinical nature of an unknown patient using current and past data of multiple vital signs of similar patients. Here HMM [86] is used to design the learning engine as this model perfectly suits this kind of problem. This system is expandable, that is, a new clinical case can be added to the classifier with a new HMM training. This model is also adaptive, because HMM parameters can be updated with a new set of observations that maximize its probability. The model also supports real-time classification where the continuous vital sign data of an unknown patient are sent to the cloud and the probability of the occurrence of a clinical event in future is continuously computed. The developed model is compared with other models in existing literature and some impressive features and good predictions accuracies are obtained. The contribution of this method, illustrated in Chapter 6, provides a helpful tool to estimate future states of a patient with unknown clinical nature. The main contributions of this method is also correspond to [90].

- A Patient’s ‘clinical class’ prediction system using multiple vital sign data of a large number of patients is developed. This is a static predictor model where the prediction goal is accomplished using different features obtained by applying discrete wavelet transform and by computing short-term statistics and correlation coefficients of six bio-signals. This analysis uses a very large collection of data and long forecast gaps (1 to 2 hours) to enhance prediction ability of different clinical events that occur when there are changes in multiple vital signs. Four different clinical classes are identified from 4,893 patient records where multiple vital signs deviate from normality and features are extracted from 10-30 minutes observed data prior to the forecast gap for each clinical class. Different data mining algorithms such as J48 decision tree (J48), random forest (RF), and sequential minimal optimization (SMO) [20] are applied and prediction performance are evaluated. The best model is selected based on accuracy and model building time. The contribution of this result is also discussed in Chapter 6 and can be used to build an effective patient management tool for doctors.

- A new patient clustering algorithm is developed to identify patients with similar
Thesis Organization

Clinical nature. This is a two-step clustering process. In the first step, different physiological states of a patient are measured dynamically using entropy measure. Multi-dimensional physiological data are transformed to single dimension by calculating aggregated instance-wise entropy values (AIE) of each observation. Dynamic partitioning is then applied based on a minimum AIE value to discover the dynamics states of each patient. In the second step, similarity analysis is performed among dynamic states of patients using distance correlation and, finally, hierarchical clustering is used to separate patient subgroups based on that similarity. The contributions of this method, described in Chapter 7, can help doctors to identify patients with similar problems. Thus, they can make a common treatment and monitoring plan for patients.

1.7 Thesis Organization

The remaining chapters in this thesis are organized as follows.

- **Chapter 2: A Cloud-oriented Context-aware Middleware for Assisted Living.** This chapter describes the solution to Research Question 1. Here we introduce a distributed cloud-based middleware for our system and related functionalities addressing the limitations of existing context-aware middlewares. We named this model “CoCaMAAL”. A detailed evaluation of the proposed middleware is presented by implementing some case studies. The main contributions to this chapter corresponds to [81].

- **Chapter 3: A Personalized Knowledge Discovery Framework.** In relation to the solution of Research Question 2, an extended version of CoCaMAAL is presented; this is called “BDCaM” (Big Data for Context-aware Monitoring). The development of learning methods using MapReduce Apriori for personalized knowledge discovery using big data is the primary focus of this chapter. A use case is implemented to illustrate the applicability of the approach that discovers the knowledge to identify the true abnormal conditions of patients having variations in heart rate and blood pressure. The primary contributions of this chapter also appear in [85].

- **Chapter 4: Long-term Change Detection and Abnormality Prediction.** A change detection framework for future abnormality identification and behaviour prediction is described in this chapter; it wraps up the solution to Research Question 2. Case
specific pattern recognition models are developed and evaluated for different patient scenarios. The major contributions of this chapter correspond to [88].

- **Chapter 5: Patient-specific Future Clinical Event Prediction.** A clinical decision support system (CDSS) is outlined in this chapter in relation to Research Question 3. It describes the process of patient-specific clinical event prediction using multi-label classification techniques. The necessity of multi-label classifier is justified through experimental evaluations and a comparative study.

- **Chapter 6: Early Prediction of Abnormal Clinical Events Using Known Knowledge.** With regard to the solution to the first part of Research Question 4, two learning models are presented in this chapter. One is a probabilistic model that estimates the future states of an unknown patient by using the classifier developed from the observations of other patients. Another is a static model that finds clinical events of an unknown patient after some forecast gap by utilizing a learning model developed from the observations of a large number of patients. The accuracy of predictions for both models are evaluated experimentally. A part of contributions of this chapter also appears in [90].

- **Chapter 7: Patient Clustering by Dynamically Partitioning Patient States.** An unsupervised technique of a patient’s clinical class identification is described in this chapter which also completes the solution to Research Question 4. A ‘dynamic pattern discovery’ using multiple vital signs in a patient is another major focus of this chapter. The clustering technique is experimentally evaluated using two labelled datasets.

- **Chapter 8: Conclusion.** This chapter concludes the thesis by summarizing the main contributions and key findings. In addition, the significance of this research and potential future directions along with some limitations of the developed methods are discussed.

**Note** Six core chapters (Chapter 2-7) are presented in a standalone and self-explanatory manner. Therefore, the relevant context including discussions on related work, description of architecture, model, dataset and evaluation metrics are presented in each of these chapters separately.
Chapter 2

A Cloud-oriented Context-aware Middleware for Assisted Living

As discussed in Chapter 1, research into ambient assisted living (AAL) strives to ease the daily lives of people with disabilities or chronic medical conditions. AAL systems typically consist of varieties of sensors and devices. These sensors generate large amounts of medical data. However, these biomedical sensors lack the processing power to perform key monitoring and data-aggregation tasks, necessitating data transmission and computation at distributed middleware.

In addressing this issue and in relation to the first research question as presented in Section 1.4, this chapter introduces a Cloud-orientated Context-aware Middleware for AAL which we named CoCaMAAL model [81]. A cloud-enabled platform eases the management of large context-aware systems, allowing simplified user access and effectively handling demand elasticity [3]. The focus here is on the development of a scalable and context-aware framework and easing the flow between data collection and data processing. This is done by efficiently aggregating raw sensor data and the timely selection of appropriate services using a context management system (CMS). With a unified model that includes patients, devices, and computational servers in a single virtual community, AAL services are enhanced. The cloud platforms offers a high-level abstraction and its services can be accessed easily via mature web service protocols [91]. Our solution emphasizes a service-oriented architecture [79] that performs context modelling from raw data, context data management and adaptation, context-aware service mapping, service distribution, and service discovery. We have prototyped the model and implemented some case studies to demonstrate its effectiveness. The performance of the model
is evaluated using queueing theory.

The rest of this chapter is organized as follows. Section 2.1 highlights the motivations for this work and the challenges that are to be overcome to design the model. Limitations of existing solutions and related contributions are also discussed. Section 2.2 extensively analyses the contributions of existing models along with their drawbacks. The proposed CoCaMAAL model is presented in Section 2.3, with its intricacies detailed in Section 2.4. Some implemented case studies for testing the model functionalities are described in Section 2.5. Section 2.6 contains the experimental results on model performance and Section 2.7 summarizes this chapter.

2.1 Motivations and Contributions

As discussed in Chapter 1, assisted living systems aim to provide medical support and monitoring services in a cost-effective way to vulnerable sections of the community such as the elderly and disabled who live alone at home [92]. This section describes motivations, challenges and contributions of this chapter.

2.1.1 Motivations

The main motivations behind a cloud-based middleware are stated as follows.

- Modern aged-care and healthcare industries depend on service-oriented and context-driven assistive technologies [93]. Existing architectural solutions of context-aware middleware are confined to specific services [6, 51, 55, 56, 67, 94] and mostly rely on a local smart agent (i.e. mobile device) for context discovery and management. The lack of storage and power in wearable sensors and mobile devices limits them to process limitless sensor data using decent computational methods. Moreover, the discovery of new smart sensors and devices is increasing the demand for more intelligent and complex assistive services. Cloud computing escalates the capability of handling data in big volumes and the provision of versatile services. This persuades us to build a well-collaborative system by transferring the context processing task from a local smart device to a distributed cloud environment to improve the processing time of context generation and convey complex services.

- Traditional context management systems are incapable of handling large numbers of AAL systems together. They depend on standalone applications on a local server [51, 95].
This drawback encourages us to design cloud-oriented middleware that will be competent enough to handle a good number of clients simultaneously. The context derived from one AAL system will become a context knowledge bank for another AAL system inside the cloud repository. This is how the cloud middleware will be an extendable knowledge source of context for assisted living, and will able to deliver assistive actions quickly.

- The integration of cloud computing will expand the diversity of services for the AAL system. We are keen to develop a system that is able to deliver every kind of service using a single model. A cloud-based software service, a set of instructions as a web service or a cloud-based API for event alerts etc., can be implemented inside the cloud. That is, from context generation to service delivery, everything can be done using the cloud platform.

### 2.1.2 Contributions

To design an SOA with scalable computing facilities that efficiently supports the above objectives involves many challenges.

- The acquisition of data feeds from body sensors and other wireless devices in real-time, processing of heterogeneous data, and categorizing the data in an appropriate context is a major challenge.

- The integration task of distributed components and massive data sources is not so simple. The raw data generated from AAL systems contain large variations. The data can be a small binary data stream from the sensor, an analog signal, or even rich multimedia content such as streaming of video, voice, and images. Accurate processing of data with such distinctions and classifying the context to trigger relevant services in a short time is a very challenging task. It becomes more complicated when the data arrive in mass volume from a large number of systems.

- A key challenging issue in ubiquitous healthcare [12] is to choose accurate services from a large set for a given context and to interact with the target user immediately with acquired services.

- If the data sets and their corresponding services are geographically distributed, the allocation of storage and data migration becomes a critical challenge.
To overcome the above challenges, we are interested in designing our framework using cloud computing technology. Here are the major contributions of this chapter.

- We designed a cloud-based middleware which enables every user, from patients to healthcare professionals, to easily collect, access, process, visualize, archive, share, and search vast amounts of data from different AAL systems and service providers. Using the immense processing power of the cloud computing, it is easy to process highly swift data and provide quick responses to the user environment. Our framework acts as a decision support system in the backend and converts raw data to intelligent services. Sizable amounts of context and service information can be processed, analyzed, and stored using computational and storage resources of the cloud.

- Our solution using cloud computing allows sharing and reuse of information for different users and applications under flexible usage scenarios and thus minimizes the extra cost. As the major computation is performed in the cloud architecture, so sensors and devices can handle other specialized processing tasks. Cloud services are easily deliverable to a system having internet connections. Thus, our proposed solution actually simplifies the work of every constituent. It reduces the computing load of sensors, helps disabled people, and minimizes the work of healthcare professionals.

2.2 Related Work

In the area of context awareness several middleware-centric solutions are proposed in different applications other than assisted living. Table 2.1 shows a summary of those works and their limitations.

There are few research works for building situation-aware assisted living systems with sensor technology [9, 99]. Some of the works focus on promoting a user’s social interaction with his/her surroundings and co-ordination between several distributed actors such as caregivers and monitoring systems [57, 67]. European Union funded projects such as AALIANCE [100], PERSONA [50], and SOPRANO [101] aim at developing a next-generation smart home with ambient intelligence and scalable open standards for building a broad range of AAL services. I-Living [102] allows different parties to work together in a dependable, secure, and low-cost system manner. The Amigo Project [103] is a good example of an “Intelligent Home” for healthcare. A case-driven Ambient Intelligence technique is described in [94] by converting
Table 2.1: Some mentionable research works describing the framework of a Context-aware System (CaS)

<table>
<thead>
<tr>
<th>CaS</th>
<th>Contributions</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOCAM [51]</td>
<td>For context modeling a reasoning mechanism is suggested using OWL (web ontology language)</td>
<td>Most of the complex tasks are performed in local network where services need to be downloaded for use. This limits the model to serve specific services.</td>
</tr>
<tr>
<td>JACF [96]</td>
<td>It relies on object oriented context model developed in java framework</td>
<td>No clear indication about context-aware service management.</td>
</tr>
<tr>
<td>CAMidO [97]</td>
<td>The interpretation of context in here is policy based which collects and evaluates context by communicating with sensors.</td>
<td>React based on collected context from local system.</td>
</tr>
<tr>
<td>RCSM [98]</td>
<td>Provides flexibility for runtime context data acquisition, monitoring and detection using adaptive object containers (ADCs).</td>
<td>Only capable of triggering decision from application level.</td>
</tr>
<tr>
<td>5W1H [63]</td>
<td>Context model based on 5W1H(who, what, where, when, why and how). This model has capability of handling large number of context and provides flexibility of service provisioning.</td>
<td>Services are bound to specific smart environment services.</td>
</tr>
<tr>
<td>HiCon [55]</td>
<td>Supports advanced context-aware services which enforce scalable monitoring and composition of dynamic context.</td>
<td>The context model is not an accepted standard for AAL.</td>
</tr>
<tr>
<td>ERMHAN [56]</td>
<td>A context-aware service platform for supporting continuous care networks for home-based assistance.</td>
<td>Only focus on activity monitoring and related services.</td>
</tr>
</tbody>
</table>

context at a given time as a particular case which is obtained through activity recognition and case-based modeling.

In most of the proposed solutions, the functionalities of middleware are achieved by distributing the tasks in a layered architecture. There are several applications that leverage cloud resources to design efficient systems. Some of the research mainly suggested how cloud infras-
structure can be used for information context processing [104, 105, 106]. MoCASH [91] used the cloud computing feature for assisted healthcare but the concern was in the context sensing mechanism using mobile devices. There are also some cloud applications that have been developed for healthcare [4, 30, 34, 107].

All the contributions described above show current efforts for building solid architecture for AAL environments. With the support of cloud computing, AAL technology can be greatly enhanced for the deployment of innovative healthcare monitoring applications.

2.3 CoCaMAAL System Overview

The abstract architecture of the CoCaMAAL system is illustrated in Figure 2.1. The system comprises of five main cloud-oriented components: AAL systems, context aggregator and providers (CAP) cloud, service providers cloud, context-aware middleware (CaM) cloud, context data visualization and monitoring cloud. That is, every piece of the functional elements of the model copes with context and has a cloud-oriented infrastructure. A detailed overview of each of the components is given below.

Figure 2.1: The Generic architecture of CoCaMAAL model showing 5 major distributed cloud-based components of the proposed system.

- **AAL Systems**: Our proposed model can serve large numbers of AAL clients. AAL clients act as sensor data providers as well as context-aware service consumers. The setup of AAL systems varies based on target user requirements. Each of the AAL systems consists of different BSN foundations and monitoring systems [11, 27]. Any new AAL system can be included in the model without altering the architecture. All the AAL systems together form a distributed cloud structure. They generate raw sensor data for the model which become the input for high-level context generation and consume
context-aware services. For example, an AAL system generates raw heart beat data from an ECG sensor and waits for automated assistive services relevant to the context value of the heart beat rate.

- **Context Aggregator and Providers (CAP):** The CAP cloud contains the computing logic and process of converting low-level raw data to abstract context representation which is recognizable to all components of the architecture. The CAP cloud uses fusion and reasoning mechanisms to infer context from sensor data [108]. A context provider can be a medical server which manipulates ECG medical data, or it can be a weather station which provides context related to a weather forecast (e.g., temperature, humidity). For instance, the process of classifying the heart beat rate into a high, medium, or low category is hosted in a cloud server of a context provider which consumes ECG data as input and outputs the context of the desired classification. The context aggregator collects data from AAL systems and distributes them among different context providers for finding context. After getting high-level abstraction as context, the aggregator integrates all the information in a single context model. It then forwards that context description to the CaM cloud.

- **Service Providers:** Service providers (SPs) are applications and services related to context awareness. An SP can be a software application running on the mobile device of an AAL system that reminds the user about appointments, or it can be an external caregiver who monitors emergency situations. The SP cloud contains information such as symptom detection for different diseases and the types of actions required. For example, the rules for detecting fever and related actions are hosted in the cloud database of a service provider. The emergency actions that are required after detecting the fever of different patients are also described in the cloud storage. Thus, the SP cloud delivers various kinds of service to the CaM cloud.

- **Context-aware Middleware (CaM):** The CaM cloud is the most important functional component of the model. The CaM cloud has the infrastructure for processing context data, storing and retrieving context, context-aware service management, access control [109] mechanisms of medical records, context-to-service mapping, delivery of assistive actions, and many other complex computational tasks. It contains an intelligent context management system (CMS) which manages incoming context and ensures that appropriate assistive services are delivered properly and in a timely manner. A context manager
is responsible for storing and managing context that is gathered from AAL systems and converts them as services. The context manager automatically reconfigures its knowledge to adapt the change in context.

- **Context Data Visualization**: Context data contain the valuable medical information of the user. A proper visual interface is needed for the healthcare professional to view the information. Some data visualization services provide useful user interfaces for this purpose. Using a flexible GUI, the monitoring systems can examine medical records. The model also has different social networks of doctors and patient’s friends and family to analyze, visualize, and discuss medical data whenever required \[110\]. All these components together form a cloud source for context data visualization.

In our model, all the AAL clients, context providers, and service providers are physically distributed. The model takes the advantage of cloud computing by hosting all the context-processing and service-generation logic inside the cloud platform. Therefore, it becomes beneficial in terms of elasticity, storage scalability, and extensibility. Here, a new AAL client can be added or separated easily. A provider (context or service) can join or leave the system anytime. The context-aware cloud adopts its behavior according to the recent contextual situation and distributes services related to that.

### 2.4 System Description

In this section, all five major cloud components of the model are briefly described.

#### 2.4.1 AAL System

Our conceptual AAL system consists of a target user, body sensor network \[111\] and other ambient devices belonging to the type of observation and monitoring required for the user, the home environment of the user, a central workstation for capturing sensor data to send it in the cloud, and the software services running on the cell phones, smart devices, and workstations. The complete setup depends on available devices and the type of running services. Each AAL system has a unique identifier in the CoCaMAAL structure to identify it in the generalized cloud architecture. Whatever the requirement is, the objective here is to generate a unique model for context representation.
Table 2.2 shows some examples of typical body sensors. These kinds of sensors together form a body sensor network (BSN) [112, 113]. We want to eliminate the computational burden from these sensors so that they can perform their task for longer periods and more quickly.

Table 2.2: Examples of some typical body sensors and their use in AAL

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Measured Signal</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECG [30]</td>
<td>Electrocardiogram wave</td>
<td>Heart Rate</td>
</tr>
<tr>
<td>PPG</td>
<td>Photoplethysmogram wave</td>
<td>Blood Volume Pulse</td>
</tr>
<tr>
<td>BP</td>
<td>Blood Pressure in mmHg</td>
<td>Blood Pressure</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalogram wave</td>
<td>Abnormality</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyograph wave</td>
<td>Muscular activity</td>
</tr>
<tr>
<td>Accelerometer [114]</td>
<td>Acceleration in 3D space</td>
<td>Activity recognition</td>
</tr>
<tr>
<td>Motion sensor</td>
<td>Motion signal</td>
<td>User movement</td>
</tr>
<tr>
<td>Activity sensor</td>
<td>3-Axis motion</td>
<td>Activity recognition</td>
</tr>
<tr>
<td>Inertial sensor</td>
<td>Motion signal</td>
<td>Position detection</td>
</tr>
<tr>
<td>BG sensor</td>
<td>Blood sugar level</td>
<td>Diabetes detection</td>
</tr>
<tr>
<td>Gyrosopes</td>
<td>Rotation angle</td>
<td>Body orientation</td>
</tr>
<tr>
<td>Thermometer</td>
<td>Body temperature in °F</td>
<td>Fever detection</td>
</tr>
<tr>
<td>Insulin pump</td>
<td>-</td>
<td>Inject insulin</td>
</tr>
<tr>
<td>RF antenna</td>
<td>RF wave</td>
<td>Position detection</td>
</tr>
<tr>
<td>Fall Detector</td>
<td>Motion signal</td>
<td>Fall detection</td>
</tr>
</tbody>
</table>

The wearable sensors and ambient devices have appropriate wireless or wired connectivity to a local workstation (LW) in the AAL environment. The LW is responsible for collecting sensor data using suitable communication protocols and forwarding them to the cloud infrastructure. Body sensors and other devices collect medical data regarding the status of the user. For example, wearable sensors communicate using Bluetooth to a mobile phone or using Zigbee to a PDA which then forwards the data to the LW using WiFi (Figure 2.2). Even sensors can directly send data to the LW using Bluetooth. The communication depends on the availability of a suitable communication medium between the pairs. As an example, a more power consuming device such as a monitor is connected using the local area network, cameras are connected using USB connections and so on.

Each sensor and device has a unique identifier in the local system so that the LW can distinguish between incoming data of from different sensors and devices. A data collector program in the LW periodically collects sensor data. The program then passes the collected sample with device information to the cloud gateway using an optimized sampling rate. Every device sample contains a complete set of information. Finally, the collected data are uploaded to
Figure 2.2: **Data Acquisition** The raw sensor data produced by body sensors and other devices are transmitted to a local workstation using a connected medium. The local workstation then forwards the data to the cloud server using a cloud gateway.

the cloud servers using a cloud gateway. Figure 2.3 illustrates the abstract system architecture of our framework for a single AAL system. Simple and lightweight communication protocols (e.g. web service call) are utilized for transmitting data to the context aggregator cloud.

Figure 2.3: **Data flow in a single AAL**: Raw data from the AAL system is converted to context by the aggregator cloud. The service provider cloud contains the cloud storage for service rules. The context-aware middleware cloud manages incoming context and finds assistive actions. Then it responds back to the AAL system as a context-aware service.
2.4.2 Context Providers and Aggregator Cloud

2.4.2.1 Context Providers

The responsibility of context providers is to convert the sensor data into context. The amount of information that can be categorized as a ‘context’ in AAL system is an extremely large set. Several levels of data abstraction are required to get the exact context value. Moreover, different feature selection techniques are required for different sensor data. For example, classifying a user’s activity from accelerometer data and classifying a user’s heart condition using ECG cannot be solved using the same classifier. That is why our system has a large number of context providers which run their own feature selection, classification, and fusion algorithm to find the context.

Any classification problem which has a large input set is computationally expensive. The system needs to be properly trained to extract the features for classification. Existing health-care systems use local servers to solve specific classification problems such as user activity recognition, fall detection, and health monitoring. An efficient classifier requires huge data storage and memory to effectively find the features. If we can utilize the cloud platform for such computation then this will minimize the burden from the local servers. Each of the context providers in our system runs as cloud service which collects raw sample data as input and responds back with classification output using a SOAP-based web service. The context providers are distributed in the cloud structure. There are several classifiers used in context-aware computing for classifying sensor data such as ANN, Naive Bayes classifier, K-means clustering, HMM, C.5 Decision Trees, and others [115, 116]. When the classification is unknown, unsupervised learning is used. Sometimes fusion is required to obtain an overall classification result. The context provider uses the best classification technique for identifying the context. Figure 2.4 shows the general flows performed by a classifier of the context provider cloud. In our model many context providers can be integrated easily.

Figure 2.4: Flow diagram of Context Classifier: All the context providers perform this generic classification process to extract the high-level features. The method inside each of the square blocks can be different based on input and output data.
2.4.2.2 Context Aggregator

The context aggregator is the main collaborator of our model. It combines the data that arrive from different AAL systems. The aggregator has computational logic for context integration and interpretation. It is a distributed cloud service and responsible for the following operations of the system.

- Collect incoming sensor data from various AAL systems.
- Separate the data for sending it to the respective context provider to obtain the context classification.
- Track the identification of AAL systems, particular devices of AAL systems, and context providers.
- Collect high-level output as context from different context providers after classification process.
- The same context can be reported by different providers. The context aggregator resolves conflicts among duplicate data.
- If the same context has different values from different providers then it performs a performance score calculation through context provisioning to pick up the best and most reliable context value.
- Aggregate all the context information in a single context model against a particular AAL system.
- Send the context model to the context-aware middleware cloud for finding context-aware services.

Context providers perform the primary abstraction, and that is followed by the reasoning mechanism in the context aggregator to convert it into suitable information. This is the standard information that can be recognized by all the entities (sensors, context providers, and service providers) of our model. It is required for interoperability between data information from sensors and service-providing entities.
2.4.2.3 Context Modeling

A major aim of this chapter is to find a unified context model so that any context and context-aware services of AAL systems can be designed easily using the model. Without a well-defined, clear and flexible information model, applications will not be able to use such information in an efficient way. The model must be rich and flexible enough to accommodate not only the current facets of context information, but also future requirements and/or changes. The model is domain independent but it depends on provider services. Most of the existing context models focus on special aspects, whereas we focus on generalized model design which can ease context-aware service composition in assisted living.

The context model is an organized structure that is formed by the aggregator cloud after identification of the key contexts. It defines the context information that is storable in a data source, retrievable using simple query, and transmittable using a suitable format. It contains a list of topics with respective attributes. The topics are structured hierarchically. The context model is also a suitable place to specify which context information is transient and which is persistent. The context model only needs to be evolved if new context information is processed by the system, e.g. when a new sensing device is added or eliminated.

![Context Model using OWL](image)

**Figure 2.5: Context Model using OWL** Each context entity has some attributes to describe some basic properties of the entity. Some context entities are part of parent entity such as characteristics, diseases, preference, social and health ontologies are part of person ontology. Each of those entities has some more children to describe them. The relation among different entities is also shown here.

In our architecture, we adopted the ontology-based context model [117] [118] because it is one of the well-accepted context models for Ambient intelligent (AmI) systems. Figure 2.5 shows the proposed ontology-based context model based on OWL (Web Ontology Language).
The context space is described in four major entities.

- **Person** ontology is used to identify the user of the AAL system and his/her profile, diseases, health conditions, doctors, social interactions, and so on.

- **Place** ontology describes the current position of the user.

- **Environment** ontology is used to identify the conditions of surrounding environments. Environment has some impact for making decisions for assistive actions.

- **Device** ontology contains the details of the body sensors and devices of the system.

This kind of abstraction described above is easily extendible and modifiable. Using this model it is easy to derive new knowledge about the current context and to detect any inconsistency in the context data. Ontology makes reasoning tasks simple. The model is also helpful for service providers to define rules for services. Moreover, it is convertible to XML in the implementation level. XML is a flexible and platform-independent tool that can be used in different stages of information representation. It is also suitable for SOAP-based communication.

### 2.4.3 Service Providers Cloud

SPs contain the cloud repository of different kinds of service. In our model, service providers (Figure 2.3) subscribe to a context-aware middleware cloud structure and provide service rules using the context model which is deployed in the cloud structure. It can be a software service which is deployed in the cloud (SaaS) or the infrastructure for delivering services (IaaS). The services are described in the rule-based model using our ontology abstraction. Inside the rules of SPs the list of assistive actions that are required for observed conditions are integrated also. SPs hold this information and deliver it to the context manager on request by means of the cloud service. Some typical examples of services in AAL systems include software service, emergency assistance service, autonomic service, comfort service, and symptom detection service.

The same types of service can be different for different patients. The treatment of the same disease for a cardiac patient and for a diabetes patient will not be the same. More importantly, services are implemented by different service providers and they are large in number. So, it is almost impossible to deploy and install all the services in a single machine. If all services are deployed in the cloud there will not be any problem of resources. Besides, if all the services can be described using a unique model then it will make the service-mapping task easier.
Table 2.3: Service rules of detecting possible heart attack

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Instance</th>
<th>Raw data</th>
<th>Context Attr.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>User X</td>
<td>Profile</td>
<td>Age</td>
<td>≥65</td>
</tr>
<tr>
<td>Person</td>
<td>User X</td>
<td>Profile</td>
<td>Weight</td>
<td>≥80</td>
</tr>
<tr>
<td>Person</td>
<td>Disease</td>
<td>Profile</td>
<td>Cardiac patient</td>
<td></td>
</tr>
<tr>
<td>Device</td>
<td>ECG Sensor</td>
<td>ECG wave</td>
<td>Heart Rate</td>
<td>Abnormal</td>
</tr>
<tr>
<td>Device</td>
<td>BP Sensor</td>
<td>BP readings</td>
<td>Blood Pressure</td>
<td>Normal or High</td>
</tr>
<tr>
<td>Device</td>
<td>PPG Sensor</td>
<td>Sensor readings</td>
<td>$O_2$ consumption</td>
<td>Low</td>
</tr>
<tr>
<td>Device</td>
<td>Audio Sensor</td>
<td>Sound wave</td>
<td>Breathing</td>
<td>Irregular</td>
</tr>
<tr>
<td>Device</td>
<td>Camera, Radar, Accelerometer</td>
<td>Video, Images, 3D Acceleration, Motion path,</td>
<td>Motion</td>
<td>Tripping or falling or flailing of arms or any rapid motion</td>
</tr>
</tbody>
</table>

Table 2.3 shows the service rules of detecting a possible heart attack by using our ontology model. SPs can easily add or modify a rule using the simple query mechanism. In this way, one of several distributed service providers can offer their services using our cloud model. If a context sample contains such a pattern as described in Table 2.3 and the CMS of our model detects this, then it picks the assistive actions described for this service from the provider’s data storage. The consequences of the events happen locally inside the AAL system but the actions are mentioned in the assistive response.

There are different AAL platforms for generating services, for example, OpenAAL [119]. OpenAAL is a framework that supports integration and communications between AAL services using ontology. The rules of some context-aware services are presented in Table 2.4. The rules are stored inside the service cloud and retrieved using appropriate service provisioning mechanism [120]. Once a service rule is retrieved it can be easily described by our four-entity-based context model with assistive actions.

### 2.4.4 Context-aware Middleware (CaM)

The high-level context model that is prepared by the context aggregator cloud is forwarded to the context-aware middleware (CaM) cloud for the final output, which is context-aware services. As input data the CaM takes context from the CAP cloud and services from the SP cloud. It stores the context which is required for future use. Then, utilizing existing knowledge and incoming context the CaM identifies assistive services for the given context. It then immediately transmits the context-aware actions to the associated AAL system. The
Table 2.4: Example of some assistive services, their rules and the provider of the service

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Rules</th>
<th>Providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autonomic (Energy Saving)</td>
<td>(?user status : sleeping) ∧ (?room light status : on) → ACTION : Turn off light</td>
<td>Power Saver</td>
</tr>
<tr>
<td>Software (Medication Reminder)</td>
<td>(?local time : 5:00 PM) ∧ (?medicine Y time : true) → ACTION : pop-up mobile app for medication reminder</td>
<td>Software application</td>
</tr>
<tr>
<td>Comfort (Search Friend)</td>
<td>(?location : study room) ∧ (?Personal Laptop : on) → ACTION : Speaker suggest using voice to use site X for finding</td>
<td>Social Context Providers</td>
</tr>
<tr>
<td>Emergency (Fall Detection)</td>
<td>(?user status : lying down) ∧ (?fall detector status : on) → ACTION : Alert emergency assistance</td>
<td>Emergency Service Provider</td>
</tr>
<tr>
<td>Symptom Detection</td>
<td>(?skin temperature : high) ∧ (?breathing : abnormal) → ACTION : fever detected. Alert Personal doctor.</td>
<td>Personal doctor</td>
</tr>
</tbody>
</table>

context-aware cloud acts as a decision support system for the whole model.

2.4.4.1 Context Management System (CMS)

A context management system (CMS) interacts with different distributed components and binds the operations of each individual component together in the CaM cloud. The context manager (CM) contains the repository of context that is gathered from different AAL systems by adopting the IaaS properties of cloud computing. IaaS encompasses the storage, computing, and network for the CMS. The CMS is the main component of our SOA where the medical information of the AAL user is stored, analyzed, and accessed. The CMS dynamically scales the storage resources via on demand provisioning. It is in charge of handling requests from AAL systems via the aggregator cloud from where it obtains context information and relates the contexts with services defined by the SPs. Since it contains sensitive medical information, it maintains different public and private clouds for storing and accessing data.

The CMS applies intelligent matching criteria to find the assistive actions for the current context. The CM is self-adaptive in nature. It measures the performance to detect the best possible assistive actions for a context sample.
2.4.4.2 Context Storing

A distributed cloud repository ensures the persistency of context information (Figure 2.6). It contains up-to-date data and relevant context of the AAL user. The CMS stores context as an expandable knowledge base of ontologies (described previously). All the attributes are stored as key-value pairs in different ontology containers which simplifies querying the instantiations. Each of the AAL systems has a unique id, and all the devices in an AAL system also have some specific id to differentiate between incoming context data.

To help the monitoring service, it is useful to store recent context and to retain previously received context. Rarely used and old context is removed. An efficient access control mechanism is used for storing and retrieval sensitive data in the cloud.

2.4.4.3 Context Retrieval

The context manager periodically retrieves context for an AAL system via the context aggregator. The context information request is triggered by the context manager itself periodically. Third-party providers, monitoring services, and social networking services can also initiate context requests from storage. Any component can request context from the context manager through a web service running in the cloud server with an AAL system id and an attribute set of context. After filtering the duplicate and unwanted context a service in the CMS retrieves related information and returns the most relevant and accurate information.
2.4.4.4 Context Manipulation

Most of the context manipulation is done inside the CAP cloud. Inside the CMS a context transformer impacts on changes of certain information. For instance, body temperature expressed in °C scale can be transferred to the °F scale using the mathematical transformation rule. In addition, the CMS performs context derivation tasks such as finding a city from longitude and latitude information. The context manipulator also requests missing context from the AAL system via the context aggregator when required, and combines that context with retrieved information.

2.4.4.5 Service Mapping

The most important role of the context manager is to map context data to a respective service so that it can find the assistive actions to accomplish those services. The context model that is generated by our proposed ontology can be converted to an XML like Listing 2.1 with attribute values. A service profile also can be described using a similar XML structure, as shown in Listing 2.2. The CMS matches the context with possible services by comparing the value and calculating the weighted score matrix of the possible match. It then picks the best possible matches and combines them in assistive actions. Both the context model and service model can be represented in a tree structure that is shown in Figure 2.7.

![Context Tree](image)

Figure 2.7: **Context Tree**: Every context sample and service of our system is converted to such a context tree to find the service mapping.
System Description

Listing 2.1: A snippet of context XML

```xml
<contextML>
  <AALSystem id="1001" />
  <timestamp>2012-11-08T12:23:56+11:00</timestamp>
  <expires>2012-11-08T23:21:56+11:00</expires>
  <contextProviders>
    <contextProvider id="3eq32dfs" />
    <contextProvider id="1eq32daa" />
  </contextProviders>
  <context>
    <person id="421332">
      <name>Alice</name>
      <gender>Male</gender>
      <age>68</age>
      <health>
        <blood_pressure>High</blood_pressure>
        <skin_temparature>Normal</skin_temparature>
      </health>
      <diseases>
        <disease>cardiac</disease>
        <disease>alzheimer</disease>
      </diseases>
    </person>
    <place><location>Bed Room</location></place>
    <environment>
      <temparature>25 C</temparature>
      <humidity>40%</humidity>
    </environment>
    <device>
      <camera id="cs221">active</camera>
      <Monitor id="T3234">off</Monitor>
      <ECG_sensor id="S2312">
        <heart_rate>normal</heart_rate>
      </ECG_sensor>
      <ECG_sensor id="S2312">
        <heart_rate>normal</heart_rate>
      </ECG_sensor>
      <Accelerometer id="A5406">
        <activity>walking</activity>
      </Accelerometer>
    </device>
  </context>
</contextML>
```

The root of the tree in level 0 is context. In level 1 there should be four main entity nodes: person, environment, place, and device. The value nodes are located in the lowest levels that have no child node. The problem is to find the list of similar paths in the context tree which is the list of services described in the service trees.
Listing 2.2: An example of service XML

```xml
<serviceXML>
    <ServiceProvider id="weq9wq4234"/>
    <service> Detect possible heart attack </service>
    <context>
        <person>
            <health>
                <blood_pressure>High</blood_pressure>
                <heart_rate>Abnormal</heart_rate>
                <O2_consumption>Low</O2_consumption>
            </health>
            <Diseases>
                <disease>cardiac</disease>
            </Diseases>
        </person>
    </context>
    <result> Talk With user </result>
    <actions>
        <action> Turn on Monitor </action>
        <action> Turn on Speaker </action>
        <action> Turn on MIC </action>
    </actions>
</serviceXML>
```

Let the CM have the knowledge base of $M$ services in the service repository. $CT$ is the context tree for current context data and $\{ST_1, ST_2, \ldots, ST_i, \ldots, ST_M\}$ are the context trees for $M$ services $\{S_1, S_2, \ldots, S_i, \ldots, S_M\}$. Our matching criterion is, if any subtree is found in $CT$ which is exactly or very similar to tree $ST_i$, then $S_i$ is a context-aware service for context $C$. So, the action item described in the structure of $S_i$ is picked which is the assistive action for $S_i$. The matching operation is performed for all the $M$ trees and all the actions are combined in a single XML structure. Then the CMS responds back to the AAL system with the action list.

$NS−CT$ is the list of nodes and $ES−CT$ is the list of edges. Similarly, $S−ST_i=(NS−ST_i, ES−ST_i)$ is a subtree of $ST_i$. To find the similarity between $S−CT$ and $S−ST_i$, we first need to find the similarity between the root nodes of $S−CT$ and $S−ST_i$ and then progress through the nodes of the subtree up to child nodes recursively. At the lowest level of the tree the matching algorithm just matches the higher level context values.

The similarity between two nodes $N_{CT_x}$ in tree $CT$ and $N_{ST_{y}}$ in tree $ST_i$ is $w=sim(N_{CT_x}, N_{ST_{y}})$, $0 \leq w \leq 1$

- $w = 1$, if nodes are exactly similar, i.e.: any parent node.
- $w = 0$, if nodes are not similar, i.e.: mismatch in name or value.

For any parent node, $sim(N_{CT_x}, N_{ST_{y}}) = sim(\text{tag name of } N_{CT_x} \text{ and } N_{ST_{y}})$. So, $w$ is either 0 or 1. If $w = 0$ for any parent node then that subtree is skipped from the
matching as that is not requires. For the child node the \( w \) value is between 0 and 1, when child node value is numeric and numerical comparison is performed between node value.

So, if \( ST_i \) has \( z \) nodes which are exactly similar or nearly similar to \( CT \) then the similarity matrix of \( ST_i \) is:

\[
W_{ST_i} = \begin{bmatrix}
w_{ST_{i1}} \\
w_{ST_{i2}} \\
\vdots \\
w_{ST_{iz}}
\end{bmatrix}, \quad \text{where} \quad 1 \leq i \leq M
\]

Among the \( z \) nodes let \( p \) nodes be child nodes and the remaining \( z - p \) nodes be parent nodes. Since all parent node similarity values are 1, we consider only child nodes for calculating the score matrix of \( W_{ST_i} \):

\[
score_{W_{ST_i}} = \sum_{j=1}^{p} w_{ST_{ij}}.
\]

The score matrix of all services is

\[
score_S = \begin{bmatrix}
score_{ST_1} \\
score_{ST_2} \\
\vdots \\
score_{ST_p}
\end{bmatrix}
\]

Therefore, the CMS picks the services and related actions with good score values.

2.4.4.6 Self-adaptation

When the CMS calculates the similarities between context and service structure, it can identify a set of actions and enrich the knowledge base by storing this information inside the cloud repository. When the same pattern in the context is detected, it is then able to respond faster than before. Furthermore, when any providers update their information, the CMS synchronizes the database accordingly. Because a change in service means a change in the service context tree. The CM needs to recalculate the score of service for a context sample.

As an example, from AAL\(_1\), the CMS has identified a list of services \( \{S_1, S_2, \ldots, S_n\} \) and corresponding actions \( \{A_1, A_2, \ldots, A_n\} \) for a context tree \( CT_1 \) and stored that information in the context cloud repository. Now if the same context tree appears from AAL\(_1\) then without performing mapping the CM delivers the actions from its past knowledge. If another AAL system AAL\(_2\) requests services with context tree \( CT_2 \), similar to \( CT_1 \), then the CMS can easily pick the actions required for \( CT_2 \) and deliver it without performing additional mapping. As a result, the CMS can acknowledge faster by self-adaptation.
2.4.4.7 Service Discovery

Not all the assistive actions are described by the service providers. In some cases the CMS builds compound services from a composition of existing services. This is known as service discovery in the system.

2.4.4.8 Security Service

The health-related context that is collected, processed, and managed inside the cloud is sensitive. The security service inside the CMS will ensure the privacy of context information that is gathered from different AAL systems. Different solutions are proposed to protect personal health information [91, 121, 122] in distributed cloud environments that ensure the privacy of the services. Context-aware role-based access control and a privacy-preserving context service protocol [121] can be adopted to ensure the privacy of context information in our model.

2.4.5 Monitoring and Data Visualization Service

Inside the CaM every action is event driven but also requires manual tracking in some cases. In CoCaMAAL we suggest different kinds of web interface for managing context-aware middleware. These kinds of interface are essential to store and retrieve context. Other data visualization interfaces parse the context and present it in a visual interpretation. Also these medical data can be published in a social network of doctors. Thus, doctors have an up-to-date health status of patients. We believe that such type of monitoring services are easily adoptable in our model. But the in-depth analysis of such services is beyond the scope of this research.

2.5 Case Study

The major objectives for building the system described in this chapter are as follows.

- Detect current situational information of an elderly user under assisted living from various context sources.
- Aggregate the information in a meaningful context model.
- Manage the context in a distributed cloud environment.
- Find related services from service providers depending on the context to assist the user.
In accordance to the CoCaMAAL functional components described in the previous section, we have developed a simulated prototype, implemented mostly in Java, to show the feasibility of the proposed model. The elements of this simulated model are presented in Figure 2.8 and the setup of the home domain is described in Table 2.5.

Figure 2.8: Evaluation: The experimental setup of simulated model for evaluating the case studies described in this section.

Table 2.5: The components considered in the experimental setup of the simulated prototype.

<table>
<thead>
<tr>
<th>User locations</th>
<th>Living room, Bathroom, Kitchen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body Sensor Network</td>
<td>ECG sensor, PPG sensor, BP sensor, Accelerometer, RF tags</td>
</tr>
<tr>
<td>Devices of Living Room</td>
<td>RFIDs, Camera, Mic, Speaker, TV, Emergency Alarm</td>
</tr>
<tr>
<td>Devices of Bath Room</td>
<td>RFIDs, Mic, Speaker</td>
</tr>
<tr>
<td>Devices of Kitchen</td>
<td>RFIDs, Mic, Speaker</td>
</tr>
</tbody>
</table>

We have synthesized numerical values from realistic observation of medical data (e.g., heart rate is 65). The generated values represent raw data from the ECG, PPG, and BP sensors and the accelerometer. Four rule-based classifiers are developed as context providers (as in 2.8) to classify those numerical values to high-level context (e.g., user activity is sleeping [123], heart
rate is high). All the data are aggregated using our Java-implemented context aggregator, and finally the context model is generated using the suggested ontology.

Table 2.6: Aggregated high-level context samples

<table>
<thead>
<tr>
<th>No</th>
<th>Time</th>
<th>Heart Rate</th>
<th>$O_2$ saturation</th>
<th>Blood Pressure</th>
<th>Location</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23:10:12</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Living Room</td>
<td>Watching TV</td>
</tr>
<tr>
<td>2</td>
<td>10:11:45</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Living Room</td>
<td>Sitting</td>
</tr>
<tr>
<td>3</td>
<td>09:42:78</td>
<td>High</td>
<td>Normal</td>
<td>Normal</td>
<td>Living Room</td>
<td>Exceeding</td>
</tr>
<tr>
<td>4</td>
<td>01:43:21</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Living Room</td>
<td>Taking Medicine</td>
</tr>
<tr>
<td>5</td>
<td>05:22:34</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Bath Room</td>
<td>Fallen Down</td>
</tr>
<tr>
<td>6</td>
<td>19:17:28</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Kitchen</td>
<td>Cooking</td>
</tr>
</tbody>
</table>

We have randomly generated 3000 samples for a single-user scenario considering a sampling rate of 10 minutes. Each sample contains different physiological parameters, current activity, time, and location of the user. A piece of such sample data is presented in Table 2.6. User-specific information (e.g. medication time, disease, age) are stored in MySql database. We also created 10 service providers ($sp_1, sp_2, ..., sp_{10}$) and distributed 67 different fuzzy rules among them. The rules are stored in XML files (as in Listings 2) including the list of possible actions for context matching. Log entries of unusual events (i.e., user has not taken medicine in time) are created with timestamps in the database and kept as context history. Every single sample (as in Table 2.6) is processed inside the CMS one by one. After the rule matching and service mapping process described previously, the CMS selects the best possible services from multiple SPs.

We used Java for prototyping and Google App Engine (GAE)[124] for evaluating the cloud infrastructure. GAE is an open-source cloud computing platform for developing and hosting applications in Google-managed data centers. Table 2.7 summaries the different software tools we used to simulate the components of the CoCaMAAL model and to examine the following case studies.

- **Case Study 1**

  1. The AAL system generates featured context of the user as described in Table 2.6.
  2. A sudden observation contains abnormal values (No 2 of Table 2.6)
  3. From extracted information in the profile database, disease context is also included (i.e., cardiac disease and hypertension) in the information.
Table 2.7: Components developed for simulated prototype

<table>
<thead>
<tr>
<th>Model Components</th>
<th>Simulated components</th>
<th>Development Platform</th>
<th>Output format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wearable Sensors</td>
<td>Java Servlet as Numerical Data Generator</td>
<td>Java thread in Tomcat Server</td>
<td>Data files</td>
</tr>
<tr>
<td>Data Collector</td>
<td>Java Servlet</td>
<td>Java thread in Tomcat Server</td>
<td>Data files</td>
</tr>
<tr>
<td>Context Providers</td>
<td>Java program running in GAE</td>
<td>API using GAE</td>
<td>High level context</td>
</tr>
<tr>
<td>Context ontology</td>
<td>OWL</td>
<td>GAE Data Source for ontology containers</td>
<td>Context model</td>
</tr>
<tr>
<td>Context Aggregator</td>
<td>Java program running in GAE</td>
<td>API using GAE</td>
<td>Context model as XML</td>
</tr>
<tr>
<td>CM Repository</td>
<td>Static database</td>
<td>GAE Data Source</td>
<td>Database records</td>
</tr>
<tr>
<td>Service Providers</td>
<td>Static XML in different web servers</td>
<td>GAE web server</td>
<td>Simple service rules with actions.</td>
</tr>
<tr>
<td>Service Mapper</td>
<td>Java program running in GAE</td>
<td>API using GAE</td>
<td>List of assistive actions as XML</td>
</tr>
</tbody>
</table>

4. After the rule matching step the CMS finds 21 services stored in different service providers.

5. The CMS picks the best possible services (using the technique stated in section 2.4.4.5).

6. The CMS reports *heart attack symptom* as a detected service. From the location and activity context the CMS detects that the user is sitting in the living room (No 2 of Table 2.6), and according to the setup of the domain the living room contains a speaker, microphone, and TV Monitor (Table 2.5) for communication. So, the CMS appends *communicate user with voice* as assistive actions in the response. This response is converted to XML and sent to the tomcat server running inside the AAL system as a SOAP response.

7. The server receives the response and sends the appropriate command to turn on the monitor and speaker (in the simulated model a flag is set in the class object of the device).

8. The voice conversation with user is a question-answer session. The questions are picked from the service providers and appended at the time of generating the service
response. A sample question is “Are you feeling chest pain?” [125].

9. The recorded conversation (as voice or video) is sent to the context aggregator to obtain more specific context like Chest pain is true. (in the case of the simulated environment it selects a random answer).

10. The CMS runs another service mapping with new context (Chest pain) and picks new service actions based on the context value. (for chest pain true value it picks the service ”risk of heart attack”)

• Case Study 2

1. The CMS receives the context generated in step 3 of case study 1 and requests some more context like ”taking medicine activity for the day” from the context history database.

2. The RF reader of the living room tracks user’s medication [8, 126]. If the user does not take medicine in time the system logs this event in the context history from the status of the RF reader. In our experiment, the CMS detects this event from a retrieved entry in the database. So, after rule mapping it picks medication reminder service and as an assistive action it reminds the user to take his/her medicine properly.

• Case Study 3

1. As in sample 5 of Table 2.6 from context the CMS detects that user’s physical condition is not good and that she/he fallen down in bathroom [127, 128].

2. The CMS identifies the situation as ”Emergency Service” from service mapping. As assistive action it notifies the server to ring the emergency alarm in the house so a nearby neighbor can come to help and it also sends an automated emergency alert to doctor (using sms).

2.6 Experimental Results

We first evaluated the model for a single AAL system with different context and service load. For each of the cases, the average service response time \( T_{response} \) is calculated. The time starts when raw data are sent to the CoCaMAAL aggregator cloud from the local server and ends when the AAL system gets a service response. So, \( T_{response} \) includes the context processing
time, service mapping time, and service delivery time. $T_{\text{response}}$ is measured by increasing the number of context sources, service providers, and services. The result is shown in Table 2.8. We also measured the service mapping accuracy to ensure that the system is picking the correct services for a scenario. To compute the accuracy, we compared the observed results with expected services and made a true/false decision for it. For 21 contexts with 67 services in 10 service providers and 3000 samples the measured accuracy was 96.34%.

Table 2.8: Average Service Response Time for different Context, Service Provider (SPs), and service load. **Note that in the 2nd column of item 2, 1 and Activity refers to Activity, Location and Time including Physical parameters of row 1 and so on for other rows of row column**

<table>
<thead>
<tr>
<th>Item</th>
<th>Context Domain</th>
<th>Number of Contexts</th>
<th>Number of SPs</th>
<th>Number of Services</th>
<th>$T_{\text{response}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Physical parameters</td>
<td>4</td>
<td>1</td>
<td>10</td>
<td>18 ms</td>
</tr>
<tr>
<td>2</td>
<td>1 and Activity, Location, Time</td>
<td>7</td>
<td>3</td>
<td>21</td>
<td>21 ms</td>
</tr>
<tr>
<td>3</td>
<td>2 and Environment</td>
<td>11</td>
<td>5</td>
<td>35</td>
<td>23 ms</td>
</tr>
<tr>
<td>4</td>
<td>3 and User Profile</td>
<td>14</td>
<td>7</td>
<td>43</td>
<td>28 ms</td>
</tr>
<tr>
<td>5</td>
<td>4 and Context History</td>
<td>16</td>
<td>8</td>
<td>54</td>
<td>30 ms</td>
</tr>
<tr>
<td>6</td>
<td>5 and Device Status</td>
<td>21</td>
<td>10</td>
<td>67</td>
<td>31 ms</td>
</tr>
</tbody>
</table>

The above results prove the benefit of the CoCaMAAL model in terms of selecting the correct services for a given context in short time.

To measure the performance of the CoCaMAAL model with a large number of AAL systems we adopted $M/G/1$ queuing system with FCFS [80]. In $M/G/1$, the inter-arrival time of request is exponentially distributed, the service time is generally distributed, and the total number of processing servers is 1. The model is presented in Figure 2.9. The model can be extended to an $M/G/c$ queuing system [129]. Computation of $M/G/c$ is very complex and beyond the scope of this thesis. For the sake of simplicity, here the whole CoCaMAAL model is considered as a single ultra-fast processing unit of an $M/G/1$ model to serve $n$ number of AAL systems. So, here the number of processing servers $c=1$, and the processing of $k$ different requests is distributed in $k$ different nodes of this very ultra-fast processing server. Our goal is to find the mean response time of a request.

If the number of requests from $n$ AAL systems is 120/minute then the service request rate is a Poisson process with mean $\lambda = \frac{120}{60} = 2$ requests/sec.

The service response time depends on the size, type and value of the context. Here, the service time is generally distributed with mean $\frac{1}{\mu}$ sec and variance $\sigma^2$
Experimental Results

Figure 2.9: The $M/G/1$ queuing system of the CoCaMAAL model

Table 2.9: General distribution of Service Time of five different sets for $k=10$ and $\lambda = 2$ sec

<table>
<thead>
<tr>
<th>Set</th>
<th>$S_i$ in % and $T_i$ in milliseconds</th>
<th>$\rho$</th>
<th>$W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>22 20 17 14 12 9 8 6 5 4 3 0.062 0.0326</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>28 38 20 11 10 7 5 2 1 0.060 0.0311</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>25 12 19 14 12 10 6 4 2 2 0.062 0.0325</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>21 12 18 16 10 9 8 5 3 1 0.060 0.0315</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>22 33 15 12 12 10 7 6 2 3 0.062 0.0324</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The average service time to process a request is

$$\frac{1}{\mu} = E[S] = \sum_{i=1}^{k} S_i T_i \tag{2.1}$$

Where $S_i$% requests are processed in $T_i$ seconds and $k$ is the number of request types. The variance $\sigma^2 = E[s^2] - (E[s])^2$. In our evaluation, we considered ten request types. So, here $k = 10$ and for $S_i$% requests, services are selected from the context with overall mean
processing time (inside CoCaMAAL) $T_i$ ms. The distribution for five different configurations (for $k=10$) are presented in Figure 2.9. The processing of the request containing large data (i.e., image) takes a longer time than the processing time of the sensor data (i.e., accelerometer data). So, in Table 2.9, different percentages of the requests have different processing times.

For example, in the case of S1 we have the following.

Using equation 2.1, we got $\frac{1}{\mu} = E[S] = 31$ ms = 0.031 sec

Variance $\sigma^2 = E[s^2] - (E[s])^2 = 0.000538$ sec$^2$.

Utilization $\rho = \frac{\lambda}{\mu} = 2 \times 0.031 = 0.062$

According to the Pollaczek-Kinchine formula [80], we obtain the following.

Average number of requests $L = \rho + \frac{\sigma^2 + \lambda^2 \sigma^2}{2(1-\rho)} = 0.0652$

Average response time of a request $W = \frac{L}{\lambda} = 0.0326$ sec.

Figure 2.10: (a) Using the lower value of the request arrival rate ($\lambda$). (b) $\lambda$ is high. That is, requests are arriving in high volume from a massive number of AAL systems. For both (a) and (b) the average response time ($W$) increases with the increase of request arrivals. The response time is still very fast (only 0.11 sec) even with a high request arrival rate($\lambda = 60$).

Figure 2.11: The average response time($W$) varies with the distribution of service time in different processing nodes for different observations. Using only a single ultra-fast server with $k$ processing slots, the average response time (using the same arrival rate) is very low for each observation. For $k \geq 5$, the response time of each observation is very similar.
The $\rho$ and $W$ values for each of the cases are presented in Table 2.9. We found that the service response is much faster in the CoCaMAAL model with a large number of requests when they are served by a ultra-fast processing unit running on a cloud.

We also evaluated the model by varying $k$ and $\lambda$ and the results are given in Figure 2.10 and 2.11. From the obtained results we find that the CoCaMAAL model responds much faster with good service recognition accuracy even for a large number of AAL systems.

2.7 Conclusion

In this chapter, we have presented a generic model to support AAL systems with context-aware service management using cloud computing platforms. The proposed model is a complete framework to support distributed features of context awareness in AAL. We started by describing the limitations of existing systems and the challenges that AAL users and healthcare professionals are facing in the traditional computing model. Our system analyzed the issues related to scalability, cost, and support of heterogeneous services using a single model. We suggested the need for a suitable context model for the AAL system which is easily modifiable, reusable, adaptable, and extendable. We demonstrated our model by describing the setup and the major functionalities of the components and their communication standards. The model is tested with some case studies and performance is evaluated using queueing theory.

2.8 Epilogue

We have a complete cloud-based model which can support incoming data from multiple AAL systems simultaneously. We want to explore and possibly extend this model to process sufficiently large data. We target the implementation of a more complex context processing task such as data mining inside the context management system. In the following chapters, we will use this model to perform complex data processing task for personalized knowledge discovery, abnormality prediction, behavioural change detection and future clinical event predictions.
Chapter 3

A Personalized Knowledge Discovery Framework

We developed a cloud-oriented model and proved it suitable for our problem domain. Such a framework is essential to process large data generated from many assisted living systems. In this chapter, we propose a knowledge discovery-based approach that allows the context-aware system to adapt its behaviour in runtime by analysing big data generated in AAL systems and stored in cloud repositories. In Chapter 2, we presented CoCaMAAL and proved its advantages in processing and managing large amount of contexts gathered from multiple AAL systems. In CoCaMAAL we described the context-aware service identification process through simulated case studies using high-level generalized medical rules. The model lacked an important feature, namely personalized knowledge discovery which could be derived from a large amount of patient data stored in the cloud repositories. Therefore in this chapter, we developed BDCaM (Big Data for Context-aware Monitoring), an extended version of the CoCaMAAL, which includes the functionalities of learning and knowledge discovery process to find patient-specific anomalies using big data. This also corresponds to the first part of the second research question on how to detect abnormalities in current context by discovering personalized knowledge and using corrections of different situational contexts.

As discussed in Section 1.1 large number of AAL systems can generate big data in terms of petabytes in every year. According to IBM data scientists, big data can be characterized in four dimensions: volume, variety, velocity, and veracity (“the 4 V’s”) [15]. BDCaM model also satisfies these four V’s because the context-aware data we are referring to have massive variations (e.g. health data, activity data), is large in volume (several petabytes), continuous
in terms of velocity and accurate to satisfy veracity. Such data also have great value and high impact on future healthcare infrastructure. The predictive analyses over large historical data provides robust solutions for disease prevention. This also simplifies the tasks of healthcare professionals and doctors by assessing the causes of any anomalous situation at an early stage and improving the quality of life of a patient.

The proposed BDCaM model facilitates analysis of big data inside the developed cloud-based middleware. It first mines the trends and patterns in the historical data of an individual patient with associated probabilities and utilizes that knowledge to learn proper abnormal conditions. The outcomes of this learning method are then applied in context-aware decision-making processes for the patient. A use case is implemented to illustrate the applicability of the framework that discovers the knowledge of classification to identify the true abnormal conditions of patients having variations in blood pressure (BP) and heart rate (HR). The evaluation shows a much better estimate of detecting proper anomalous situations for different types of patients. The accuracy and efficiency obtained for the implemented case study demonstrate the effectiveness of the proposed model.

The rest of this chapter is organized as follows. Section 3.1 describes the motivations and contributions. Section 3.2 summarizes the related works in the area. The architecture of the BDCaM model is presented in Section 3.3. Section 3.4 describes the definitions and concepts used for the model. Section 3.5 introduces the detailed methodology and algorithms used for the model. A case study along with experimental evaluations and the results of proposed methodology are illustrated in Section 3.6 and 3.7. Finally, Section 3.8 summarizes this chapter.

3.1 Motivations and Contributions

3.1.1 Motivations

The main motivations behind this chapter are the following.

- The need of an abstract context-aware framework that improves the confidence of abnormality detection in the home healthcare environment by correlating physiological statistics with various physical activities and environmental factors.

- The use of cloud computing enables faster learning with greater knowledge from continuously generated big data gathered from heterogeneous contexts of various assisted living systems. This also improves the discovery of user-specific rules with stronger support.
• The improved knowledge of understanding the patient’s situation through iterative learning on present contexts and substantial historical data can reduce the transmission of repeated false alerts to the remote monitoring systems.

3.1.2 Contributions

The primary contributions of this chapter are as follows.

• We build an innovative architectural model for context-aware monitoring, BDCaM that uses cloud computing [3] platforms. Every generated contexts of AAL systems are sent to the cloud. A number of distributed servers in the cloud store and process the aforementioned contexts to extract required information for decision-making using this novel technique.

• We develop a 2-step learning methodology. In the first step, the system identifies the correlations between context attributes and the threshold values of vital signs. Using MapReduce Apriori algorithm [84] over long-term context data of a particular patient, the system generates a set of association rules that are specific to that patient. At the second step, the system uses supervised learning over a new large set of context data generated using the rules discovered in the first step. In this way, the system becomes more robust to accurately predict any patient situation.

• We demonstrate the performance and efficiency of BDCaM model in situation classification by implementing a case study. Our system refines patient-specific rules from big data and simplify the job healthcare professionals by early detection of anomalous situation with good accuracy.

3.2 Related Work

In the research literature, many examples exist that introduce an integrated system using big data for context-awareness in assisted healthcare. However, most systems are described from an architectural point of view and there has been no practical implementation of any of those systems [91, 130]. In the previous chapter, we described the CoCaMAAL [81] model and here we have extended that model to illustrate the learning process from big context-aware data to find abnormalities in an individual patient. In our system we have used the MapReduce-Apriori algorithms proposed in [84] which is an effective process to measure correlations between
context attributes. MapReduce is also an efficient programming model to process big data using distributed clusters. In an AAL system, most of the generated contexts are numerical or categorical. So together with MapReduce-Apriori, we used the techniques described in [131] to generate rules using numerical attributes of our model. As we do not intend to explore the generation, preprocessing and transmission of real sensor data in the present work, we have assumed that subsystems are responsible for doing this from existing knowledge.

There have been several studies about the context-aware approach for assisted healthcare. The different works are differentiated by: context-aware platform for supporting continuous care [56, 67], activity monitoring [50, 52], cloud-based healthcare [34, 91] and personalized care [83]. The context-aware systems that are developed using rule-mining and data mining only solve some specific diseases [66, 132]. That is, most proposed systems are restricted to supporting some specific context-aware services and are not capable of detecting a wider range of anomalies. The system that relies on generic rules is not able to predict all the critical situations and suffers from misclassification of normal situations. The studies of big data for healthcare mostly focus on the area of mining electronic health records [133], feature extraction from medical image or pattern recognition based on genome data. A very few works have done that combine context-awareness with big data to develop a generalized system for assisted care.

The necessity of context-awareness in continuous BP monitoring was proven in some previous works [134, 135, 136] that encouraged us to evaluate the experiments of BP monitoring with large context data having massive variations. The effectiveness of context-aware monitoring for anomaly detection and disease prediction is also studied in some research works [18, 56, 66]. The need of cloud-based model for such data-intensive systems with high quality of service [137] and user-specific preference [138] using low cost MapReduce process [139] and systematic data mining techniques [140] is also established in some recent contributions.

All these contributions have motivated us to develop this cloud-enabled system with big data. The unique advancement of our model is to learn user-specific anomalies accurately in an assisted living system and take immediate context-aware actions. The robust learning methods reduce unnecessary false alerts to the monitoring systems.

### 3.3 Architecture

The general architecture extended from CoCaMAAL model for the proposed knowledge discovery-based context-aware model is visualized in Figure 3.1. The flow of raw data, context, rules and services between different distributed components are also shown in Figure 3.1 and as a work-
flow diagram in Figure 3.2. The overall architecture can be split into five cloud components. Most of these components are briefly described in Chapter 2. The following subsections discuss the new components of BDCaM model including a short description existing components in context of BDCaM model.

![Diagram of BDCaM model](image)

**Figure 3.1**: The complete architecture of BDCaM model showing all the cloud-based components of the system.

### 3.3.1 Ambient Assisted Living (AAL) Systems

The big data producers of BDCaM model are a large number of AAL systems. To learn the daily activity patterns of the patient and the effects of other contexts on his/her medical conditions, all data need to be stored and processed. The high level contexts are obtained from these low level data. The big data scenario of a single AAL system is shown in Figure 3.3.

### 3.3.2 Personal Cloud Servers (PCS)

Each AAL System is connected with a personal cloud server (as in Figure 3.1). This is a virtual server in the cloud that is highly scalable and managed by trusted entities. It has secure storage facilities to store patient-specific information (e.g. Amazon S3, Microsoft HealthVault) such as the profile (e.g. age, sex, BMI), recognized patterns of his/her daily activities (e.g. smoking habits), identified threshold values of different vital signs [49], medication time, disease...
Figure 3.2: Data flow between cloud components of BDCaM model.

Figure 3.3: Big data case for context-awareness in AAL

treatment plans, prescriptions, preferences, emergency contacts and personal medical records. The local processing device in AAL system (data collector) can easily exchange information with the PCS. In some cases, the PCS can contain latest pathological and laboratory test
reports, biomedical images (e.g. X-rays) or even raw sensor data (e.g. the latest ECG) that is produced in AAL system [141]. When daily patterns or personalized medical rules are learned, they are stored in the PCS and thus can be retrieved easily when required.

### 3.3.3 Data Collector and Forwarder (DCF)

Traditional context-aware systems process the low level data and perform the computation in a local server or mobile device and then forward the high level context data to the cloud [56]. But the lack of storage and power in wearable sensors and mobile devices limits them to process large volume of sensor data using decent computational methods. In our proposed model, the job of a local server (which can be a mobile device) is only to collect the low level data (i.e. accelerometer data, ECG data, BP Monitor data, GPS coordinates, RFID status, captured images) from the AAL system and forward them directly to the CA (when processing is required) or to the PCS (Figure 3.1 and 3.2). From current knowledge we are assuming that the DCF has the mechanisms (e.g. blue-tooth) to communicate with all sensors and devices that produce raw data. The computations for the conversion from low level data to high level context are performed inside cloud servers.

### 3.3.4 Context Aggregator (CA)

The job of the context aggregator (CA) is to integrate all the primitive contexts together in a single context state using a context model [81]. Sometimes a single context attribute value as an individual has no meaning if it is not interrelated with other contexts. For example, an increment in HR seems an abnormal condition as a single context, but at the same time if the activity of the user is exercising, then 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 together to classify a situation [36] accurately. The CA does this work and forwards the information to the context management system for the individual user.

### 3.3.5 Context Providers (CP)

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 [48, 142], 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 (Figure 3.2).

3.3.6 Context Management System (CMS)

A Context management system (CMS) is the core component of the framework. The CMS consists of a number of distributed cloud servers that hold the big data. It stores the context histories of millions of patients. Different machine learning techniques run inside the CMS [63] that infer different personalized and generic rules for various user events. When the CMS discovers any personalized rules, they are sent to the corresponding PCS. Any newly identified generic rules are forwarded to the service providers (SP) cloud. This is how the CMS keeps every component of the model up to date with new knowledge. Sometimes, existing rules are required to reason new high-level knowledge. In that case, CMS uses general rules from SPs or personalized rules from PCSs (Figure 3.2). After rule generation, the CMS runs another training stage, using different data mining algorithms to obtain the best classifier for a situation classification. Once optimized accuracy is achieved, the CMS retains the 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. 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 (three clusters are shown in Figure 3.1).

3.3.7 Service Providers (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 behaviours 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.

3.3.8 Remote Monitoring Systems (RMS)

When the CMS discovers any anomalous pattern in the context for a specific user it sends appropriate notification to the RMS. For example, when the BP level of a patient goes relatively high for a given situation the CMS alerts the doctor to investigate it, but if it goes abnormally high then the CMS sends alerts to the emergency centre. Thus, the selection of RMS depends
on situation classification. A major goal of our system is to classify a situation correctly to send proper alerts to the right RMS.

### 3.4 Definitions and Concepts

As stated, in our model by context we mean any high level information of AAL system such as, user activity, location, HR, BP, ambient temperature, disease history and age. In each AAL system, different contexts are sampled in different time intervals in multiple domains. To represent a domain, its information and context space we have the following definitions.

**Definition 3.1.** In any AAL system a single context attribute $a_i$ has value $v_i$, where $v_i$ is obtained from a set possible values of $a_i$ which is described by $A_i$. That is, $v_i \in A_i$.

The set $A_i$ is directly generated from sensor readings or using some well-known techniques such as sensor data classification, activity recognition [48], sequential pattern mining [143], image processing [50], and other data mining approaches. All these processes run inside context providers cloud.

**Definition 3.2.** For a single AAL system, the information of domain $k$ at any given time $t$ is represented by $I_{D_k}^t = \{a_i^t|1 \leq i \leq n, a_i \in A_i\}$. Where, $n$ is the number of context attributes observed from that domain, $a_i$ is the $i$-th context attribute which has a value from $A_i$. $A_i$ is a set that can have values within numerical range, $A_i = [v_i^{max}, v_i^{min}]$ or discrete types, $A_i = \{v_i^1, v_i^2, ..., v_i^m\}$. If any information is spanned between start and end time, then such information is presented by, $I_{D_k}^{t\rightarrow t+\lambda}$, where $\lambda$ is the time duration. The static information of persistent domain $D_S$ have the from $I_{D_S} = \{a_{S_i}|1 \leq i \leq n_1, n_1$ is number of static context.}

For example, vital signs is a domain that reports the health conditions of a patient in terms of BP, HR, respiration, body temperature etc. These are the context attributes of vital signs domain and have patient-specific numerical ranges. Another domain example is current activity which reports the present activity state of the patient such as resting, sleeping, walking etc. So, this domain has only one context attribute (that is, $n = 1$) and its value set $A_i$ contains discrete values (or categorical values). A context of a domain also can be obtained from long-term observations such as smoking habit of the patient. So, this is a static context and has boolean value.

**Definition 3.3.** The dynamic context state at any given time $t$ from AAL system $j$ is represented by, $C_j^t = \{I_{D_k}^t|1 \leq k \leq N \}$. Where, $N$ is total number of domains in AAL system.
j. \( t \) is single observation time or time duration (\( t = t_s \to t_s + \lambda \)). Also the information from all static contexts is represented as \( I_{D_{Sj}} \). So the overall context state for \( j \)-th system become, \( C_{jt}^t = \{ I_{D_{kj}}^t \cup I_{D_{Sj}} \} \). For long-term observations, \( t \to \infty \) and for numerous AAL systems \( j \to \infty \). Thus the amount of data that need to be processed will be very large and become big data.

An example context state is, Patient is eating in dining and his \((SBP, DBP) = (120, 80)\) and \(HR = 77\) and current room temperature = 28\(^\circ\)C and before that he was resting and less sleep symptom is detected.

**Definition 3.4.** The context space of \( j \)-th AAL system for time span \( T \) is, \( C_{jT} = [C_j]_{t=t_y}^{t_x}, \) where \( T = t_y - t_x = m\lambda \). \( T \) is large amount of time (e.g. 1 month), \( \lambda \) is small time interval (e.g. 1 minute) and \( m \) is number of context states.

**Definition 3.5.** The decision vector for context space \( C_{jT} \) is, \( U_j = (R_j, P_j, Th_j) \). Where \( R_j \) is set of association rules that have a set of correlation probabilities \( P_j \) and threshold values \( Th_j \).

The goal of our model is to
- Generate \( U_j \), after learning from context space \( C_{jT} \)
- Apply \( U_j \) to a new context space \( C_{jT_l} \), where \( T_l > T \). From the observation result generate training set \( Tr_j = \{ C_{j}^T, r_{j}^T \mid T < t_x \leq T_l, t_{x+1} = t_x + \lambda \} \), where \( r_{j}^T \) is the result of context state \( C_{j}^T \) after applying \( U_j \).
- Find the best classifier \( Class_j \) using \( Tr_j \) that also measures the accuracy of \( U_j \).
- Apply \( Class_j \) for another new context space \( C_{jT_m} \) where \( T_m > T_l > T \) and use the classification results for making context-aware decision.

### 3.5 System Functionalities

In this section, the functionalities and algorithms of overall framework are briefly described. The different modules of the whole process is shown in Figure 3.4.

#### 3.5.1 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 3.1 and 3.2). From existing research literature we assumed that such capabilities of context conversions already exist. To make the computation simpler each context attribute value set \( A_i \) is converted to numerical value. Some context attributes already have numeric value (e.g. HR, BP, room temperature). Numerical annotations are used for contexts having nominal value (e.g. activity). The static or historical context that have boolean values (e.g. symptoms) are combined together in a single binary string which results a decimal value (e.g. 001100 converted to 12). So, after such numerical conversion every \( A_i \) has the value set described in Definition 3.1.

### 3.5.2 Context Aggregation

For a single AAL system, after converting all the context attributes to numerical values described above, the context information (as in Definition 3.2) for each of the domain are generated. Then they are converted to a context state (Definition 3.3) which 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, are required. Some attributes have discrete time interval (e.g. BP measured in time interval \( \Delta t \)) and some have time duration (e.g. activity \( x \) starts at \( t_s \) and end at \( t_e \)). To represent everything in a single time(\( t \)) slot a standard time interval \( \lambda \) is chosen. That is, a context state is sampled in \( \lambda \) 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 to a single time slot using the following steps.

- **Case 1**: Domain $D_k$ (e.g. vital signs) where every context attribute (e.g. SBP, DBP, BR) are sampled in a fixed time interval ($\Delta t$) - To find the context attribute value at time $t$ pick the latest value between time $t - \Delta t$ to $t$ and add it to $I^{t}_{D_k}$.

- **Case 2**: Domain $D_k$ (e.g. activity domain) where identified context progress in a time duration ($t = t_s \rightarrow t_e$) - Two separate domains $D_x$ (current activity) and $D_y$ (past activity) are created. During the sampling at time $t$, the ongoing context observation is added in $I^{t}_{D_x}$. If there is an overlap of contexts between 2 observations ($t$ and $t + \lambda$) then the last recent observation is added in $I^{t}_{D_y}$.

- **Case 3**: For the static context domain $D_s$ - a context value is added in $I^{t}_{D_s}$ at time $t$ only when this is requested by the CMS.

Once the above processing are completed for every domain $D_k$ and $I^{t}_{D_k}$ is generated, the next step is to aggregate all these $I^{t}_{D_k}$ in a context state (Definition 3.3). A MapReduce process [84] which runs in multiple clusters inside the CA cloud does this aggregation task for every AAL system. This is described in Algorithm 3.1.

Table 3.5 shows an example dataset that is generated after aggregation step.

### 3.5.3 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 role of the CMS is to detect the trend in the dataset. Some of the patterns are detected using statistical analysis. As example, by summing up the duration of sleeping activity it is possible to summarize, how many hours the user sleeps in a night. Using this statistics the daily mean of sleep hours can be measured say, from the observation of 1 month data. So, for any new data if sleep hours have large deviation from mean, then it is considered as *less sleep* symptom. When any symptom is detected from trend the CMS act as a CP and notifies CA. CA then include this in the $I_{D_s}$ of next context state. In this way, other symptoms (e.g. smoking, weight gain, less exercise) also can be detected.
**Algorithm 3.1.** Aggregate all contexts to a context state

1: **Input**: A set of context information \( I_{D_k} \) for all AAL systems
2: **Output**: Context state \( C_j^t \) for each AAL system \( j \)
3: **Procedure** Mapper()
   4: begin
   5: for each AAL system \( j \) do
   6:   for domain \( \leftarrow 1 \) to \( k \) do
   7:     generate \( I_{D_k}^t \) for time \( t \)
   8:     output(\( key=(j,t), value=I_{D_k}^t \))
   9:   end for
10: if \( I_{D_s} \neq \phi \) then
11:   output(\( key=(j,t), value=I_{D_s} \))
12: end if
13: end for
14: end
15: **Procedure** Reducer(\( key=(j,t), value=\text{set of } I_{D_k}^t \))
16: begin
17: for each AAL system \( j \) do
18:   \( C_j^t \leftarrow \phi \)
19: end for
20: for each \( I_{D_k}^t \) at \( t \) in AAL system \( j \) do
21:   \( C_j^t \leftarrow C_j^t \cup I_{D_k}^t \)
22: end for
23: if \( \text{Exists}(I_{D_s}) \) in AAL system \( j \) then
24:   \( C_j^t \leftarrow C_j^t \cup I_{D_s} \)
25: end if
26: output(\( key=(j,t), value=C_j^t \))
27: end

**3.5.4 Correlations Learning and Association Rule Mining**

This is the knowledge acquisition phase as shown in Figure 3.4 and the major part of learning process of our model. Once \( m \) number of context states are gathered in the cloud storage of the CMS for an AAL system \( j \) (Definition 3.4), it starts this learning process. In this phase, the associations between context attributes are measured. Let after aggregation, a context state \( C_j^t \) contains \( q \) different context attributes which is represented by, \( C_j^t = \{ a_i | i=1,2,...,q \text{ and } a_i \in A_i \} \). The situation space for context state \( C_j^t \) can be described as, \( S_j = \{ a_i^{L_{A_i}} | i=1,2,...,q \text{ and } a_i \in A_i \text{ and } L_{A_i} \text{ is length of value set } A_i \} \). That is, set \( S_j \) has every possible combination of all \( a_i \). When most of \( A_i \) have numerical value range, \( S_j \) become infinitely large set. For this reason, \( S_j \) is reduced to \( S_{m_j} \) by attribute value minimization process which is described
in Algorithm 3.2. Note that, each iteration of Algorithm 3.2 can run in parallel in multiple clusters inside the CMS. This algorithm significantly reduces the situation space that is used for rule generation. It generates the minimized value set of vital signs using statistical features.

**Algorithm 3.2.** Attribute value minimization

1: **Input:** \(C_j^T\) and \(S_j\) for all AAL system \(j\)
2: **Output:** \(S_{mj}\) for all AAL system \(j\)
3: for each AAL system \(j\) in parallel do
   4: \(S_{mj} \leftarrow \phi\)
   5: \(B_j\) is the set of context attributes in \(C_j^T\)
   6: \(B'_j\) is the subset of \(B_j\) those are vital for users and have numerical range (e.g. \(A_i = [v_{i_{\text{max}}}, v_{i_{\text{min}}}]\))
   7: \(S'_j\) is the subset of \(S_j\) that does not contain attributes in \(B'_j\)
   8: for each situation \(s\) in \(S'_j\) in parallel do
      9: find min, mean, max and standard deviation of each attribute in \(B'_j\) from dataset \(C_j^T\)
      10: generate minimized attribute value set \(b\) of the form \(A_i = \{[a_{i_{\text{min}}}, a_{i_{\text{mean}}}, a_{i_{\text{stddev}}}], [a_{i_{\text{mean}}}, a_{i_{\text{stddev}}}], [a_{i_{\text{mean}}}, a_{i_{\text{stddev}}}, a_{i_{\text{max}}}]\}\) for all attribute in \(B'_j\). So \(L_{Ai}\) is minimized from \([v_{i_{\text{max}}}, v_{i_{\text{min}}}]\) to 3
      11: Add \((b,s)\) in \(S_{mj}\)
   12: end for
13: output \(S_{mj}\)
14: end for

To find the correlations \((P_j)\) among context attributes and the threshold values \((Th_j)\) of vital parameters, Apriori-based \([144]\) association rule mining approach is adopted. We used MapReduce version of Apriori \([84]\) for our model which is a more efficient technique for mining association rules from big data that utilizes the computational resource of multiple dedicated clusters of distributed cloud model. In typical MapReduce process there is a Map function that process a key-value pair to generate a set of intermediate key-value pairs. The reduce function then merges all intermediate values associated with same intermediate keys \([145]\). We have also used MapReduce technique for context aggregation which is described in Algorithm 3.1. Like in \([84]\) 3 MapReduce jobs are used. One of them is described in Algorithm 3.1. The second MapReduce process finds the support count for a specified threshold range of different vital signs in different situations \(s\) and mine frequently occurred correlated events using \(\text{minSupp}\). The third MapReduce process finds the association rules using \(\text{minConf}\). In our model, the support count of \(x\) is defined as, \(\sigma(x) = \{v_i|v_i \in C_j^T, x \subseteq C_i\}\). Here \(x\) has the from \(a_i = v_i\) or \(a_i \epsilon [v_y, v_z]\). If \(C_j^T\) contains \(m\) context states then, support \((x) = \frac{\sigma(x)}{m}\) and confidence of the rule \(x \Rightarrow y\) is, confidence \((x \Rightarrow y) = \frac{\sigma(x \Rightarrow y)}{\sigma(x)}\). The strength of association rule \(x \Rightarrow y\) is measured
by this support and confidence value. Support determine, how often the rule is applicable to the dataset and confidence determine the strength of correlation between x and y.

There are two important pre-defined parameters namely minSup and minConf. In 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 $U_j = (R_j, P_j, Th_j)$. The overall process is described in Algorithm 3.3. Note that, 2 MapReduce processes run inside the algorithm block for calculating support and confidence as described in Algorithm 5 and 6 of [84] and filter out the infrequent situations. 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 (described in section 3.7).

Algorithm 3.3. Association Rule Mining

1: **Input:** $C_j^T$ and $Sm_j$ for all AAL system $j$
2: **Output:** Decision vector $U_j = (R_j, P_j, Th_j)$ for all AAL system $j$
3: for each AAL system $j$ in parallel do
4: $R_j \leftarrow \phi$, $P_j \leftarrow \phi$, $Th_j \leftarrow \phi$
5: $q \leftarrow$ number of context attributes in $C_j^T$
6: for each $s$ in $Sm_j$, find $\sigma(s)$ in $C_j^T$ using MapReduce and prune $Sm_j$ to $Sp_j$ by eliminating every $s$ that has $\sigma(s) < minSup$
7: for each combination $(x, y)$ in $Sp_j$, find association rule $x \Rightarrow y$ using MapReduce over temporal support of $x$ and $(x, y)$. Here, $x$ are vital sign attributes having the form $a_i \epsilon [v_y, v_z]$ and $y$ are other context attributes
8: calculate confidence($x \Rightarrow y$)
9: if confidence($x \Rightarrow y$) ≥ minConf then
10: Add $[v_y, v_z]$ in $Th_j$
11: Add confidence($x \Rightarrow y$) in $P_j$
12: Add association rule $x \Rightarrow y$ in $R_j$
13: end if
14: return $U_j = (R_j, P_j, Th_j)$
15: end for

All these learning processes run inside the CMS cloud. When $U_j$ for an AAL system is obtained, the CMS sends this to the PCS of that AAL system where the rules and thresholds are stored.
3.5.5 Learning using the Association Rules

After discovering the knowledge of every AAL system $j$, the next step is to verify the validity of this learning process using a new set of big data. When a new context state $C_j^t$ arrives in the CMS cloud, it classifies the state according to the rules and thresholds found in decision vector $U_j$. In conflicting case the CMS uses the rule that has the highest confidence value in $P_j$.

Let, the classification result of context state $C_j^t$ is $r_j^t$ where $r_j^t \in Result$. The set, Result contains nominal value such as normal, warning, alert, emergency. After classifying $C_j^t$ using $U_j$ the data instance $C_j^t, r_j^t$ is stored in the CMS cloud. When large number of such instances are available, the CMS uses that big dataset for building classifier model. The classifier is used for future classification.

3.5.6 Data Mining

The dataset generated in previous phase are 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 [132] 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.

3.5.7 Context-aware Decision Support

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 2 or more context attributes are abnormal or anyone is extremely abnormal then notify to emergency.
A context state will not always be the same. It can change due to change of a patient’s condition, 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 have sufficient storage and processing capability to run the whole process iteratively.

3.6 Use case Implementation

In accordance to the BDCaM functional components described in the previous sections a case study is implemented to evaluate our algorithms. The objectives of this implementation are, (i) discover knowledge of BP and HR change on different situations [134] for different patients; (ii) find association rules for specific patient situations using distributed cloud model, and (iii) classify unknown situations based on learned model.

3.6.1 Description of use case

The continuous BP level of a patient is determined by examining systolic BP (SBP) and diastolic BP (DBP) value in mmHg using body worn BP sensor, and HR is measured in bpm using ECG sensor [27, 77]. Abnormal BP or HR is difficult to diagnose and treat based on measuring them once or twice in a day. The variations in BP and HR occurs due to change in ambient temperature, physical activity, noise, sleep, fatigue, stress etc. (e.g. HR high for exercise, BP high for eating) [146, 147]. The variations are also common from patient to patient for other factors such as disease history and family profile (e.g. BP always high for a hypertensive patient). The abnormal variations are not always dangerous in practical situation. Therefore, if a patient situation is classified using only the generalized threshold value, then for most cases this will trigger false alarms to the receiver. Thus, it is important to consider the correlations of other contexts for making final clinical decisions. The goal of this experiment is to detect those abnormal cases which are not actually that critical for raising an alarm. The learning process utilizes large historical data of multiple patients and generates personalized knowledge for each patient individually. The knowledge generation and abnormality detection cases run in cloud platform. For evaluating cloud platforms we have used Google App Engine [42] and Amazon web services (AWS) [40].

Some general medical rules [148, 149] for identification of patient situation based on these two vital signs (HR and BP) are described in Table 3.1.
Table 3.1: General medical rules to identify different diseases related to the variation in BP and HR

<table>
<thead>
<tr>
<th>Category</th>
<th>HR (bpm)</th>
<th>SBP (mmHg)</th>
<th>DBP (mmHg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>60-100</td>
<td>90-119</td>
<td>60-79</td>
</tr>
<tr>
<td>Hypotension</td>
<td>-</td>
<td>≤ 90</td>
<td>≤ 60</td>
</tr>
<tr>
<td>Pre-Hypertension</td>
<td>-</td>
<td>120-139</td>
<td>80-89</td>
</tr>
<tr>
<td>Hypertension</td>
<td>-</td>
<td>≥ 140</td>
<td>≥ 90</td>
</tr>
<tr>
<td>Bradycardia</td>
<td>≤ 60</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tachycardia</td>
<td>≥ 100</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

3.6.2 Synthetic data generation

To validate a learning model that uses a large amount of data, it is important to have that data which are very similar to real data. To the best of our knowledge, no real-life dataset that contain daily monitoring data of a patient for a relatively long period are publicly available. Therefore, the first step is to generate data based on continuous BP and HR monitoring that is well established in medical practice. So artificial data are generated using some real medical observations. The dataset of continuous BP and HR monitoring by Faini [150] and Parati [151] are used for this purpose. The dataset from Physionet MIMIC-II database [152] is also used, because this contains a large number of samples of multiple vital signs (including SBP, DBP and HR). Some of the MIMIC-II records contain more than 24 hours ICU patient data. From these referred works the distributions of SBP, DBP and HR in a day for different patient category (as in Table 3.1) during different activities are measured.

In analysis of the BP and HR data distribution, a normal distribution is identified when data is limited to a specific activity (e.g. BP is higher when patient is awake and lower when he/she is in sleep). The use of normal distribution in synthetic data generation is also reliable for model validation because in that case the generated data are nearly similar to real data. The advantage of using such distribution for biomedical data analysis is also proven in some previous works [153, 154]. Therefore, using normal distributions SBP, DBP and HR dataset of 365 days (1 year) are generated. Then, the values of those distributions are increased or decreased with random probability. The values are also varied by assuming a current activity and last activity (e.g. resting, sleeping) that a user can do during that period in a day (e.g. sleeping at 2 a.m.). The activity sequences are placed in a separate array (as activity domain) which implies the ongoing activity of the user in that particular time. Some random entries are again varied by assuming correlation with room temperature, medication taken and different
symptoms in the same time-series of SBP, DBP and HR. So the correlation of generated data are preserved with other contexts and disease history (e.g. mean BP is high for hypertensive patient). To make it more realistic, some data are again modified with random uncertain probability (e.g. SBP/DBP/HR value high/low without any effect of other contexts which can lead some medical emergency). An upper and lower bound is assumed within physiological bound of each vital sign for each patient considered for the experiment. By doing this we avoid the extreme values which are not practical in real case and thus the produced data become very close to realistic data. The dataset is generated using MATLAB and stored in multiple files. Three types of patients with different BP levels (‘disease history’ context) is considered. The patient classifications are shown in Table 3.2.

Table 3.2: Different patient categories considered in the experiment and their average values of vital signs

<table>
<thead>
<tr>
<th>Patient</th>
<th>Category</th>
<th>Avg. HR</th>
<th>Avg. SBP</th>
<th>Avg. DBP</th>
<th>BP Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Hypertension</td>
<td>74</td>
<td>149</td>
<td>91</td>
<td>generally high</td>
</tr>
<tr>
<td>P2</td>
<td>Normal</td>
<td>76</td>
<td>103</td>
<td>69</td>
<td>generally normal</td>
</tr>
<tr>
<td>P3</td>
<td>Hypotension</td>
<td>65</td>
<td>79</td>
<td>64</td>
<td>generally low</td>
</tr>
</tbody>
</table>

**3.6.3 Cloud evaluation**

As described in the previous step, synthetic data of 1 year for 3 patients are generated using 15 minutes sampling interval that results 35040 samples per patient. So, the generated data satisfy variety property. The dataset will be very large if millions of patients are considered (instead of 3) and thus it satisfies volume property. A java worker thread is implemented to simulate an AAL system. Multiple threads run in parallel that simulates the scenario that, many AAL systems simultaneously send data to the cloud. In our simulated environment 1 min is normalized to 1 ms. To simulate the velocity of big data, the java worker thread read one sample of synthetic data from local files in every 15 ms and sends it to the Google Cloud Storage (GCS) [42]. The data is written in a file of GCS bucket. 3 separate threads run for 3 patients in our simulated model and do the same job in parallel. The use of MapReduce techniques in our system for generating rules also satisfies the velocity criteria, as MapReduce has the streaming paradigm that deals with velocity.

To test the performance of our algorithms in cloud model and to obtain the association rules for the generated dataset the following 3 modes are created. The tools that are used to create these modes are Matlab parallel computing toolbox [155], and Amazon Elastic MapReduce.
The association rule generation process for 3 patients are run in all 3 modes using same dataset and parameter settings. The obtained results are compared and discussed in section 3.7.

- Standalone mode: Where the algorithms of association rule generation run as a sequential process on a single core.
- Pseudo-distributed mode: Where the algorithms run as a parallel process on 2 to 4 cores. Each of the core has same processing capability.
- Cloud mode: Where the algorithms run on public cloud using Amazon Elastic Mapreduce with 1 master node and 2 slave nodes.

### 3.6.4 Implementation

Using the methodology described in section 3.5, the correlations of BP and HR with other context attributes (e.g. activity, last activity, symptoms) are measured to predict the accurate state for a context situation. The domains and the context attributes along with their value set considered in the experiment are presented in Table 3.3. Table 3.4 describes some static context attributes (including their values) which are detected using trend analysis for the patients of such domains.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Name</th>
<th>Context Attributes</th>
<th>Name/Type</th>
<th>Range/Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>Vital Signs</td>
<td>HR</td>
<td>Numeric</td>
<td>[30,120]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Systolic BP (SBP)</td>
<td>Numeric</td>
<td>[50,230]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diastolic BP (DBP)</td>
<td>Numeric</td>
<td>[30,140]</td>
</tr>
<tr>
<td>$D_2$</td>
<td>Activity</td>
<td>D2: Current Activity, D3: Last Activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>resting</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sleeping</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>walking</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eating</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exercising</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>household</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>$D_4$</td>
<td>Ambient Conditions</td>
<td>Room Temperature</td>
<td>Normal</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hot</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cold</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>$D_{S1}$</td>
<td>Medication</td>
<td>Medication Taken</td>
<td>Boolean</td>
<td>0 or 1</td>
</tr>
<tr>
<td>$D_{S2}$</td>
<td>Specific Symptoms</td>
<td>Symptoms</td>
<td>boolean</td>
<td>[0,63]</td>
</tr>
</tbody>
</table>

After data aggregation and trend analysis steps as described in section 3.5, the context space with 8 context attributes is obtained. A small part of the generated context space $C_j^T$
Table 3.4: Symptoms that are considered in the experiment

<table>
<thead>
<tr>
<th>Type</th>
<th>Major Symptoms that affect</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypertension</td>
<td>Smoking, Drinking alcohol, Overweight, Salty food, Less exercise, Less sleep.</td>
<td>6-bit binary (value: 0 to 63)</td>
</tr>
<tr>
<td>Hypotension</td>
<td>Low heartbeat, Fast breathing, Oversleep, Vomiting, Feeling week, Dizziness.</td>
<td>6-bit binary (value: 0 to 63)</td>
</tr>
</tbody>
</table>

is shown in Table 3.5. This data sample is from a hypertensive patient who’s average BP is always high.

Table 3.5: Sample aggregated data of Patient P1 for rule mining

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

For each of 3 patient, 3 context space like Table 3.5 of one year is generated and Algorithm 3.2 and 3.3 are applied. Many association rules for each of 3 dataset along with corresponding thresholds and probability of correlations are obtained. The number of rules for each of the case are evaluated by varying \( minSupp \) values which is shown in Figure 3.5.

![Figure 3.5: The change in no of rules with minSupp value](image_url)
We observed that, with low $\text{minSupp}$ value, number of rules are highest for hypertensive patient and lowest for normal. We picked $\text{minSupp} = 0.55$. Then we got corresponding number of rules for each of the patients and shown in Figure 3.5.

Once the rule generation step is completed another new context space (that is a new dataset) of 1 year is created using the generated rules for each of the patient. During this new dataset generation following classes (Table 3.6) are considered for describing a context-aware situation of a patient.

Table 3.6: Class attributes used for situation classification

<table>
<thead>
<tr>
<th>Class</th>
<th>Value</th>
<th>Classification Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>Satisfy Rule $B \Rightarrow A[v_x, v_y]$ that is HR, SBP and DBP all value are in expected range of threshold in current context</td>
</tr>
<tr>
<td>Warning</td>
<td>1</td>
<td>Any one of HR, SBP or DBP is not in expected range/ Medication not taken / Any symptom is true</td>
</tr>
<tr>
<td>Alert</td>
<td>2</td>
<td>More than 1 of HR, SBP or DBP is not in expected range and medication not taken or any symptom is true</td>
</tr>
<tr>
<td>Emergency</td>
<td>3</td>
<td>All vital signs are out of expected threshold in any context state</td>
</tr>
</tbody>
</table>

After completion of new data generation, different data mining techniques using WEKA [156, 157] machine learning software are applied and the accuracies are obtained. Four Amazon EC2 instances (type m3.medium) [40] are created with same processing capability. Using 10-fold cross validation the classification process is run first on a single EC2 instance and then on four EC2 instances using weka-parallel [157]. As the association rules are obtained based on some probabilistic measurement (e.g. confidence of a rule), so another learning process with new dataset is applied using data mining. One reason for applying this step is to verify the accuracies of generated rules. Another reason is to find an optimized and faster classifier model for a specific patient rather than storing every rule for every situation along with threshold values. Therefore, this simplification makes classification task straightforward. The obtained results are discussed in the following section.

3.7 Results and Discussion

To evaluate the sensibility, we compare the situation classification of our model with generalized clinical classification based on medical rules [148, 149]. Table 3.7 shows the result of such comparison. In general the rule is, when SBP and DBP values go above/below certain threshold
Results and Discussion

(e.g. above 135 and 85 for hypertension) then the situation is classified as abnormal. Emergency state is not identifiable; it is the doctor’s responsibility to manually analyse the value and then make decision. The term general medical rule is adopted because this is the current manner with which traditional context-aware [56] systems process vital signs data. In contrast, our system learns these patient-specific thresholds and quickly adapts to new changes.

Table 3.7: Comparison of BDCaM model generated rule-based classifications with generic rule-based classification

<table>
<thead>
<tr>
<th>Patient</th>
<th>Total Data</th>
<th>Generic Rules</th>
<th>BDCaM Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Abnormal</td>
<td>Normal</td>
</tr>
<tr>
<td>P1</td>
<td>34301</td>
<td>1</td>
<td>34300</td>
</tr>
<tr>
<td></td>
<td>24518</td>
<td>8584</td>
<td>1116</td>
</tr>
<tr>
<td></td>
<td>1116</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>34395</td>
<td>179</td>
<td>34216</td>
</tr>
<tr>
<td></td>
<td>24911</td>
<td>7980</td>
<td>1360</td>
</tr>
<tr>
<td></td>
<td>1360</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>25274</td>
<td>1</td>
<td>25273</td>
</tr>
<tr>
<td></td>
<td>17221</td>
<td>6945</td>
<td>1023</td>
</tr>
<tr>
<td></td>
<td>1023</td>
<td>85</td>
<td></td>
</tr>
</tbody>
</table>

From the result of Table 3.7 we can summarize that, the BDCaM model performs extremely well to find a true normal situation of a patient where a system without considering context can only identify few such cases. Thus, it can reduce the generation of false alerts at receiver’s side. Since no similar work was found in the literature to compare our results, it is compared with generalized medical observations. In some remote monitoring system the thresholds values are manually adjusted by doctors and in this way they try to minimize false alerts. But in our system no manual interventions are required.

As described in section 3.6.3 to differentiate between centralized system and distributed cloud environment, the standalone and pseudo-distributed modes are evaluated. The execution time from context aggregation to association rule generation for these 2 modes are presented in Table 3.8. As obvious, the best result is obtained for the highest number of cores.

Table 3.8: Comparison of association rule generation time using single and multi-core processing on same machine

<table>
<thead>
<tr>
<th></th>
<th>1 core</th>
<th>2 cores</th>
<th>4 cores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.68 sec</td>
<td>1.78 sec</td>
<td>0.415 sec</td>
</tr>
</tbody>
</table>

To test our MapReduce algorithms in real cloud environment we have used Amazon EMR [40] and evaluated them using 1 master node and 2 slave nodes. The data are stored in Amazon S3 buckets [40]. In EMR the process took nearly 2 minutes. There are some overhead time of data partitioning and collecting so it took much longer time in EMR than pseudo-distributed mode. After data partitioning, it is observed that total 11 tasks were created for dataset size 35040 where 8 of them were map tasks and the rest 3 were reduce tasks. It has been already
proven that the Amazon EMR performs the best for very large dataset; for example the 100 Genomes project [158] that contains 200TB of data. Since we have limitations to generate and upload such large data for our experiment so we tested only with our generated data which are relatively small. From the developed proof of concept and experiments it can be concluded that, our algorithms will work fast in the real cloud environment when large number of samples such as 1 billions and many slave machines (e.g. 100) are used.

As mentioned in section 3.6.4, the new context space is tested with different classifiers using 10-fold cross validation. The classification results obtained for 3 patients are presented in Table 3.9. The accuracy of J48 decision tree classifier is very high in comparison to others because the training data are generated using association rules. To guarantee unbiased result we tested the dataset with other classifiers. As shown in Table 3.9, Bayes Network (BN) and Radial Basis Function (RBF) does not have good accuracy. The performance of Multi Layer Perceptron (MLP) is better other than J48 and decision table.

Table 3.9: Anomaly detection accuracy and false positive rate for 3 different types of patient over 1 year data

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Classification Accuracy</th>
<th>Avg. False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Patient P1</td>
<td>P2</td>
</tr>
<tr>
<td>J48 Decision Tree (J48)</td>
<td>99.14%</td>
<td>99.78%</td>
</tr>
<tr>
<td>Decision Table (Dtable)</td>
<td>95.91%</td>
<td>97.08%</td>
</tr>
<tr>
<td>Multilayer Perceptron (MLP)</td>
<td>91.46%</td>
<td>95.54%</td>
</tr>
<tr>
<td>Bayes Network (BN)</td>
<td>86.68%</td>
<td>95.11%</td>
</tr>
<tr>
<td>Radial Basis Function (RBF)</td>
<td>81.52%</td>
<td>83.95%</td>
</tr>
</tbody>
</table>

Figure 3.6: Classification time for different classifiers using 1000 to 34000 context states for hypertensive patient.

We measured the time for building the classifiers using first on a single Amazon EC2 instance and then on four instances. Recently it is proven that such classification can be done even faster using distributed weka hadoop [159] with millions of samples. In our case
trivially the classification processes in multiple instances took shorter time. To evaluate the performance of the model, the time of building each classifier model is measured with small to large number of samples using weka-parallel [157]. The result for hypertensive patient data is shown in Figure 3.6. Similar behaviours are identified for normal and hypotensive patient. In Figure 3.6(a) for single machine, we found that when the sample size becomes larger the execution time tends to reach a constant value for MLP and J48. Also if the classification process is distributed in multiple machines as in cloud (in Figure 3.6(b)) then we observe that all the classifiers run much faster with same sample size (e.g. execution time is less than 22 ms for all classifiers in our case) and can make quicker decision.

3.8 Conclusion

In this chapter, we have presented BDCaM, a generalized framework for personalized healthcare, which leverage the advantages of context-aware computing, remote-monitoring, cloud computing, machine learning and big data. Our solution provides a systematic approach to support the fast-growing communities of people with chronic illness who live alone and require assisted care. The model also simplifies the tasks of healthcare professionals by not swamping them with false alerts. The system can accurately distinguish emergencies from normal conditions. The data used to validate the model are obtained via artificial data generation based on data derived from real patients, preserving the correlation of patient’s vital signs with different activities and symptoms. The stronger relationship between vital signs and contextual information will make the generated data more consistent and the model will be more accurate for validation. The experimental evaluation of our system in cloud model for patients having different HR and BP levels has demonstrated that the system can predict correct abnormal conditions in a patient with great accuracy and within a short time when it is properly trained with large samples.

3.9 Epilogue

The contributions of this chapter partly answer the second research question. The BDCaM model utilizes context correlations for detecting patient-specific abnormalities in ongoing situation but ignores functionalities related to future change detection. In the next chapter we include such functionalities for future behaviour predictions.
Chapter 4

Long-term Change Detection and Abnormality Prediction

The techniques we developed so far can detect abnormal changes in an ongoing context using generalized or personalized rules but there is no option to predict future behaviour. This and subsequent chapters in this thesis will focus on future abnormalities and behaviour predictions. The accurate estimation of future change is extremely vital feature context-aware monitoring.

As discussed in Chapter 1, modern remote monitoring technologies utilize body sensors [1] which continuously monitor different physiological parameters of the user. The most recently observed information is sent to the healthcare professionals or doctors. Doctors observe the current medical condition and activities of the user and provide a diagnosis based on the information provided by the monitoring system. However, that system can only notify at the stage when any chronic disease attains a mature state in the patient. That is, these systems can not detect a sudden attack of chronic disease. Some systems can detect certain chronic diseases such as cardiovascular illness or, diabetes, but they only make a decision based on a single context observation (e.g. ECG, blood sugar) [58]. This kind of system suffers from high false alert rates because they do not consider the effect of other related contexts (e.g. user activity, time of blood sugar measurement) and context histories. If a single context has any growing trend of abnormality then in most cases the final prediction can not be made without checking the status of other contexts (e.g. an increasing trend of heart rate is normal when a user exercises). In summary, existing context-aware monitoring technologies are insufficient and not intelligent enough to predict future change. Considering these limitations and in relation to research question 2, a framework for future change detection is presented in this
In this chapter, we described a Hidden Markov Model (HMM) based approach for detecting abnormalities in daily activities, a process of identifying irregularity in routine behaviours from statistical histories and an exponential smoothing technique to predict future changes in various vital signs. The outcomes of these different models are then fused using a fuzzy rule-based model for making the final guess and sending an accurate context-aware alert to the health-care service providers. According to BDCaM model described in Chapter 3 context space of our context-aware system can be divided into different context domains. The best knowledge from each such domain is then extracted using well-known learning techniques. The technique is chosen based on the distribution of targeted contexts. For example, in learning what is a normal or abnormal pattern of daily activities, we have used the Hidden Markov Model (HMM) because it is a simple and efficient model for learning sequential data having temporal dependency [86, 160]. The sequence of activities varies according to the individual user and the duration of each activity. For this reason, we picked HMM which is a good temporal probabilistic model for handling such uncertainties. Moreover, it was proven that HMM performs better than other learning models in such domain [161, 162, 163]. Using HMM it is possible to detect future states from current observations as well as the sequence of states from an observed sequence.

The rest of this chapter is organized as follows. Section 4.1 describes motivations and contributions of this chapter. Section 4.2 summarizes the related works in the area. The system methodology along with architecture is presented in Section 4.3. The case studies along with the experimental evaluations and the results of the proposed methodology are illustrated in Section 4.4. Finally, Section 4.5 summarizes this chapter.

4.1 Motivations and Contributions

4.1.1 Motivations

The primary goals of our context-aware system are to learn the behavioural patterns of a patient using long-term context history, to predict future abnormalities based on gathered knowledge, to detect disease symptoms at an early stage and to help healthcare professionals in long-term diagnosis process. To accomplish these objectives we have the following motivations.

- Develop a generic change detection framework for a context-aware assisted living system.

The framework works as an intelligent module of cloud-based context-aware middleware
that not only detects abnormalities in what is observed but also can predict what is going to be observed using prior knowledge and observations.

- Identify and develop suitable pattern recognition models for different context domains. The selection of models depends on the distribution of context attributes in the content domain. The models will be strong enough to identify future changes in any context situation.

- Reduce the rate of wrong predictions of context situations as well as the number of false alerts to the monitoring systems by fusing the outcome knowledge of every trained model. All models are trained continuously to make accurate predictions.

4.1.2 Contributions

The main contributions of this work are as follows.

- We developed a novel change detection framework on top of generalized context-aware framework for AAL users. The framework is capable of predicting future behaviours, physiological states and disease symptoms of the user. The frameworks build the personalized knowledge-base of the user using machine learning methods, statistical analysis and mathematical prediction.

- We developed a Hidden Markov Model (HMM) based technique, which is a completely new process for detecting abnormalities in the sequence of activities of an AAL user. The developed model is also capable of finding irregular activities in a given sequence. The model is trained with a large number of samples containing normal and abnormal behaviours. We also developed a statistical method for detecting abnormalities in the routine behaviour of the user.

- We developed a forecasting technique using Holt’s liner trend method for predicting the increasing or decreasing trend of different physiological parameter’s of the user. The detected trend can predict the growth of several chronic diseases of the system user.

- We built a fuzzy fusion process that combines the outcomes of every well-trained model to reach final conclusion for a given situation. The process is demonstrated and validated by implementing case studies and by evaluating the performance of the individual as well as the overall framework.
4.2 Related Works

With the advance of wearable and wireless sensor technologies, real-time monitoring of patient’s daily behaviours and physiological parameters have become more common. In an existing, remote healthcare model, the failure to perceive change in health or behaviour can create a high risk for a patient whose health condition changes rapidly, sometimes with multiple contextual effects. For understanding future changes and abnormal behaviour, a system should consider previous actions and current contextual events. Thus an approach that can understand a patient’s behavioural deviation from usual lifestyle and presence of certain disease symptoms using pattern recognition and mathematical models would be of great assistance to healthcare professionals.

In the area of context-aware monitoring using wireless sensor technology most of works to date have mainly focused on understanding primitive activities or behaviours [45, 50, 69, 70]. These systems only can detect some activities and are far from modelling daily life behaviour. Machine learning techniques are broadly used to detect daily behaviour, abnormality in regular activities and detection of cardiovascular disease syndromes. Examples of such learning model includes HMM [164, 165, 166], support vector machine [167, 168], Naive Bayes classifier [169] and neural network [170]. These systems can detect abnormalities in a single context domain such as only in body parameters or daily behaviour but cannot correlate health status with daily activities. Some researchers proposed predicting some specific disease symptoms (e.g. asthma, stress, hypertension) by observing changes in other related contexts [65, 66, 73]. These systems only consider a few types of contexts and try to relate those contexts to identify the true abnormality. But they are still only able to predict abnormality in a current situation. There are also many research works [58, 171, 172] for prediction of cardiovascular diseases such as tachycardia, bradycardia and other chronic disease such as diabetes and hypoxia. But they only focused on the detection of abnormal changes in biosignals and physiological parameters [46]. In summary, there has been no attempt to develop an integrated system that detects change in multiple contextual behaviours and predicts the final outcome by combining the information.

4.3 Methodology

In this section, we first describe the overall system architecture of the proposed technique. We then briefly discuss different models which we evaluated for different context domains.
4.3.1 System Description

The overall system description including the background and an example scenario is presented in the following sections.

4.3.1.1 Background

This work has been done as part of evaluation for the cloud-oriented context-aware middleware for ambient assisted living (CoCaMAAL) model, which is described in Chapter 2. CoCaMAAL is a context-aware framework where a distributed context-aware middleware manages and processes a large amount of context information of many assisted living system users using the scalable power of cloud computing, and delivers appropriate context-aware services based on the observed condition. The cloud architecture of the CoCaMAAL model is capable of storing large amounts of personalized context history of every user inside its cloud repositories. In this chapter, we briefly describe pattern recognition models to train our system using those large amounts of context histories and so the system can predict any abnormality or change by fusing all the information in a single context model.

According to the CoCaMAAL model [81] an assisted living system consists of a target user with many wearable sensors which periodically send different vital signs data of the user to the cloud servers. The system also contains many ambient sensors and software services which report different environmental conditions and activities of the user. All the low level sensor data are converted to high level contexts inside the CoCaMAAL architecture. In this chapter, we propose that those high level contexts be stored inside a cloud database along with timestamps. So, we will have daily activity logs of a single patient. The information in a daily log for the individual patient whom we considered in our model is described in Table 4.1.

Table 4.1: Illustration of patient logs for our system

<table>
<thead>
<tr>
<th>Category</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity log</td>
<td>Start time, end time and type of activity</td>
</tr>
<tr>
<td>Location information</td>
<td>Start time, end time, and location</td>
</tr>
<tr>
<td>Lifestyle pattern</td>
<td>Sleep time, wake up time, toilet going frequency, breakfast time etc.</td>
</tr>
<tr>
<td>Vital signs</td>
<td>Sampling time and value of vital sign (e.g. blood pressure, heart rate, temperature)</td>
</tr>
</tbody>
</table>
4.3.1.2 A motivating scenario

In this section, we describe a motivating real-life scenario for our model. Steve is aged 67 years and is alone. He has diabetes and high blood pressure. He likes to wake up around 7 a.m. and after toileting he takes Benazepril medicine for his high blood pressure. He generally breakfasts about 9. Then he has to do recommended 30 minutes exercises around 10 a.m. He takes his lunch between 1:00 and 2:00 p.m. after which he keeps himself busy at household tasks (e.g. works at his computer, phone-calls, reading, cleaning and preparing dinner). He dines about 7:30 p.m. He has to take insulin before dinner. Afterwards he generally watches television or reads. He goes to sleep about 11 p.m. On average, he generally visits the toilet 15 times a day. He can rest any time during his different daily activities. His various vital signs (e.g. blood pressure, blood glucose, heart rate) are monitored continuously using wearable sensors.

A situation is said to be anomalous when any behavioural change (e.g. late in waking up, delay in lunch, high increment in BP) occurs. The goal of our system is to learn the daily behaviour along with the health status of a user of AAL system, like Steve, and to detect any current or future change in that behaviour by aggregating all the information. The learning process will be adaptive and individualized. That is, the learning methods of our system will continuously discover the personalized knowledge of each user (e.g. BP levels during different activities) and will enhance that knowledge when any change occurs. We also want to suppress the rate of false predictions by combining the knowledge of every entity involved in such changes which we named as context in our design.

4.3.1.3 Proposed fusion-based architecture

To detect abnormal changes in any system, the first step is to build a systematic model from normal observations. Later, the normal pattern is compared with new observations and the deviation is estimated. A significant variation indicates a change. The overall system architecture for the proposed change detection technique is shown in Figure 4.1.

In the AAL system, context means any high level information such as ongoing or past activity of the user, location of the user, time of the activity, varying health status of the user and environmental conditions of the system. For simplicity, we divided the context space of our model into different context domains (e.g. daily activity, location routine behaviour, vital signs). Each of the context domain $D_j$ consists of a set of context attributes $a_i$ where each $a_i$ has a value $v_i$. $v_i$ is obtained from a set $A_i$. $A_i$ can have a categorical (e.g. user activity such as sleeping, resting) or a range of values (e.g. heart rate range 50 to 110 bpm). At any given
time $t$ each of the domain contains the context information $I_{Dj}$ of that domain. Each $I_{Dj}$ is represented by a set of attribute-value pairs. If each context domain is trained by a set of past context information (from time $t - T'$ to $t$) using a proper machine learning model so that the model can: (i) find abnormality in $I_{Dj}$ at current time $t$; (ii) identify which context attributes or attribute values are causing abnormality, and (iii) predict a future situation from time $t$ to $t + T$. The set of past context information (from time $t - T'$ to $t$) is called context space in our model [85].

In the suggested approach, different context domains are trained using different learning methodologies. The learning algorithms are selected based on the characteristics of context information of the domain. Finally, the predicted outcomes of all domains are fused and correlated using fuzzy model to make eventual context-aware decisions. Each components of the architecture is briefly described in the following subsections.

### 4.3.2 Change detection in activity and location patterns using HMM

The first context domain of our context-aware model is user activity and location domain. The goal of the model is to detect the consequence a change occurs in the sequence of daily activities and also predict an observation sequence based on the current situation. The model is described below.
4.3.2.1 Data description

Each activity log dataset of our design has 3 attributes - start-time, end-time and high level description of the activity. Similarly, the location information log also has 3 attributes - the time when the subject entered into the location, end time of exiting the location and name of the location. Examples of high level activity are sleeping, resting, eating, exercising and, toileting; and examples of location are kitchen, bathroom and, bedroom. Here we will only describe the model design for activity patterns because a similar approach was used for the location transition pattern.

The large activity log dataset is first separated by date-wise to isolate a sequence of daily activities. The end-time of one activity is the start time of the next. So, for each day we will have an activity dataset as shown in Table 4.2. Each activity has a temporal duration and there is a temporal relation between two activities.

Table 4.2: Daily activity log. Where each $x(t_j) \in X$, $X = \{x_1, x_2, \ldots, x_N\}$, $N$ is the number of activities considered in the design. $t_{k+1} - t_0 = 24$ hours

<table>
<thead>
<tr>
<th>Start time</th>
<th>End time</th>
<th>activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0$</td>
<td>$t_1$</td>
<td>$x(t_0)$</td>
</tr>
<tr>
<td>$t_1$</td>
<td>$t_2$</td>
<td>$x(t_1)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$t_{j-1}$</td>
<td>$t_j$</td>
<td>$x(t_{j-1})$</td>
</tr>
<tr>
<td>$t_j$</td>
<td>$t_{j+1}$</td>
<td>$x(t_j)$</td>
</tr>
<tr>
<td>$t_{j+1}$</td>
<td>$t_{j+2}$</td>
<td>$x(t_{j+1})$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$t_k$</td>
<td>$t_{k+1}$</td>
<td>$x(t_k)$</td>
</tr>
</tbody>
</table>

4.3.2.2 Modelling observable Markov Model

From the above observations, we can conclude that the daily routine of the user is a sequence of activities that forms a first order Markov chain as in Figure 4.2. The goal here is to develop a model based on the samples of normal activity sequences. Then we can determine the likelihood of any incoming observation sequence that could have been produced by the model. If the new observation sequence is not likely to match the normal pattern then we can detect the presence of an abnormal pattern.

We first build an observable Markov model where each of the state of the model corresponds to an activity of the Markov process. An observable Markov model [86] with $N$ states $S_i$ is characterized by $\lambda = (A, \pi)$ where:
Methodology

Figure 4.2: Markov process for activity sequences.

- $\pi$ is the vector of initial state probabilities $\pi_i$ ($i = 1, 2, \ldots , N$) and $\pi_i = Pr[x(t_0) = S_i]$.
- $A$ is the matrix of state-transition probabilities, $a_{ij}$ ($i = 1, 2, \ldots , N; j = 1, 2, \ldots , N$). The transition probability from state $S_i$ at time $t_{n-1}$ to the state $S_j$ at time $t_n$ is $a_{ij} = Pr[x(t_n) = S_j | x(t_{n-1}) = S_i]$. Each $a_{ij} \geq 0$ and $\sum_{j=1}^{M} a_{ij} = 1$.

After observing the activity sequence for several days it is possible to compute the transition probabilities $a_{ij}$ between each state. For simplicity, we assumed the initial probability $\pi_i$ for some states is 1 and 0 for others. A sample of such observable Markov model is shown in Figure 4.3 where each $S_i$ can be an activity (e.g. sleeping, resting, toileting).

Figure 4.3: Observable Markov process

4.3.2.3 Modelling Hidden Markov Model

Using the initial and transition probability matrix of the observable Markov model, it is possible to calculate the probability of any new activity sequences from the current activity sequence. That is the model can estimate the probabilities of new sequence but is not capable of detecting any anomaly in the progression. But our main goal here is to detect anomalous changes in
behaviour. A widespread method of solving such a problem is the Hidden Markov Model (HMM).

HMM is a generative probabilistic model used for generating hidden states from observable data [86]. In our case, the problem of interest is to build a HMM that can best explain a given sequence of activities (e.g., normal or abnormal). If we assume that there are \( M \) observations \( O_i \) and \( N \) hidden states \( S_i \) then a HMM is characterized by the following parameter set.

\[
\lambda = (A, B, \pi). \tag{4.1}
\]

where \( A \) and \( \pi \) are already defined above. For the HMM the states are not activities such as the observable Markov process. Here, states are hidden and observable activities are called emissions.

\( B \) is a matrix of observation probabilities \( b_{ij} \) \((i = 1, 2, ...M, j = 1, 2, ...N)\). The emission probability from observable state \( S_i \) at time \( t_n \) to the hidden state \( H_j \) at time \( t_n \) is \( a_{ij} = Pr[h(t_n) = H_j | x(t_n) = S_i] \). Each \( b_{ij} \geq 0 \) and \( \sum_{j=1}^{N} b_{ij} = 1 \)

In our model, the hidden states can vary according to the design of the model. For example there can be different health conditions of the user based on observed activities (e.g., normal, abnormal, critical). To build the prior knowledge for the model the first step is to compute the model parameters \( A, B \) and \( \pi \).

To estimate the parameters of HMM, a common approach is to use the Baum-Welch algorithm [173, 174]. Here we used this algorithm to train the model with the large number of activity sequence samples for each user. The Baum-Welch algorithm uses the well-known Expectation Maximization (EM) algorithm to find the maximum likelihood estimate of the parameters. The algorithm finds \( \lambda^* = \arg\max_{\lambda \in \Lambda} P(O | \lambda) \) where, \( \Lambda = \{\lambda_1, \lambda_2, ...\lambda_k\} \) set of available models. We have varied number of hidden states in each \( \lambda_i \) to find optimal number of states. The probability of an observation sequence \( O = (O_1, O_2, ...O_M) \) given the model \( \lambda \) is computed by Equation 4.2.

\[
P(O | \lambda) = \sum_{s \in \text{valid}(S)} \prod_{t=1}^{T} a_{s_{t-1} s_t} b_{s_t}(o_t) \tag{4.2}
\]

Here \( \text{valid}(S) \) are all the valid state sequences starting in a starting state and ending in an ending state. [86]. The models are trained using both normal and abnormal activity sequences. So, after the training phase we obtained a set of models \( \Lambda \) with model parameters.
4.3.2.4 Abnormality prediction using HMM

Once the model parameters are computed, we can use that model to classify possible abnormalities for a given activity sequence. The hidden state sequence corresponding to an activity sequence can be identified using the Viterbi algorithm [175].

Using the generated HMM $\lambda$, it is possible to compute the log likelihood of an observation $\log P(O|\lambda)$. So, if $\lambda$ is built using training data of normal activity sequences (where each sample is a sequence of a day) then any new daily activity sequence that is not likely with $\lambda$ is considered abnormal. If a model is constructed using abnormal patterns then the same observation will be more likely to apply to that model. If a model can be constructed using every possible pattern using the hidden states labelled with abnormal and normal, our model can tell which activities are abnormal. Some examples of normal and abnormal activities and location patterns are presented in Table 4.3.

Table 4.3: Example of activity and location sequences

<table>
<thead>
<tr>
<th>Type</th>
<th>Normal</th>
<th>Abnormal</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>sleeping-waking-up-toileting-breakfast-resting-exercise-resting-lunch</td>
<td>sleeping-waking-up-toileting-breakfast-toileting-resting-toileting-household-lunch</td>
<td>no exercise, more toileting activity</td>
</tr>
<tr>
<td>Activity</td>
<td>sleeping-waking-up-toileting-Medication-breakfast-resting</td>
<td>sleeping-waking-up-toileting-breakfast-resting-household-resting</td>
<td>medication is not taken before breakfast</td>
</tr>
<tr>
<td>Activity</td>
<td>lunch-household-resting-household-dinner-household-sleeping</td>
<td>lunch-household-resting-sleeping-waking up-household-resting-dinner</td>
<td>sleep activity in midday before dinner</td>
</tr>
<tr>
<td>Activity</td>
<td>lunch-household-resting-household-dinner-household-resting-sleeping</td>
<td>lunch-household-resting-toileting-household-resting-sleeping</td>
<td>user goes to sleep without having dinner</td>
</tr>
<tr>
<td>Location</td>
<td>bedroom-toilet-living-dining-living-kitchen-living-toilet-bedroom</td>
<td>bedroom-toilet-living-kitchen-toilet-bedroom</td>
<td>more toilet visit</td>
</tr>
</tbody>
</table>

If a model $\lambda$ is developed through training with normal observations then $\log P(O|\lambda)$ will be lower for any sequence that is not in normal observation list. So, this will be classified as abnormal behaviour. If a model is built with 2 hidden states (normal, abnormal) with both normal and abnormal observation then the Viterbi algorithm will produce the hidden state
sequences. The system can then identify which activities are anomalous and draw conclusions (as shown in Observation of Table 4.3).

Abnormality can be frequent in an activity sequence. As the HMM models are built using long-term observations of normal and abnormal patterns, so the effect of such frequency during classification will be minimal. Also, the situation in other context domains will suppress the false anomaly in activity context domain. The model will also be fine-tuned through continuous adaptation for any quick change in activity patterns.

With a similar approach, using the sequence of locations as observable state, we can also determine the abnormality in location transition behaviour.

4.3.3 Change detection in lifestyle pattern and routine behaviour

In the previous step we have determined abnormality in activity sequence using HMM. But in using that model it is not possible to detect any behavioural change of a user’s lifestyle. To detect such changes, we suggested a model based on statistical analysis of some routine behaviour. This model is presented below.

4.3.3.1 Routine behaviour discovery

The HMM model cannot tell whether the user is maintaining a fixed timetable for daily activities or if any change is occurring in the timetable. For finding abnormalities in daily behaviour and detecting changes in lifestyle patterns, we relied on statistical estimation. Some activities are performed daily within a particular time. For example, a user generally takes lunch between 1:00 p.m. and 2:00 p.m. everyday. But one day he/she took lunch at 3:00 p.m. The goal of this model is to detect such types of behavioural abnormalities that is related to time, time duration or number of times in a day.

From daily activity log it is possible to summarize some particular routine activities of the user such as sleeping time, waking up time, breakfast time, lunch time. From a large set of samples with normal pattern we can measure the Gaussian distribution of those activities where the start time of each activity is around a specific mean time, $\mu$ and standard deviation $\sigma$. This process summarize the normal lifestyle pattern of the user. A sample summarized example is shown in Table 4.4.
Table 4.4: Normal routine behaviour of the user. Distribution is Gaussian. Mean in start time, SD in minutes. Min and Max duration is in minutes

<table>
<thead>
<tr>
<th>Activity</th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep</td>
<td>22:30</td>
<td>30</td>
<td>400</td>
<td>500</td>
</tr>
<tr>
<td>Wake up</td>
<td>06:00</td>
<td>15</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Breakfast</td>
<td>08:00</td>
<td>20</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Exercise</td>
<td>10:00</td>
<td>20</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>Lunch</td>
<td>13:15</td>
<td>40</td>
<td>25</td>
<td>40</td>
</tr>
<tr>
<td>Dinner</td>
<td>19:30</td>
<td>30</td>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>

4.3.3.2 Statistical model design

From the calculation of mean $\mu$ and standard deviation $\sigma$ we define 3 intervals $[T_0^*, T_0]$, $[T_1^*, T_1]$ and $[T_2^*, T_2]$ where

Normal range, $[T_0^*, T_0] = [\mu - \sigma, \mu + \sigma]$

Alert region range, $[T_1^*, T_1] = [\mu - \delta_1 \sigma, \mu + \delta_1 \sigma]$ and

Critical region range, $[T_2^*, T_2] = [\mu - \delta_2 \sigma, \mu + \delta_2 \sigma]$

Here $\delta_2 \geq \delta_1 > 1$ and are called behavioural region factors. $\delta_1$ is alert region factor and $\delta_2$ critical region factor. That is if for any individual routine, if behavioural region is inside $\delta_1$ times of $\sigma$ from the mean ($\mu$) then the situation is abnormal and classified as alert condition and a region in more greater margin ($\delta_2$ time of $\sigma$) implies critical abnormality. If the routine activity is cross critical region then we call it severe anomaly. The parameters are illustrated in Figure 4.4.

Figure 4.4: Different regions and related parameters to identify abnormality in routine behaviour. The distribution is Gaussian. Normal pattern shown in solid line and a behavioural shift shown in dashed line.
4.3.3.3 Abnormality classification

As shown in Figure 4.4, each daily routine behaviour is classified into 4 classes annotated as 0, 1, 2, and 3. This is determined according to the regional interval it fits. So, the non-zero classification in any routine means abnormal behaviour. But subsequent same non-zero values (e.g. 1) in a particular routine indicates that the user behaviour for that routine activity has shifted. This situation shown as dashed line of Figure 4.4. Such changes are validated using predefined fuzzy rules. If the changes are not harmful then Gaussian distribution for the routine is updated (in Figure 4.4) and new intervals are calculated, otherwise the change is considered to be abnormal.

Fuzzy rules are used to identify true abnormality in behavioural shift by correlating other changes. For example, if the user wakes late and takes breakfast late but does other things as usual then this not abnormal. If the occurrence of such change is continuous then this indicates that the routine behaviour has shifted from current mean (as shown in Figure 4.4) and so distribution of related parameters is updated. Another example is a user’s sleep time shift to alert region and wake up time shift to critical region. The outcome of fuzzy rule concludes that the user is sleeping more than usual. A continuity in such behaviour implies that the user is experiencing an oversleep problem and this may be due to some weakness. So, this is a serious abnormality. By correlating such routine behaviour changes with other health-related changes, our system can make the final decision for such kinds of abnormality.

Using similar statistical measures it is possible to detect some other behaviours of the user. For example, by counting the number of toileting activities in a particular period (e.g. 4 hours) the toileting frequency of the user can be calculated. After measuring the Gaussian distributions and related intervals of each period the abnormality in toileting behaviour can be determined using the described process. This is how other behavioural activities such as exercise duration, smoking, alcohol consumption, and food consumption can be tracked.

4.3.4 Physiological change detection in different vital signs

Changes in physiological parameters or ‘vital signs’ [49, 176] commonly precedes serious health-related issues or even death. So early detection of such changes can minimize health-related risks for the user. In a context-aware assisted living system, some vital signs of the user are continuously measured periodically. The deviation from normal thresholds in any vital sign indicates an abnormality. Upon detection of an abnormality a notification is sent to the healthcare professional for further investigation. So the system triggers only current occur-
Methodology

Numerical thresholds of different vital signs of the user in different physical conditions using a mix of association rule mining and data mining techniques. A two step personalized knowledge discovery-based model is developed where in first step, association rules are generated from the statistical observation of vital sign data using MapReduce Apriori process. In the second step, the dataset generated using those rules is then trained and tested using different supervised learning techniques (e.g. J48 Decision Tree, Naive Bayes, Multi Layer Perceptron, Radial Basis Function). We achieved good accuracy (above 90%) for J48 and MLP classifier for different kinds of patients (e.g. having variations in blood pressure values). The developed classifier can detect user-specific abnormality in different vital signs with good accuracy. Consequently, abnormality detection from vital signs is beyond the scope of this paper. We assume that we already have a model for it. In this work we mainly focus on the forecasting technique of any disease from current observations of vital signs data.

4.3.4.1 Vital signs and related disease identification

An abnormal condition may develop for a continuous change in any vital sign. Our goal here is to detect the trends of those changes to forecast future abnormalities. In our experiment, we only used those vital sign contexts which have numerical value. Some typical vital signs and their normal range according to the medical rule are presented in Table 4.5 [149, 177, 178]. The disease development for exceeding or falling behind these normal ranges is described in Table 4.6 [149, 177, 178].

Table 4.5: Normal range of vital signs

<table>
<thead>
<tr>
<th>Vital Sign</th>
<th>Normal Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart Rate</td>
<td>50-100 beats per minute</td>
</tr>
<tr>
<td>$O_2$ saturation</td>
<td>93-100%</td>
</tr>
<tr>
<td>Blood pressure</td>
<td>Systolic: 90-160 mmHg, Diastolic: 60-90 mmHg</td>
</tr>
<tr>
<td>Respiration</td>
<td>9-20 breaths per minute</td>
</tr>
<tr>
<td>Temperature</td>
<td>36.1$^\circ$C to 37.9$^\circ$C</td>
</tr>
<tr>
<td>Blood Sugar</td>
<td>from 4.0 to 5.9 mmol/L (before meal) and under 7.8 mmol/L (after meal)</td>
</tr>
</tbody>
</table>

4.3.4.2 Forecasting model development

The continuous monitoring of any vital sign $\nu_i$ produced time-series data of a sequence of observation. Such time series data of two vital signs - heart rate (HR) and blood pressure (
Methodology

Table 4.6: Different diseases for the change in vital signs

<table>
<thead>
<tr>
<th>Disease</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tachycardia</td>
<td>heart rate greater than 100 beats per minute</td>
</tr>
<tr>
<td>Bradycardia</td>
<td>heart rate fewer than 50 beats per minute</td>
</tr>
<tr>
<td>Hypertension</td>
<td>Systolic BP ≥ 160, Diastolic BP ≥ 90</td>
</tr>
<tr>
<td>Hypotension</td>
<td>Systolic BP ≤ 90, Diastolic BP ≤ 60</td>
</tr>
<tr>
<td>Fever</td>
<td>temperature above 37.9° C</td>
</tr>
<tr>
<td>Hypothermia</td>
<td>temperature below 36.1° C</td>
</tr>
<tr>
<td>Hypoxia</td>
<td>O$_2$ saturation less than 93%</td>
</tr>
</tbody>
</table>

SBP and DBP) for 3 users are shown in Figure 4.5.

![Time-series graph of 2 vital signs(HR,BP) with 96 observations for 3 users](image)

Figure 4.5: Time-series graph of 2 vital signs(HR,BP) with 96 observations for 3 users

The observed phenomenon is a random process and for such mechanism exponential smoothing techniques suit best. For our model, we have used an extended version of simple exponential smoothing which is Holt’s liner trend method [87]. This method allows forecasting of data with a trend (increasing or decreasing). This method involves two smoothing equation and one forecast equation. They are

Level equation: $l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$  \hspace{1cm} (4.3)

Trend equation: $b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$ \hspace{1cm} (4.4)

Forecast equation: $F_{t+h} = l_t + hb_t$  \hspace{1cm} (4.5)

Where, $l_t$ denotes an estimate of the level of the series at time $t$, $b_t$ denotes an estimate of trend (sloop) of the series at time $t$, $\alpha$ is the smoothing parameter for the level, $0 \leq \alpha \leq 1$ and $\beta$ is the smoothing parameter for the trend $0 \leq \beta \leq 1$. In the level equation $l_t$ is adjusted for the trend of last observation $b_{t-1}$, by adding it to the last smoothed value of $y_t$. $l_t$ becomes
a weighted average of actual observation $y_t$ and the within-sample one-step-ahead forecast for time $t$ ($l_{t-1} + b_{t-1}$). Thus, the lag is eliminated and $l_t$ is brought to the approximate level of the current data value of $y_t$.

The trend equation shows that, $b_t$ is a weighted average of previous estimated trend, $b_{t-1}$ and the difference between last two smoothed values, $(l_t - l_{t-1})$. The forecast function is trending. The $h$-step ahead forecast is equal to the last estimated level in addition with $h$ times of last estimated trend. That is, forecast is a liner function of $h$. Table 4.7 shows an example of this process for Tachycardia prediction from heart rate value [179]. Figure 4.6 shows an example of fever trend detection after applying this model on observed body temperature value that has increasing tendency. For both cases $\alpha = 0.5$ and $\beta = 0.5$.

<table>
<thead>
<tr>
<th>Period</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>39</th>
<th>40</th>
<th>$h$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_t$</td>
<td>71</td>
<td>71</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>...</td>
<td>80</td>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$l_t$</td>
<td>71</td>
<td>71</td>
<td>71.5</td>
<td>72.875</td>
<td>72.0938</td>
<td>...</td>
<td>80.2021</td>
<td>80.7391</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$b_t$</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.3125</td>
<td>0.2656</td>
<td>...</td>
<td>0.2764</td>
<td>0.4065</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F_t$</td>
<td>-</td>
<td>71</td>
<td>71.75</td>
<td>72.1875</td>
<td>...</td>
<td>80.4042</td>
<td>80.4782</td>
<td>81.156</td>
<td>81.5521</td>
<td>81.9586</td>
<td>...</td>
<td>84.8042</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.6: Estimation of body temperature value for detecting fever symptoms. Here, the forecast is for 10 periods. We can see that the forecast graph follow a liner trend.

### 4.3.5 Information fusion and decision making using fuzzy rules

From the three of the previously described models, it is possible to draw some conclusions from detected abnormalities. Individually each model can confirm the presence of some abnormal behaviour or trend for the particular context domain. But when the outcomes are combined it is possible that the final situation may not be abnormal. In order to avoid of sending false alert to the healthcare providers the correct interpretation is required by correlating the
abnormalities of all domains. We have used fuzzy rule-based models [60] to detect such global abnormalities. For example, the HMM model outputs abnormality for frequent occurrence of toileting activities. The daily routine model detects the breakfast time of the user for that day shifted in critical interval and the disease prediction model found a tendency of increasing of body temperature. When a user visits an expert and tells him about these symptoms, the expert tries to guess some diseases according to the patterns of the symptoms. In our model, those opinions are converted to fuzzy rules and stored in the cloud database. So, when our system detects such symptoms in individual domain it tries to match those in a set of fuzzy rules. For example in the given scenario above, a fuzzy rule can conclude that user is not feeling well and suggests diarrhoea with possible fever. Such fuzzy rules are designed using references from medical journals, medical observations and medical experts opinions.

In fuzzy rule-based modelling the rule base is constructed with the structure: *if < antecedents> then <consequent>.*

In our model, we standardize the fuzzy rules with the following structure.

- Set of rules $R = \{R_1, R_2, \ldots, R_n\}$
- Set of outcomes from context domains, $V = \{V_1, V_2, \ldots, V_x\}$
- Set of possible decisions, $D = \{D_1, D_2, \ldots, D_z\}$
- Rule, $R_1$: *if $(V_1, V_2, \ldots)$ and/or $(V_x, V_y, \ldots)$ then $(D_p, D_q, \ldots)$*

For example,

*if (less exercise) and (more eating) and (wake up in critical zone) and (BP increasing) and (BMI increasing) then (state is Warning)*

Here our models can detect that the user is doing less exercise, eating more food and waking up late. This behavioural change in daily routine is affecting his health condition. This may result in possible weight gain and increase in blood pressure level in the near future. So, for this kind of change it is not necessary to send alert or emergency request to the healthcare service providers. Rather, our fuzzy rule classifies this situation as warning. In this case, a context-aware action can be a warning to the user to change his activities (do more exercise, eat proper food and wake up in time). Otherwise, this irregularity will increase his BP and weight, and that can cause some critical health problem in future. So, if the user follows the instructions accordingly, our models will adapt to those changes also. If the system detects
that the user is reverting to his usual routine and his health condition is also predicted to be normal, it classifies the current situation as normal and thus avoid sending needless alerts to the doctor. This is how our model predicts changes, minimizes false alerts and notifies at an early stage before there is any severe danger. In our fuzzy model a situation is classified as one of the four types: normal, warning, alert and emergency. More about this fuzzy model is described in Section 4.4.5 with some more realistic examples.

4.4 Experimental Results

In accordance with the developed models for change detection in the context-aware AAL system described earlier, some case studies are implemented to evaluate the proposed system. The objectives of these experiments are: (i) to generate the described models using train dataset; (ii) to test the validity of the models using test data; (iii) to apply outcomes of different models in rule-based recognition, and (iv) to evaluate the performance and accuracy of change detection.

4.4.1 Case study description

A large amount of data is generated synthetically from the observations of some real-life datasets of existing research [180, 181]. The data represents activities and vital sign logs of multiple users. Such data logs for 3 months for 10 different users are generated using MAT-LAB. To make dataset more realistic, a mixed combination of abnormal and normal data is produced randomly for each user. The examples of activity, location and vital sign logs for a user are presented in Table 4.8 and Table 4.9.

For simplicity, eight activities are used for the experiment. They are: resting, sleeping, waking up, walking, eating, toileting, exercising and household, denoted by \( x_1 \) to \( x_8 \). Here, the household activity represents activities such as showering, cooking, cleaning, working on computer or watching TV. Four locations are used in the experiment and they are bedroom, kitchen, toilet and dining. We assumed that in our context-aware model these activities and locations are easily identifiable using existing knowledge of activity recognition [48, 70, 142, 182].
4.4.2 Implementation and validation of proposed HMM

From the generated data for 10 users, first a single user data is chosen for this experiment. Activity sequences for 200 samples are created from those daily logs using MATLAB program. Then, utilizing Hidden Markov Model library of MATLAB and Simulink tool, a HMM is created with random transition ($A$) and emission matrix ($B$). The model is initialized by the the first observation sample. Initially the model is created using 8 hidden states and is trained repeatedly using the same observation sequence until it terminates. The generated HMM model was trained again with different sequences of normal activities using the transition and emission matrix of first training as initial condition. The same process was repeated by increasing number of hidden states. Afterwards, a random normal observation sequence of a day, $O$, is picked and $LogP(O|\lambda)$ is measured for all the constructed HMM, $\lambda \in \Lambda$. Figure 4.7
show shows the change in log likelihood value for an observation \( O \) with the increment in number of hidden states.

![Graph](image)

Figure 4.7: Model performance improves when the number of hidden states increase. But gain is low after hidden states reaches 40. So the model with 40 hidden states is considered the optimized model and that is selected as standard model for classification.

![Diagram](image)

Figure 4.8: The Hidden Markov Model for detecting abnormal activity

Using 40 hidden states the first HMM is built from normal observations. A second HMM is built using a similar process but trained only with abnormal activity sequences (e.g. frequent toileting, absence of exercise or eating activity, frequent resting activity and less household activity, frequent sleep). Finally, a third HMM is trained using both normal and abnormal observations and predefined hidden states. In the experiment 3 hidden states (normal, abnormal and critical) are used. The model is shown in Figure 4.8.

20 normal days and 20 abnormal days are kept as test dataset to validate the first two models. The classification results observed for user 1 using the first 2 models with normal and
abnormal patterns are shown in Figure 4.9. \( \log P(O|\lambda) \) is calculated using forward-backward procedure [86].

Figure 4.9: Observed value of \( \log P(O|\lambda) \) for 20 normal and 20 abnormal samples. The line indicates the cluster boundary to distinguish between the two classes.

From the above results, we can see that the first two HMMs can isolate the normal and abnormal activity sequence with good accuracy. The third model is built to find the hidden states; that is, which activities in the sequence are abnormal and critical. Using Viterbi algorithm, the hidden state sequence of normal and abnormal activities are measured. Some of the observed results are shown in Table 4.10.

<table>
<thead>
<tr>
<th>Observed Sequence</th>
<th>Hidden State Sequence</th>
<th>Manual observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_3x_4x_6x_8x_5x_8x_7</td>
<td>s_1s_1s_1s_1s_1</td>
<td>normal behaviour</td>
</tr>
<tr>
<td>x_3x_4x_5x_6x_4x_2</td>
<td>s_1s_1s_2s_2s_1s_1</td>
<td>abnormal behaviour, frequent toilet visit</td>
</tr>
<tr>
<td>x_3x_8x_7x_1x_8x_5x_8</td>
<td>s_1s_1s_1s_1s_1</td>
<td>normal behaviour</td>
</tr>
<tr>
<td>x_3x_5x_7x_1x_8x_5x_5</td>
<td>s_1s_1s_2s_2s_1s_1</td>
<td>abnormal behaviour, no exercise</td>
</tr>
<tr>
<td>x_3x_5x_8x_1x_8x_1x_6x_2</td>
<td>s_1s_1s_2s_2s_2s_3</td>
<td>critical behaviour, frequent toilet visit, slept without dinner</td>
</tr>
</tbody>
</table>

This third model is tested using 35 observation sequences. We did similar experiments for location behaviour. The above process is repeated for all the 10 users and the classification accuracies with number of false positives are presented in Table 4.11.

From the above experiment it can be concluded that the developed HMM models can detect abnormality in current observation sequence with good accuracy. It is also capable of predicting change in future if subsequent observation sequence is given (e.g. next most likely activity). The developed models can also sense the presence of abnormality and answer what activities will cause these anomalies. So, from the outcomes of the model, some conclusions of user behaviours can be summarized and used for the fuzzy fusion model later.
Table 4.11: Activity sequence classification accuracy using different HMM for 10 users

<table>
<thead>
<tr>
<th>User</th>
<th>HMM built with normal observations</th>
<th>HMM built with abnormal observations</th>
<th>HMM with 3 predefined hidden states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>FP</td>
<td>Accuracy</td>
</tr>
<tr>
<td>1</td>
<td>97.5%</td>
<td>1</td>
<td>95%</td>
</tr>
<tr>
<td>2</td>
<td>97.5%</td>
<td>1</td>
<td>97.5%</td>
</tr>
<tr>
<td>3</td>
<td>95%</td>
<td>1</td>
<td>92.5%</td>
</tr>
<tr>
<td>4</td>
<td>97.5%</td>
<td>1</td>
<td>90%</td>
</tr>
<tr>
<td>5</td>
<td>90%</td>
<td>3</td>
<td>95%</td>
</tr>
<tr>
<td>6</td>
<td>100%</td>
<td>0</td>
<td>97.5%</td>
</tr>
<tr>
<td>7</td>
<td>92.5%</td>
<td>3</td>
<td>95%</td>
</tr>
<tr>
<td>8</td>
<td>95%</td>
<td>1</td>
<td>90%</td>
</tr>
<tr>
<td>9</td>
<td>97.5%</td>
<td>0</td>
<td>97.5%</td>
</tr>
<tr>
<td>10</td>
<td>87.5%</td>
<td>4</td>
<td>92.5%</td>
</tr>
</tbody>
</table>

4.4.3 Implementation and validation of routine behaviour change detection method

The HMM model can detect anomalies in activity patterns but cannot detect any behavioural shift in daily routines. For this purpose, a model is developed to understand the normal routine of the user. The dataset of the last of the above experiments is also used here. To begin with, the Gaussian distribution of some routine behaviours (e.g. sleep time, wake up time, breakfast time, lunch time, dinner time, exercise time) are measured (Table 4.4). Any eating activity between 7:00 and 10:00 is considered as breakfast; between 12:00 and 15:00 considered as lunch and between 18:00 and 22:00 is dinner. For each routine behaviours, maximum and minimum durations of the activity (in minutes) are also calculated.

After building the model for normal routine the 3 intervals for each routine are measured as described in Section 4.3.3 and Figure 4.4. The interval values are kept in personal threshold database. For this experiment $\delta_1 = 1.3\sigma$ and $\delta_2 = 2\sigma$ are used. 50 samples containing regular and abnormal routine are tested in the generated models. The observed results are presented in Table 4.12 and 4.13. Note that, we did not generate any data for severe case.

For some routines like toileting frequency, a day is divided into 4 periods and the Gaussian distributions of that behaviour for each period in each day are also measured.

If alert value is observed for same routine for 5 consecutive days then the Gaussian distribution of that routine is recalculated by including these new data and the threshold tables are updated with a new value (e.g. mean time shifted as in Figure 4.4). If more than two anomalies are detected in a single day sample then fuzzy rules are applied to decide about a
### Experimental Results

#### Table 4.12: Classification for routine behaviours

<table>
<thead>
<tr>
<th>Routine behaviour</th>
<th>Normal</th>
<th>Alert</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep Time</td>
<td>32</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>Wake up Time</td>
<td>27</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>Breakfast Time</td>
<td>23</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>Lunch Time</td>
<td>33</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Dinner Time</td>
<td>29</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>Exercise Time</td>
<td>21</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>

#### Table 4.13: More than 1 anomalous behaviour cases

<table>
<thead>
<tr>
<th>Number of abnormalities</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>anomaly in 2 routine</td>
<td>19</td>
</tr>
<tr>
<td>anomaly in 3 routine</td>
<td>8</td>
</tr>
<tr>
<td>anomaly in 4 routine</td>
<td>4</td>
</tr>
<tr>
<td>anomaly in 5 routine</td>
<td>2</td>
</tr>
<tr>
<td>Sleep less than usual</td>
<td>8</td>
</tr>
<tr>
<td>Sleep more than usual</td>
<td>6</td>
</tr>
<tr>
<td>No exercise</td>
<td>5</td>
</tr>
<tr>
<td>Exercise less than usual</td>
<td>9</td>
</tr>
</tbody>
</table>

final anomaly. We will describe this in a later section.

#### 4.4.4 Implementation and validation of trend detection in vital signs

As described in Section 4.3.4.1, Holt’s formula is applied to detect increasing and decreasing trends in vital signs. The technique is applied over heart rate, systolic blood pressure, diastolic blood pressure, respiration, $O_2$ saturation, body temperature, body mass index and blood glucose value. In the experiment, these forecasting formulas are applied in vital sign logs of 100 cases (for 10 users, 10 cases each) for a lengthy period. Afterwards, the forecast value for next 100 periods is calculated. No linear trend in any vital sign was found in 37 cases. For the other 63 cases, the trend is detected in one or more vital signs. Table 4.14 summarizes the findings of those 63 samples.

The above table validates our process that the proposed model can predict any growing trend in vital signs at an early stage and thus can detect the presence of any disease symptoms.
Table 4.14: Trend detection result in different vital signs

<table>
<thead>
<tr>
<th>Type</th>
<th>increasing</th>
<th>decreasing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart rate</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>SBP &amp; DBP</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>Body Mass Index</td>
<td>21</td>
<td>14</td>
</tr>
<tr>
<td>SpO\textsubscript{2}</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Body Temperature</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Blood Glucose</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Respiration</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4.4.5 Implementation of fuzzy rule-based model and wellness detection

All the above implementations validate our developed methodology and show the ability to detect abnormalities or trends with high accuracies. From each anomaly and trend it is possible to draw some high level conclusions which are used as antecedents of fuzzy rules. The main feature of a context-aware system is to predict behaviour by combining and correlating information from multiple contexts. In our case, for decision fusion a fuzzy rule-based model is used. Different anomalies can occur in different domains and the type of abnormality largely depends on the user. It is difficult to train such a model with many possible cases. Therefore, a set of fuzzy rules are used to make the final decision. An integrated case study for 10 different user scenarios are implemented. A portion of observed decisions along with the fuzzy rules is shown in Table 4.15.

In Table 4.15 we can see that, although anomalies in the individual domain occurred for User 3 but after combining the information in a fuzzy rule it is found that the user has growing trend of heart rate for excessive physical activity. In other cases, anomalies in multiple domains confirm that user is unwell and there is a need to send an appropriate context-aware action.

The decision of the fuzzy rule model is categorized into 4 states as shown in Table 4.16. The detected abnormality of each context domain is classified into different types based on the decision for that domain. This is shown in Table 4.17 (We do not consider location domain for this experiment). These values are used to build the antecedents of each fuzzy rule (as shown in the Antecedents column of Table 4.15). To test the accuracy of fuzzy model 1000 random rules are generated for 10 users using different combinations of abnormality values as shown in Table 4.17 along with their output state (consequent) value as shown in Table 4.16. The rules are then tested using J48 decision tree classifier with 10-fold cross validation. The decision tree is chosen because by using this specific classifier it is easy to describe a rule-based model.
### Table 4.15: Decision using fuzzy rules from detected abnormalities in different domains

<table>
<thead>
<tr>
<th>User</th>
<th>Activity Anomaly</th>
<th>Routine Behaviour Anomaly</th>
<th>Vital Sign Anomaly</th>
<th>Antecedents</th>
<th>State</th>
<th>Description of Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>resting instead of doing exercise</td>
<td>sleep duration low and shifted in alert range</td>
<td>increasing trend detected in BMI</td>
<td>more rest, less exercise, more eat, less household, sleep duration in alert zone, wake up time in critical zone, BMI increasing, BP increasing.</td>
<td>Warning</td>
<td>User is experiencing multiple problems. There is a possibility of hypertension development for less physical activity, improper sleep and unbalanced diet. So, context-aware action is to send a warning to the user to follow proper routine.</td>
</tr>
<tr>
<td>2</td>
<td>frequent toileting activity detected</td>
<td>delay in taking lunch</td>
<td>more toileting, less household, lunch time in alert zone, toilet going frequency in critical zone, body temp. increasing</td>
<td>Alert</td>
<td>User is not well and there is a possibility of fever with diarrhoea. So, the context-aware action is to send an alert to the doctor before any severe danger.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>exercise activity detected during resting time</td>
<td>exercise duration is higher than usual time</td>
<td>increasing trend in HR</td>
<td>more exercise, less rest, exercise duration in alert zone, HR increasing</td>
<td>Normal</td>
<td>HR is increasing for more physical activity. Situation in individual domain detected abnormality. But overall it is a normal condition.</td>
</tr>
<tr>
<td>4</td>
<td>sleeping activity detected during exercise</td>
<td>exercise duration is 0, that is no exercise detected</td>
<td>more sleeping, no exercise, sleep time in critical zone, wake up time in critical zone, HR increasing</td>
<td>Alert</td>
<td>User is sleeping less and not doing any exercise. Also there is a high possibility of bradycardia. This is a serious issue and so the context-aware action is to send an alert to the doctor.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>sleeping activity detected during wake up time</td>
<td>sleep duration very high and wake up time shifted in critical region</td>
<td>increasing trend in SpO₂ and Respiration</td>
<td>more sleep, sleep duration in critical zone, wake up time in critical zone, BP increasing, Resp. decreasing, SpO₂ decreasing</td>
<td>Emergency</td>
<td>User is not waking up and sleeping for long time. At the same time abnormal changes are detected in multiple vital signs. So, the system sends an immediate alert to the emergency.</td>
</tr>
<tr>
<td>6</td>
<td>frequent toileting activity detected</td>
<td>toilet going frequency high</td>
<td>increasing trend in blood sugar</td>
<td>more toileting, toilet going freq. in alert zone, BG increasing</td>
<td>Alert</td>
<td>Diabetes symptoms detected. So the system sends alert to the doctor.</td>
</tr>
</tbody>
</table>

The achieved accuracy is 95.10%. The detail accuracy by class of this experiment is presented in Table 4.18. The observed results are compared with some context-aware models and this is shown in Table 4.19.

All the experiments are performed over synthetically-generated data from real observations.
Table 4.16: Type of decisions from fuzzy rule model

<table>
<thead>
<tr>
<th>Class</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>The detected abnormal changes are normal</td>
</tr>
<tr>
<td>Warning</td>
<td>1</td>
<td>The detected abnormal changes can be harmful</td>
</tr>
<tr>
<td>Alert</td>
<td>2</td>
<td>Require a doctor to investigate the detected abnormal changes</td>
</tr>
<tr>
<td>Emergency</td>
<td>3</td>
<td>The detected changes are very dangerous</td>
</tr>
</tbody>
</table>

Table 4.17: Type of abnormalities for different context domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Number of type</th>
<th>Abnormalities in attribute value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>8 types of activity (Figure 4.8)</td>
<td>more, less or no occurrence</td>
</tr>
<tr>
<td>Routine behaviour</td>
<td>12 type of routine behaviours (Listed 6 in Table 4.12)</td>
<td>behaviour shifted in alert or critical region</td>
</tr>
<tr>
<td>Vital signs</td>
<td>7 types of vital signs (Table 4.14)</td>
<td>increasing or decreasing</td>
</tr>
</tbody>
</table>

Table 4.18: Detailed accuracy by class

<table>
<thead>
<tr>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>ROC Area</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.985</td>
<td>0.028</td>
<td>0.989</td>
<td>0.985</td>
<td>0.987</td>
<td>0.997</td>
<td>0</td>
</tr>
<tr>
<td>0.958</td>
<td>0.044</td>
<td>0.869</td>
<td>0.958</td>
<td>0.911</td>
<td>0.989</td>
<td>1</td>
</tr>
<tr>
<td>0.424</td>
<td>0.008</td>
<td>0.692</td>
<td>0.424</td>
<td>0.526</td>
<td>0.98</td>
<td>2</td>
</tr>
<tr>
<td>0.172</td>
<td>0.002</td>
<td>0.913</td>
<td>0.172</td>
<td>0.29</td>
<td>0.997</td>
<td>3</td>
</tr>
<tr>
<td>0.951</td>
<td>0.031</td>
<td>0.949</td>
<td>0.951</td>
<td>0.946</td>
<td>0.994</td>
<td>weighted</td>
</tr>
</tbody>
</table>

and good accuracy is obtained when compared to other context-aware models. Therefore, the proposed model is helpful for early detection of anomalies in context-aware system as well as minimizing false alerts in remote healthcare monitoring. Overall, our system can make predictions at an initial stage and take appropriate context-aware actions before the user is in a potentially dangerous situation.

Table 4.19: Comparison of our model with some context-aware models

<table>
<thead>
<tr>
<th>Other Model</th>
<th>Accuracy</th>
<th>Our Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress classification [66]</td>
<td>Weighted ROC area: 0.841</td>
<td>Weighted ROC area: 0.994</td>
</tr>
<tr>
<td>Asthma attack classification [65]</td>
<td>accuracy 93.8%</td>
<td>95.10%</td>
</tr>
<tr>
<td>Fuzzy Model[73]</td>
<td>accuracy 90%</td>
<td>95.10%</td>
</tr>
</tbody>
</table>
4.5 Conclusion

In view of the increasing demand for remote patient-care and elderly-care, and for early detection of the symptoms of chronic illnesses, a context-aware change detection model has been developed using machine learning and statistical models. In contrast to existing solutions where abnormality detection is performed only in a current context, our system can recognize anomalies both in present and future behaviours using pattern recognition and mathematical models based on long-term context histories. Fusing outcomes of individual model to a fuzzy rule-based system, our developed solution is also capable of predicting symptoms and can make appropriate context-aware decisions for future well-being. A HMM-based technique was developed for detecting anomalies in the sequence of daily activities. A statistical model was described to understand irregularities in daily routines and to identify any behavioural shift in regular activities. An advanced disease prediction model is described using Holt’s linear trend method for forecasting the progression of various syndromes by measuring the trends in different physiological states. Finally, a fuzzy rule-based model is described that can combine the anomalies from all domains to make the final decisions such as whether it is a true anomaly and, if so, to whom context-aware actions need to be sent. The system also reduces subsequent false alerts to the monitoring bodies. Overall, it can be an excellent tool for health-care professionals by early prediction of symptoms and sending notification only when a patient needs attentions.

4.6 Epilogue

The contributions of this chapter complete the answer to Research Question 2. Our system can detect current abnormalities and predict future behaviour using context correlations. The next aim is to predict future value of some health parameters and future states after a time gap. Moreover, we want some mechanism for continuous prediction. So far we have mostly used simulated data for our evaluations. But due to the unavailability other contextual data of patients such as activities being undertaken at the time of changes in vital signs, we limited our research focus to use only vital signs data (e.g. heart rate, blood pressure, respiration, SPO$_2$), that is, vital sign context domain for further learning model development. Furthermore, a public dataset with long-term vital signs data are available which we find is suitable to validate our model. Therefore, from and including Chapter 5 we will only use vital signs context domain for future predictions.
Chapter 5

Patient-specific Future Clinical Event Prediction

Changes in multiple vital signs are highly correlated. It is possible to make better predictions if the changes of multiple vital signs are considered together rather estimating them individually. With reference to the third research question, this chapter aims to develop an intelligent tool for personalized monitoring and clinical decision support through early estimation of patient-specific vital sign values, and prediction of anomalies using the relationships among multiple vital signs. Here, the interrelations of vital signs are considered as context. In this chapter, multi-label classification (MLC) algorithms are extensively applied in classifier design to forecast vital sign values and related abnormalities. We propose a completely new approach of patient-specific vital sign prediction system using their correlations. The developed technique can guide healthcare professionals to make accurate clinical decisions.

A decision support system (DSS) [183] in remote patient monitoring is designed to assist healthcare professionals with decision making tasks such as disease prevention and diagnosis [77]. Context-awareness [44] is an essential part of clinical decision analysis [184]. In a real-time monitoring, various health-related data of a patient are collected at a fixed sampling interval and analysed continuously to discover the current health situation of the patient. Using the ongoing context information of a patient the application can send proper alerts or messages to the doctors in remote monitoring centres. Some clinical decision support systems (CDSSs) [184] are capable of sensing clinically abnormal consequences in advance based on the intelligent analysis over recently observed medical data of a patient and before they are presented to the clinicians. The traditional CDSSs primarily advise medical abnormalities by assessing data of
a single vital sign such as heart rate (HR), blood pressure (BP), respiratory rate (RR), oxygen saturation (SPO$_2$), and body temperature. However, they suffer from high false alerts. Most of the serious clinical anomalies occur as a result of irregularities in multiple vital signs [49] at the same time. Therefore, an important part of a context-aware remote monitoring application is the accurate and early prediction of abnormalities [88] that occur due to the changes in multiple vital signs.

![Vital Signs Correlations](image)

Figure 5.1: Strong correlations between a pair of vital signs. (a)-(e) examples of positive correlation (f) example of negative correlation. Here SBP, DBP and MBP refers to Systolic, Diastolic and Mean blood pressure respectively.

For many clinical abnormalities, a single vital sign does not contain enough information for the doctors. The symptomatic patients are likely to have several abnormal vital signs. For example hypotension (low blood pressure), tachycardia (elevated heart rate) and hypothermia (decrease in body temperature) can cause sepsis. Hypoventilation can occur when low respiratory rate is accompanied by low oxygen saturation (SPO$_2$) [149]. Therefore, strong positive or negative correlations in multiple vital signs contain useful information for predicting disease symptoms or anomalies. Some examples of such correlations are shown in Figure 5.1. These correlations can be repetitive in some patients and vary over time in some other patients. The remote monitoring doctors must be aware of these anomalies and must incorporate them explicitly into a decision to avoid any potentially dangerous clinical situation associated with the changes in multiple vital signs. Unfortunately, no good learning model exists that can perceive such patient-specific changes early enough to assist the physicians in real-time to make proper clinical decisions. Therefore, an enhanced CDSS is required with a fast, well-trained and adaptive learning model which is less likely to make false predictions and so the doctors can take proper diagnostic actions.

This chapter shows how multi-label classification (MLC) techniques [89] can be utilized to
identify the changes in multiple vital signs quickly and efficiently. Considering the short-term summary statistics and correlations of all vital signs in parallel a suitable feature vector for multi-label classifier is generated. Those features are then used to build machine learning models to make short-term predictions of vital sign threshold values. The overall system will reduce false alerts in the monitoring stations and will also help the early detection of clinical dangers. We formulate our model to a MLC problem because we want to achieve multiple targets (the range of all vital sign values) using the same information and at the same time.

The outline of this chapter are as follows. Section 5.1 describes the motivations and contributions of this chapter. Section 5.2 presents a literature review. Section 5.3 describes the overall system design and concepts. Section 5.4 explains the theoretical methods and related implementations. Section 5.5 shows the experimental results and comparisons. Finally, the chapter is summarized in Section 5.6.

5.1 Motivations and Contributions

5.1.1 Motivations

The long-term effects of the irregularities in multiple vitals can result serious chronic illness. A clinical decision support system (CDSS) with early prediction capability can mitigate such problems. In our previous chapters we developed learning techniques for personalized knowledge discovery [85], future abnormalities prediction, and behavioural change detection [88] using various contexts of a patient in an assisted living environment. We have also showed the advantage of utilizing cloud platforms for such learning tasks. In continuation of these works, in this chapter we mainly focus on vital sign correlations and utilize these contexts to produce a useful tool for the healthcare professionals by estimating patient-specific future trend of various vitals in advance. Our developed methodology will assist the doctors to make better decisions, diagnosis and treatment, resulting in improved healthcare service quality and less chronic disease-related deaths.

5.1.2 Contributions

The developed methods make several contributions to the biomedicine and healthcare related research. They are as follows.

- We developed an intelligent CDSS by adopting MLC techniques that can detect the upcoming trends in multiple vital signs at the same time using their correlated features.
To make an accurate clinical decision a doctor should not only consider the occurrence of abnormality in each vital sign separately, but also take into account the effect of their correlations. In our approach, multi-label classifiers are extensively applied to detect such correlations in advance. According to the literature review, this is the first attempt of employing MLC in vital sign predictions using their correlations. There are no previous studies which provide such an experimental comparison of state-of-the-art MLC algorithms on vital signs data.

- We utilized the high resourceful cloud technologies for classifier training and decision support so that the system can work simultaneously for many patients. As a result, our innovative technique provides clinical decision support to a big community containing versatile patients by utilizing a common platform. At present no such system exists that can serve such a large number of patients. The proposed approach also reduces hospital load, because many patients can be monitored from home continuously. Ultimately, the adoption of our techniques can reduce the high cost of treatment.

- The developed approach provides personalized and real-time clinical decision support by detecting patient-specific anomalies, disease symptoms and emergencies in advance. In addition, this can assist healthcare experts in diagnostic decision-making with greater knowledge. The accuracy of vital signs prediction is greatly improved by considering the patient-specific correlations of those vitals. Thus, this individualized system reduces the amount of false predictions in remote monitoring centres.

5.2 Related work

There are several studies related to discovering correlation patterns in multiple vitals such as heart rate, blood pressure, respiration, O$_2$ saturation, and ECG. They mainly focus on finding future abnormalities in a specific vital sign [74, 75, 185]. However, very few attempts have been made to find future abnormalities in multiple vital signs. In biomedical area, multi-label classification techniques are mainly used in clinical text mining [186] and finding adverse effects of a patient in response to different drug events [187]. Some recent works show the advantage of multi-label classifier in clinical data analysis [188, 189].

Furthermore, machine learning techniques are widely adopted in biomedical data analysis, healthcare and clinical abnormality predictions [21]. There are quite a number of CDSSs being proposed in the literature for different purposes. Various classifiers are developed using
data mining methods for these CDSSs to aid healthcare providers in clinical decision-making process [190]. Classical data mining techniques such as support vector machine [76], artificial neural network [191], naive bayes [192] are used to detect clinical abnormalities and predict future behaviours in vital signs such as ECG [77], blood pressure [73], respiration etc. In summary, it is clear from the literature that the researchers have emphasized towards the direction of analysing a collection of continuous physiological data in real-time to extract the best knowledge of a patient situation and to find future behaviours. Most of these predictive systems are based on single goal prediction. But in this chapter, we intend to find values of multiple vitals at same time.

5.3 The framework for clinical decision support system (CDSS)

Our objective is to develop a CDSS that helps doctors by estimating the future values and abnormalities in multiple vital signs. The framework for the proposed CDSS is presented in Figure 5.2. The processes involved in this architecture are continuous data collection, data pre-processing and feature extraction, classification, personalized knowledge discovery, anomaly detection and clinical decision support.

5.3.1 Patient data collection

We consider an assisted living system where a patient lives alone in his/her home. Several body sensors are attached to patient’s body that collect data of various vital signs continuously (i.e. per minute). The measured data are sent to a mobile device using wireless connection (e.g. wifi, blue-tooth, zigbee). The mobile device then transmit these vital signs data to the cloud in small batches (e.g. every 1 hour) for processing. The cloud has vast storage and high processing capability. Therefore, it can store a large amount of incoming data from many patients [81, 85]. The pre-processing and machine learning steps are also performed inside the cloud environment for parallel and fast processing.

5.3.2 Correlations of vital signs for clinical inference

In this study we considered six vital signs collected as numerical trend data from the time-series data of corresponding six different bio-signals listed in Table 5.1. These vital signs measure different physiological functions of human body and are used to monitor patient’s clinical status
The framework for clinical decision support system (CDSS) 114

Patient with wearable sensors
Multiple vital sign time-series data
Stored batches (data) in the cloud

Data transmission in continuous small batches
Continuous data from Patient

Clinical alerts to patient

Data Preprocessing and feature generation
- Data cleaning
- Segmentation using fixed-size window
- Feature extraction and class labeling

Multi-label classification and evaluation
- Training and validation using several multi-label learners in parallel
- Classifier evaluations
- Predicted mean values of vital signs

Feature vector with multi-label class

New patient-specific knowledge

Clinical decision support

Clinical Inference

Diagnostic decisions
Healthcare professional

Clinical alerts/recommendations

Disease symptoms
Clinical emergency
Anomaly detection

Clinical rules
General Knowledge

Discovered Knowledge
Thresholds and Rules

Vital signs can be abnormal in virtually any disease process. Patients who are actually ill, likely to have several abnormal vital signs. Certain patterns of abnormalities develop through a strong correlations between multiple vitals. For example, a strong correlation between Hy-
pertension (elevated SBP and DBP) and Bradycardia (low HR) can create serious clinical emergency known as Cushing reflex. Fever (increasing body temperature) is accompanied by tachycardia (increasing heart rate) with the general rule of thumb that the heart rate will increase by 10 beats per minute for every 1 °C increase in body temperature. The progression of such diseases can be inferred prior to emergency situation using our proposed learning model as it uses the correlated features and estimate future values of vitals by continuous monitoring.

### 5.3.3 Data preprocessing and feature generation

In the pre-processing stage, noisy data are cleaned and then segmented into fixed-sized sliding window for calculating statistical and correlated features and corresponding class labels. The feature extraction process from raw data is described below.

![Figure 5.3: Observation, lead and prediction window time](image)

Given the continuous value of \( p \) vital signs sampled in per minute as time series \( V(t) = [V_1(t), V_2(t) \ldots V_p(t)] \). Let, \( V(T_s \rightarrow T_e) \) is a batch (e.g. 24 hours) of continuous data that starts at time \( T_s \) and ends at time \( T_e \). Data between time \( T_s \) and \( T_e \) are divided in three time slices: observation time \( t_o \), lead or forecast time \( t_l \) and prediction time \( t_p \). To generate \( o \) number of samples from this batch \( t_o \) is divided into \( o \) windows of size \( w \). That is, \( t_o = o \times w \). A feature vector, \( f_j \) is generated from each of \( o \) windows where \( 1 \leq j \leq o \). The corresponding multi-label class vector \( c_j \) of size \( p \) is obtained from the mean values of each \( p \) vitals from the prediction window of same size \( w \) located at the time after \( l \) times of \( w \) from the point where the \( j \)-th observation window \( t_{o_j} \) ends. This formulation is shown in Figure 5.3.

Let, we have numerical trend data of \( p \) vitals between time \( t - t_o \) to \( t \), the goal is to estimate future value of those \( p \) vitals at time \( t \). That is, we want to predict what the average value
The framework for clinical decision support system (CDSS)

of those vitals will be in between time \( t + t_l \) to \( t + t_l + t_p \). The prediction is not a continuous value, but instead is a vector of size \( p \) where each value is defined in one of 15 different ordinal classes. Table 5.2 summarizes what each class label stands for.

Table 5.2: Normalized class value for different ranges of six vital signs

<table>
<thead>
<tr>
<th>Class Value</th>
<th>HR</th>
<th>SBP</th>
<th>DBP</th>
<th>MBP</th>
<th>RR</th>
<th>SpO2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>( \leq 30 )</td>
<td>( &lt; 60 )</td>
<td>( &lt; 40 )</td>
<td>( &lt; 30 )</td>
<td>( &lt; 5 )</td>
<td>( &lt; 83 )</td>
</tr>
<tr>
<td>-6</td>
<td>( &gt; 30 ) AND ( \leq 40 )</td>
<td>( \geq 60 ) AND ( &lt; 60 )</td>
<td>( \geq 30 ) AND ( &lt; 35 )</td>
<td>( \geq 40 ) AND ( &lt; 50 )</td>
<td>( \geq 5 ) AND ( &lt; 6 )</td>
<td>( \geq 83 ) AND ( &lt; 85 )</td>
</tr>
<tr>
<td>-4</td>
<td>( &gt; 40 ) AND ( \leq 45 )</td>
<td>( \geq 60 ) AND ( &lt; 70 )</td>
<td>( \geq 35 ) AND ( &lt; 40 )</td>
<td>( \geq 50 ) AND ( &lt; 55 )</td>
<td>( \geq 6 ) AND ( &lt; 7 )</td>
<td>( \geq 85 ) AND ( &lt; 87 )</td>
</tr>
<tr>
<td>-3</td>
<td>( &gt; 50 ) AND ( \leq 55 )</td>
<td>( \geq 70 ) AND ( &lt; 80 )</td>
<td>( \geq 40 ) AND ( &lt; 45 )</td>
<td>( \geq 55 ) AND ( &lt; 60 )</td>
<td>( \geq 7 ) AND ( &lt; 8 )</td>
<td>( \geq 87 ) AND ( &lt; 89 )</td>
</tr>
<tr>
<td>-2</td>
<td>( &gt; 55 ) AND ( \leq 60 )</td>
<td>( \geq 80 ) AND ( &lt; 90 )</td>
<td>( \geq 50 ) AND ( &lt; 55 )</td>
<td>( \geq 60 ) AND ( &lt; 65 )</td>
<td>( \geq 8 ) AND ( &lt; 9 )</td>
<td>( \geq 89 ) AND ( &lt; 91 )</td>
</tr>
<tr>
<td>-1</td>
<td>( &gt; 60 ) AND ( \leq 70 )</td>
<td>( \geq 100 ) AND ( &lt; 110 )</td>
<td>( \geq 55 ) AND ( &lt; 60 )</td>
<td>( \geq 70 ) AND ( &lt; 75 )</td>
<td>( \geq 9 ) AND ( &lt; 10 )</td>
<td>( \geq 91 ) AND ( &lt; 93 )</td>
</tr>
<tr>
<td>0</td>
<td>( &gt; 70 ) AND ( \leq 90 )</td>
<td>( \geq 110 ) AND ( &lt; 120 )</td>
<td>( \geq 60 ) AND ( &lt; 65 )</td>
<td>( \geq 80 ) AND ( &lt; 85 )</td>
<td>( \geq 10 ) AND ( &lt; 12 )</td>
<td>( \geq 93 ) AND ( &lt; 95 )</td>
</tr>
<tr>
<td>1</td>
<td>( &gt; 90 ) AND ( \leq 100 )</td>
<td>( \geq 120 ) AND ( &lt; 130 )</td>
<td>( \geq 80 ) AND ( &lt; 90 )</td>
<td>( \geq 100 ) AND ( &lt; 105 )</td>
<td>( \leq 15 ) AND ( \leq 17 )</td>
<td>( \leq 17 ) AND ( \leq 19 )</td>
</tr>
<tr>
<td>2</td>
<td>( &gt; 100 ) AND ( \leq 120 )</td>
<td>( \geq 130 ) AND ( &lt; 140 )</td>
<td>( \geq 90 ) AND ( &lt; 100 )</td>
<td>( \geq 105 ) AND ( &lt; 110 )</td>
<td>( \leq 17 ) AND ( \leq 19 )</td>
<td>( \leq 17 ) AND ( \leq 19 )</td>
</tr>
<tr>
<td>3</td>
<td>( &gt; 120 ) AND ( \leq 140 )</td>
<td>( \geq 140 ) AND ( &lt; 150 )</td>
<td>( \geq 100 ) AND ( &lt; 105 )</td>
<td>( \geq 110 ) AND ( &lt; 115 )</td>
<td>( \leq 19 ) AND ( \leq 21 )</td>
<td>( \leq 19 ) AND ( \leq 21 )</td>
</tr>
<tr>
<td>4</td>
<td>( &gt; 140 ) AND ( \leq 160 )</td>
<td>( \geq 150 ) AND ( &lt; 160 )</td>
<td>( \geq 105 ) AND ( &lt; 110 )</td>
<td>( \geq 115 ) AND ( &lt; 120 )</td>
<td>( \leq 21 ) AND ( \leq 23 )</td>
<td>( \leq 21 ) AND ( \leq 23 )</td>
</tr>
<tr>
<td>5</td>
<td>( &gt; 160 ) AND ( \leq 180 )</td>
<td>( \geq 170 ) AND ( &lt; 180 )</td>
<td>( \geq 110 ) AND ( &lt; 115 )</td>
<td>( \geq 120 ) AND ( &lt; 125 )</td>
<td>( \leq 23 ) AND ( \leq 25 )</td>
<td>( \leq 23 ) AND ( \leq 25 )</td>
</tr>
<tr>
<td>6</td>
<td>( &gt; 180 ) AND ( \leq 200 )</td>
<td>( \geq 190 ) AND ( &lt; 210 )</td>
<td>( \geq 115 ) AND ( &lt; 120 )</td>
<td>( \geq 130 ) AND ( &lt; 140 )</td>
<td>( \leq 25 ) AND ( \leq 26 )</td>
<td>( \leq 25 ) AND ( \leq 26 )</td>
</tr>
<tr>
<td>7</td>
<td>( &gt; 200 )</td>
<td>( \geq 210 )</td>
<td>( \geq 120 )</td>
<td>( \geq 140 )</td>
<td>( \geq 25 )</td>
<td>( \geq 25 )</td>
</tr>
</tbody>
</table>

As an example, if \( w \) is 10 minutes and \( l \) is 6 then \( t_l \) is \((6\times10=60 \) minutes\) 1 hour. \( t_p = w \), that is 10 minutes. If a batch size \((T_n-T_s)\) is 24 hours \((1440 \) in minutes\) then the size of \( t_o \) can be at most \((t_o=1440-60-10)\) 1370 minutes. Thus, here number of observations \( o \) is 137 (as \( o = t_o \div w \)). Therefore, using \( w=10 \) and \( l=6 \) we can extract total 137 examples from 24 hours data. If we use two-third data for model training then we will have 92 samples for training and 45 samples for model validation.

5.3.4 Multi-label classification (MLC) and evaluation

When adequate number of instances are available for training, they are sent to the classification engine where different MLC algorithms are applied in parallel using cloud platforms. Finally, all results are compared using various evaluation measures to evaluate their performance and the best model is determined. The model is then used to classify new unknown instances of future batches. Thus, the model can predict the normalized class value (as described in Table 5.2) of all vitals in advance.

5.3.5 Knowledge adaptation

In our design, the learned model for a specific patient is also continuously adapted for new batch of data to ensure that the model is up-to-date to detect future behaviours using most
recent information. The learned personalized knowledge are stored and utilized along with different clinical rules and correlations to detect anomalous situation.

5.3.6 Clinical decision support

The abnormal mean values in one or multiple vitals indicate different types of abnormalities, disease symptoms or clinical emergencies. The healthcare professionals are notified on occurrence of such anomalies. They can then further investigate the data and make diagnostic decisions. An appropriate clinical alert is sent to the patient based on the decision so that the patient get notified before any potential danger.

5.4 Research methodology and implementation

In this section research methodology and implementations are described.

5.4.1 Data sources

We have used vital sign data from MIMIC [193] and MIMIC-II [152] numerical dataset of MIT Physiobank archive [181]. Data in the MIMIC/MIMIC-II database contain multi-parameter recordings, which are obtained from both bedside monitors and the medical records of the patients stayed in ICUs. We preferred to use this dataset because it fulfilled the criteria to evaluate our implementations. Moreover, there is no public dataset available which contain multiple vital sign data of various home-monitoring patients with different correlations for a long period of time. Here we considered that, home-monitoring data have similar nature when they are collected in a controlled environment and in supervision of a nurse.

MIMIC and old version of MIMIC-II dataset contain records of various physiological signals of about 4000 adult patients. Most of the data are sampled per minute. Some are sampled per second and those are converted to per minute sampling by taking the mean value in a minute. The data those contain clean values of the six vital signs for more than 24 hours are considered. Finally, 30 from MIMIC and 55 from MIMIC-II in total 85 patient records are used for evaluations. Patients involved in this study have a wide range of clinical problems such as sepsis, respiratory failure, congestive heart failure, pulmonary edema, myocardial infarction, cardiogenic shock and acute hypotension. Most of these clinical cases occur due to abnormalities in multiple vital signs at the same time.
5.4.2 Data cleaning and preprocessing

Like other real-world databases, a few pre-processing steps are required to improve the data quality before computing the features. This is practical for real-world data. Even for a monitored patient in home, data will contain noise and outliers. The noisy or missing data can occur due to sensor errors, disconnections, equipment changes, network connection interruptions and many other reasons. If all vital signs data are missing for a long period, they are considered as non-recoverable due to network interruptions or sensor errors and thus deleted. On the other hand, the case where one or more vital signs data are missing while clean values of others are available, was considered as recoverable and imputed using median-pass [194] and k-nearest neighbour filter [195].

5.4.3 Segmentation

For each patient we start with a batch of first 24 hours data. The first batch is used as bootstrap learning for building the initial model. The incoming data are classified using this model and after every 2 hours the model is refreshed using last 24-hours batch data to ensure adaptability. This process considers that the upcoming future values are vastly depend on most recent past values (in our case 24 hours) which is also true for real life data. Thus, this incremental batch based segmentation and learning process is able to handle potentially infinite amount of data.

In our data, vital sign signal is sampled per minute. Therefore, a batch of 24-hour data contain $24 \times 60 = 1440$ minutes samples of six vital signs. As described in problem formulation, here $T_e - T_s = 1440$ minutes. According to Figure 5.3 this 1440 minutes data are divided into 3 time slices ($t_o$, $t_l$ and $t_p$). The value of $l$ and $w$ are varied and corresponding feature vector and class labels are generated. For training we have used 66%-split that is, first two-third data are used for model training and last one-third are used for model validation.

5.4.4 Feature extractions and class labelling

The filtered numerical trend data of each vital sign (as in Table 5.1) are used to calculate the features. For a observation window size $w$ statistical features of each six signal such as mean, median, minimum, maximum, standard deviation, skewness, kurtosis, percentiles and inter-percentile ranges are calculated [196]. Minimum and maximum contains the information about extreme values, mean and median represent the magnitude of each vital sign, standard deviation describes the variability and skewness is the third moment of amplitude distribution. Moreover, different percentiles (5th, 10th, 50th, 90th and 95th), IPR (inter-percentile range,
Research methodology and implementation

which is the difference between 2 percentile values such as 95th and 5th), and kurtosis are other important statistical measures that provide a snapshot of how the relation between vital parameters varies over time.

We also calculated the sequential trend of each vitals which is the number of increasing and decreasing values within a sliding window \( w \) \[197\]. Another feature is the regression slope which is calculated by fitting a liner least square regression line to the small curve of \( w \)-sized window. The slope determines sharp changes in a vital sign which can indicate a dangerous situation. Moreover, we have measured the pairwise correlation coefficients of six signals that contain the information about actual correlation between a pair of vitals. All these measures are simple and very easy to implement. Overall, total 123 features are computed from six vital signs.

To find the corresponding multi-label classes of each feature set, we measured the mean values of \( w \)-sized window located after \( l \times w \) minutes from the observation window. Afterwards, the measured mean values of each six vital sign are labelled to the normalized value between -7 to 7 ( for SPO\(_2\) it is from -7 to 0 ) as described in Table 5.2. Therefore, each instance of a classifier is a set of 123 features and corresponding set of 6 classes.

As described above, The overall training examples are generated as a batch of 24 hours data. Therefore, the system generate an incremental batch of 24-hour training samples and they are fed to different multi-label classifiers.

5.4.5 Multi-label classification

Multi-label classification problem corresponds to searching for a function \( h \) that assigns to each instance. The goal is to minimize the expected prediction loss with respect to a specific loss function. An instance is represented by a vector of \( m \) features or attributes \( \mathbf{X} = (X_1, X_2, X_3, ..., X_m) \) and a vector of \( d \) output labels \( \mathbf{Y} = (Y_1, Y_2, Y_3, ..., Y_d) \). The \( h \) function should assign to each instance \( \mathbf{X} \) that finds the most likely combination of class labels, that is, \( \arg \max_{Y_1, Y_2, ..., Y_d} P(Y_1 = Y_1, Y_2, ..., Y_d | \mathbf{X}) \). The \( m \)-dimensional input space is represented by \( \Omega_X = \prod_{i=1}^{m} \Omega_{X_i} \). In our case all the features are numeric. So, \( \Omega_{X_i} \subseteq \mathbb{R} \). A multi-label dataset with \( N \) training samples is represented by, \( D = \{(x^{(1)}, y^{(1)}), ..., x^{(N)}, y^{(N)}\} \) where \( x^{(i)} \in \Omega_X \) and \( y^{(i)} \subseteq Y \) for all \( i \in \{1, ..., N\} \). As stated above, in our formulation, \( m \) is 123 and \( d \) is 6. The learning task of obtaining \( h \) is represented by, \( h : \Omega_{X_1} \times \Omega_{X_2} \times ... \times \Omega_{X_m} \rightarrow y \subseteq Y \).

Multi-label classification problem can be categorized in two steps, (1) Problem transformation method and (2) Algorithm adaptation method \[89\]. The first one transform the learning
Research methodology and implementation

120

task into one or more single label classification tasks. They are algorithm independent. Second method extends existing machine learning algorithms (e.g. decision tree, support vector machine [198], neural network, k-nearest neighbour [199]) to handle multi-label data directly. There are various problem transformation methods such as binary relevance(BR)-based methods, label combinations (LC)-based methods, pairwise methods, ranking methods via single label learning and ensembles methods.

Many variants of these problem transformation methods are described in the literature [89]. We evaluated our data with most of these methods and picked the best eight methods based on the evaluation measures and training time to interpret our results. The eight methods are binary relevance (BR), classifier chain (CC), Bayesian CC (BCC), Monte Carlo optimization of CC (MCC), Classifier Trellis (CT), Fourclass Pairwise (FW), Pruned Sets (PS) and Ranking + Threshold (RT). The classification algorithms are performed in MEKA software [200] which is a Multi-label extension of popular data mining software WEKA [156]. A short description of these algorithms are presented below.

5.4.5.1 Binary relevance(BR)

The BR method transforms the original dataset into \( c \) datasets (where \( c \) is total number of classes in a dataset), one for each class label, where each dataset includes all the instances of the original dataset and learns one binary classifier for each label independently of the rest of labels. To classify a new instance, BR outputs the union of the labels that are predicted by the \( c \) classifiers. It does not consider label relationship [201]. In our case, we have 6 classes with 15 different labels. Thus, the class labels of our problem are easily transferable to BR method.

5.4.5.2 Classifier chain(CC)

The CC method contains classifiers which are linked along a chain, where each classifier handles the binary relevance problem associated with each label. It creates a chain of classifier \( C_1, C_2, ..., C_L \), where \( L \) is the total number of labels. To classify a new instance, CC starts from \( C_1 \) and runs down along the chain. Each classifier determines the probability of being classified into \( L_1, L_2, ..., L_L \). The chain method passes label information between classifiers to take into account label correlation. It combines the advantages of binary relevance and label dependency. The CC method is based on the decomposition of the conditional probability of the class vector \( \mathbf{Y} \) using the product rule of probability. \[ p(\mathbf{Y}|\mathbf{X}) = p(Y_1|\mathbf{X}) \prod_{l=2}^{L} p(Y_l|Y_1, ..., Y_{l-1}, \mathbf{X}). \] [201]
A variant of CC is Bayesian Classifier chain (BCC). The objective of BCC is to find a joint distribution of the classes $Y = Y_1, Y_2, ..., Y_d$ for a given feature set $X = (X_1, X_2, X_3, ..., X_m)$ such that, $p(Y|X) = \prod_{i=1}^{n} (Y_i|pa(Y_i)|X)$, where $pa(Y_i)$ represents the parents of class $Y_i$. A Bayesian Network (BN) is induced to represent the joint distribution. In this settings, a classifier chain can be constructed by inducing first the classifiers that do not depend on any other class and then proceed with their predecessors. Another variant of CC is Monte Carlo classifier chain (MCC) that uses Monte Carlo search method for estimating joint probability. Another classifier chain method that approximate the probability $p(Y|X)$ by maintaining a lattice structure graph known as Classifier Trellis (CT) [202]. As the class labels of our formulated feature vector have high correlations so most of the variants of classifier chains produced better predictions.

5.4.5.3 Pruned Sets (PS)

Pruned Sets (PS) is a variant of Label Power Set or Label Combination (LC) method. LC takes into account label dependency. Label power set considers each unique occurrence of a set of labels as one class.[203]. In LC each different set of labels becomes a different class in a new single-label classification task. Unfortunately, basic LC must discard any new training examples that have a labelset combination that is not one of the class labels. So, this does not suites for incremental learning. Pruned sets (PS) uses pruning to focus on core combinations. It is much better suited to this incremental learning context. PS drastically reduces the number of class-labels in the transformation by pruning all examples with infrequent label-sets. It then additionally sub-samples the infrequent label-sets for frequent ones so it can reintroduce examples into the data without reintroducing new label-set combinations. PS thus retains (and often improves upon) the predictive power of LC, while being up to an order of magnitude faster.

5.4.5.4 Fourclass Pairwise (FW) and Ranking + Threshold (RT)

These are ranking-based methods. It learns $\frac{d}{d-1}$ binary models, one for each pair of labels. Each model is trained based on examples that are annotated by at least one of the labels but not both. The FW model compares each class pair $Y_j, Y_k$ to one of the four classes 00, 01, 10, 11 with threshold. The RT method duplicates multi-label examples into examples with one label each, trains a multi-class classifier, and uses a threshold to reconstitute a multi-label classification.
5.4.5.5 Algorithm Adaptation method

In our evaluations we used three popular machine learning algorithms for result analysis. They are J48 Decision Tree (J48), Random Tree (RT) and Sequential Minimal Optimization (SMO, a simplified version of support vector machine). We have also tested some other algorithms such as Naive Bayes (NB) and Multi Layer perception (MLP) but the outcomes of those adaptation methods were not satisfactory and so not included in this study for result interpretation.

5.4.5.6 Evaluation Measure

The evaluation methods for multi-label classifications are different from those used for single-label classifications. The evaluation methods can be divided into example-based measures, label-based measures, and ranking-based measures [201]. Here we have used four measures for the performance evaluations of our experiments. These are described below.

1. Hamming score: is the accuracy for each label (class) to correctly predicted, averaged across all labels. This is the opposite of Hamming Loss which reports how many times on average, the relevance of an example to a class label is incorrectly predicted. Hamming loss takes into account the prediction error (an incorrect label is predicted) and the missing error (a relevant label not predicted), normalized over total number of classes and total number of examples.

2. Accuracy: is the ratio between the correct labels to the total number of labels for each instance, averaged across all instances.

3. F1 micro average: is the harmonic mean between precision and recall, where the recall refers to the percentage of relevant labels that are predicted, and precision refers to the percentage of predicted labels that are relevant.

4. Exact match: is the accuracy of each example where all label relevance must match exactly for an example to be correct.

5.4.6 Abnormality prediction

The classifier model we developed can detect the output label of multiple vitals at same time. In our formulation the normal ranges (according to general medical rule) of all vital signs have class label 0. We can consider values near 0 (that is 1 or -1) are nearly normal. Other than this, a high or low values in class labels are considered as abnormal, and high or low
values in multiple vitals and very high or low value in one or more vitals can be treated as dangerous situation. Thus, our system has the mechanism for early prediction of such abnormal conditions well ahead of time and send proper alert to the doctors.

5.4.7 Incremental and patient-specific learning

The best classifier is picked in terms of hamming score, F1 measure and model building time. Once a classifier model is selected for a patient using the bootstrap batch, the new incoming instances are classified using only this classifier. If the performance of classification for new instances fall below an expected threshold value (in our case hamming score < 90%) before the model is refreshed, the classifier is re-trained using most recent batch (past 24-hour data from current) to maintain the desired performance level. Otherwise, the model is refreshed in every 2 hours. It is expected that, the nature of correlations of vitals will not be same for a long duration for a specific patient. Thus, this incremental learning process is adaptive and keep the knowledge of the model up to date with new instances. Therefore, our model can be easily used in patient-specific abnormality prediction as it maintains patient-specific knowledge.

5.5 Experimental evaluations and results

To evaluate the performance of the proposed approach different MLC algorithms are applied over our experimental data. We have performed the testing for multiple patients individually and concurrently inside cloud environment. We have used multiple m3.2xlarge Elastic Compute Cloud (EC2) instances provided by Amazon Web Service (AWS) for data mining process using MEKA software tool. We have also used Amazon Simple Storage Service (S3) for storing patient data as incremental batch. All the results are presented below.

5.5.1 Selection of window size

As described in the implementation we have used fixed-length sliding window for calculating the feature vectors of all vital signs. Thus, the selection of optimal window size $w$ is important in our analysis. A short window size (e.g. 4 minutes) does not contain enough information for calculating statistics and correlations. On the other hand, a long window size (e.g. 30 minutes) is also not suitable as there is high possibility that physiological signals can fluctuate in this long duration.
In our experiment we have evaluated window size $w$ as 8, 10, 12, 16, 20 minutes for multiple patients using BR classifier. The values are selected in such a way that they can be divided by 60 minutes (1 hour). The evaluations of window size for patient a40493n is shown in Figure 5.4. From our analysis, we found that window size 10 has better performance than others. Therefore, we have used this window size for all patients for further evaluation.

### 5.5.2 Selection of forecast period

If $l$ is unit for lead time and $w$ is the window size then the forecast period (or lead period) $t_l = l \times w$. For example, for $w=10$ and $l=6$ lead period is 60 minutes or 1 hour. That is, the prediction window is located after 1 hour and our system will predict the mean value of 10 minutes prediction window 1 hour before.

We varied the $l$ value from 1 to 7 (i.e, forecast window from 10 to 70 minutes). The shorter forecast window has better prediction accuracy. Normally accuracy will degrade when lead time becomes longer. 1 hour preceding prediction window is considered fair enough for a doctor to make clinical decisions. Therefore, we have used $l$ value up to 7 for our experiment. Once again, this evaluation was performed for all 85 patients with different classifiers. The evaluation for patient a40493n using lead time 1 to 7 is presented in Figure 5.5. For all cases the achieved hamming score was satisfactory ($\geq 85\%$).
Figure 5.5: Performance for single patient for different lead time using $w=10$. Three evaluation measures from 30 days monitoring data of patient a40493n using $l$ 1 to 7. Here the multi-label classifier is BR and learning algorithm is Random Tree. For all 7 cases the classification has good hamming score value ($\geq 85\%$).

### 5.5.3 Classifier performance evaluation

As stated above, to evaluate the performance of all classifier combinations we have tested it with 8 problem transformation methods and 3 algorithm adaptation methods (in total 24 combinations) for 85 patients.

#### 5.5.3.1 Performance for a single patient

Table 5.3: Performance of different multi-label classifier using the features of random 24 hours data of patient a40215n. Here $w=10$ and $l=7$

<table>
<thead>
<tr>
<th></th>
<th>Hamming Score(%)</th>
<th>Accuracy(%)</th>
<th>Exact Match(%)</th>
<th>F1 micro average(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>J48</td>
<td>RT</td>
<td>SMO</td>
<td>J48</td>
</tr>
<tr>
<td>BR</td>
<td>92.4</td>
<td>92.9</td>
<td>93.4</td>
<td>71</td>
</tr>
<tr>
<td>BCC</td>
<td>92.3</td>
<td>92.6</td>
<td>93.4</td>
<td>70.8</td>
</tr>
<tr>
<td>CC</td>
<td>91.8</td>
<td>92.4</td>
<td>93.6</td>
<td>68.4</td>
</tr>
<tr>
<td>MCC</td>
<td>89.7</td>
<td>92.4</td>
<td>93.6</td>
<td>63.5</td>
</tr>
<tr>
<td>FW</td>
<td>93.1</td>
<td>94.1</td>
<td>93.8</td>
<td>74.3</td>
</tr>
<tr>
<td>CT</td>
<td>92.9</td>
<td>93.3</td>
<td>93.5</td>
<td>72.7</td>
</tr>
<tr>
<td>PS</td>
<td>89.1</td>
<td>91.3</td>
<td>93.7</td>
<td>60.5</td>
</tr>
<tr>
<td>RT</td>
<td>86.3</td>
<td>92</td>
<td>93.8</td>
<td>68.3</td>
</tr>
</tbody>
</table>

Table 5.3 shows the observed result for a single patient on random 24 hours data where first two-third data are used for training and rest one-third for validation. Different classifier combinations perform well in different patients. As in Table 5.3, we can see for all combinations we have hamming $>86\%$ (at least), accuracy $>60\%$, exact match $>31\%$ and F1 score $>70\%$. For this particular patient SMO has better accuracy than J48 and Random Tree for most of
the problem transformation methods. Here, FW and Random Tree combination produces the best results which has hamming score of 94.1% and accuracy 77.7%.

Figure 5.6: Hamming score of 85 patients separately using the statistics from 24 classifier combinations

According to our problem formulation the class labels are mean values of vital sign within small ranges. Therefore, it is very common for a classifier to predict something that is not exactly accurate but near the class label (e.g. HR class predicted as 2 where original class label is 1). As the metric hamming loss actually represents the average distance from the correct class labels, so a smaller class deviation that is a small value of hamming loss (or in reverse large value of hamming score) indicates a better classifier performance. Figure 5.6 shows the boxplot of the hamming score values across all 24 classifier combinations for 85 different patients used in our experiment. We can see that the most of the mean values are inside the range of 90-95%.

Moreover, our class distribution is uneven. Thus, the accuracy metric does not indicate the true performance of the classifiers. That’s why we also measure F1 score because it represents a classifier performance in case of an uneven class distribution in terms of precision and recall performance metrics. However, we also kept the accuracy and exact match measure for our evaluations to interpret the performance of the classifier with hamming score and F1 score. But here we mostly decide depending on hamming score as this is the best accuracy measure to describe our problem.

Figure 5.7 shows graphical interpretation of the classification performance of another patient in terms of 4 measures we considered. Once again, we find that for most of the combinations we have hamming score > 90%. For this patient also FW and Random Tree combination achieved the best hamming score but SMO performs well as base learner in most transformation methods. In the analysis, other than FW we can see that PS and CT also has better
Experimental evaluations and results

5.5.3.2 Performance for multiple patients

To understand the overall average performance of all classifiers for all patients we measured the average hamming score of 85 patients for different classifier combinations using random 24 hours data. The results are presented in Figure 5.8. We found that the mean hamming score is still over 90%. The best average mean is obtained for FW and Random Tree combination. This analysis proves the efficiency of prediction for our model. That is, it can still perform well if class output label varies across multiple patients.
Figure 5.9 shows another average performance chart using boxplot. Here the results of hamming score of 85 patients are summarized in terms of hamming score of 24 classifier combinations. As we also see here most of the values are between 90-95%.

![Figure 5.9: Hamming score of 24 different classifier combinations using the statistics of 85 patients](image)

5.5.3.3 Performance in terms of model building time

In a clinical decision support system it is also important to take a quick decision. Therefore, we need to measure the efficiency of our model in terms of learning time. Figure 5.10 graphically shows the average model training time for 85 patients. We can see that for most cases the average model building time is less than a second using 24-hour data. The FW method is a slow learner. SMO is slow learner than Random Tree and J48. However CT, PS and RT can learn fast using our data. PS and Random Tree combination has the minimum building time. Therefore, when we consider both hamming score and building time we can say that CT, PS and RT problem transformation methods along with Random Tree adaptation method are the best classifiers for our techniques.

5.5.4 Performance for using correlations

To understand the importance of correlation in MLC performance we have tested the same dataset that were used for MLC with our 3 adaptation methods (J48, Random Tree and SMO) to classify the output label of each vital individually with the following three settings.

1. Considering all 123 features generated from 6 vitals to predict the output label of each vital individually.
Experimental evaluations and results

Figure 5.10: Average building time for different classifier using random 24 hours data of 85 patients. PS-Random Tree is the fastest process. SMO takes more time than Random Tree and J48. FW has very high building time.

2. Excluding correlated features and considering only the statistical features of corresponding vital sign.


Table 5.4: A comparison of prediction accuracies for 6 vital signs as individual using 3 classification algorithms. This is using same dataset and settings as in Table 5.3. Here, WCF means with all correlated features of all vitals, OF means using Only features of corresponding vital and SF using subset of features selected by a feature selection algorithm from all features.

<table>
<thead>
<tr>
<th></th>
<th>J48 Decision Tree</th>
<th>Random Tree</th>
<th>SVM Polynomial Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WCF</td>
<td>OF</td>
<td>SF</td>
</tr>
<tr>
<td>HR</td>
<td>89.36</td>
<td>72.34</td>
<td>85.1</td>
</tr>
<tr>
<td>SBP</td>
<td>72.08</td>
<td>60.28</td>
<td>70.21</td>
</tr>
<tr>
<td>DBP</td>
<td>70.21</td>
<td>61.7</td>
<td>68.08</td>
</tr>
<tr>
<td>MBP</td>
<td>72.34</td>
<td>63.82</td>
<td>63.82</td>
</tr>
<tr>
<td>RR</td>
<td>69.7</td>
<td>65.95</td>
<td>72.34</td>
</tr>
<tr>
<td>SPO2</td>
<td>95.74</td>
<td>93.61</td>
<td>95.74</td>
</tr>
</tbody>
</table>

The results of this comparison for a single patient is presented in Table 5.4. The results are generated using WEKA software. We can see that the prediction accuracy is higher when we consider all the correlated features for all types of classifier. This proves the importance of using correlation in MLC. The correlated features have high impact on predicting other vitals.
5.5.5 Discussions

From the above observations, we can conclude the MLC is a better option than single-label classifiers for the developed CDSS. Because for estimating values of 6 vitals we will need to build 6 learning models using same set of features. We can see from Table 5.4 that the vital such as SPO$_2$ which have low variations have higher accuracy than the vitals those have high variations and this makes the prediction inconsistent. In case of multi-label classifier we can estimate the values of all vital using just one training model which have high hamming score. That is, using MLC we can achieve high prediction rate and low model training time which are essential for real-time patient monitoring.

Moreover, this technique can easily be used for patient-specific vital sign predictions. The learning models for each patient is obtained by training different models using bootstrap data (first 24 hours). The best learning for $w=10$ and $l=7$ is obtained using the highest hamming score and lowest model building time while trained in m3.2xlarge Amazon EC2 instance using MEKA classifier. The best model for each patient is stored in Amazon S3. Then, the future values are predicted for subsequent data. The abnormality alarm is when multiple vitals have high or low value, or one vital has very high/low value. The prediction is verified using hamming score and the same model is re-trained when hamming score goes below 90%.

5.6 Conclusion

In this chapter, we propose a model for a CDSS to predict multiple vital sign values of a home-monitoring patient using their correlations. This also helps to find patient-specific anomalies in advance. The proposed technique takes the advantage of multi-label classification. Numerical trend dataset of multiple vital signs are prepared for multi-label classification engine. Different multi-label classification methods are performed over data of many patients using MEKA software and their performance was evaluated in order to extract patient-specific knowledge and predict future abnormality. Using the experimental evaluations we showed that multi-label classification outperformed single label classification for this type of decision support system. Based on hamming score and model building time we obtained the best classifier for each patient individually. The model is also made adaptive and situation-aware using incremental learning process. Thus, patient-specific clinical decision can be made using the predicted outcome produced by the multi-label classifier in short time.

The major contribution of this work is the investigation of multi-label classification meth-
ods to forecast the future value of multiple vital signs at the same time using their correlated features. Therefore, the outcome of multi-label classification can assist the healthcare professionals in decision-making through the CDSS and thus help to detect instances when patient would be in serious clinical danger. Our model is extendible for other vital signs.

5.7 Epilogue

In this chapter we have developed a continuous and adaptive prediction model for a patient using his/her own data. However, we are also interested to build predictive systems where knowledge is developed using the observations of many similar patients. We want to examine the prediction performance of such systems when they are applied to a new patient with an unknown clinical condition. In the next chapter, we mainly focus on developing such predictive models.
Chapter 6

Early Prediction of Abnormal Clinical Events Using Known Knowledge

A predictive home-based health monitoring system can provide early notification to a patient’s doctors to avoid life-threatening diseases. With regard to the fourth research question, the focus of this chapter is to build a reliable, flexible and personalised remote monitoring system that can accurately identify incoming abnormal clinical events of home-monitored patients using the temporal correlations of multiple vital signs learned from a large number of similar patients. Changes in multiple vital signs at the same time indicate a transition of a patient’s health status; if such changes are abnormal then it may lead to serious physiological deterioration. Here, we consider the problem of predicting future clinical events of a patient without any prior knowledge of his/her clinical history. The initial knowledge about some clinical events is developed through training on a large collection of available samples from many patients which are already labelled. The separation of different clinical events is validated by the Principle Component Analysis (PCA) method. Before training, features are extracted from past observations of targeted clinical events. These features are then used for learning model development. As in Chapter 5, here also the correlations of six vital signs are considered to design the learning models. To make predictions, we build a probabilistic and a static predictor model. In this chapter we described the purpose and development of both models. We also demonstrate the effectiveness of both models on cloud platforms through comparative evaluations. The evaluations show their potential to become a new tool for predictive healthcare.
To design the models four different clinical events are identified from a large patient records of publicly available database where multiple vitals deviate from normality. Our probabilistic model is capable of performing predictions of future clinical states of a patient based on maximum likelihood probability in real-time using the learned temporal behaviours of multiple vitals from patients with known clinical situation. In this model, Hidden Markov Model (HMM) is adopted for probabilistic classification and prediction of future clinical events using temporal observations and correlations of six vital signs. HMM models are trained and evaluated using continuous monitoring data of 2000 samples. The best models are selected using expectation maximisation (EM) algorithm and used in personalized remote monitoring system to forecast the most probable forthcoming clinical states of a continuously monitored patient. In static predictor model, features are extracted prior to 1-2 hour(s) from 10-30 minutes observed data of targeted clinical events. More than 140000 samples are compiled from 4,893 patient records and different data mining algorithms are applied. The best accuracy (95.85%) was achieved for Random Forest classifier using all features. The encouraging learning performance of these models proves that the discriminatory patterns in vital signs are useful for early and accurate prediction clinical abnormalities.

The rest of this chapter is organized as follows. Section 6.1 describes the motivations and contributions of this chapter. Section 6.2 presents some related work relevant to the proposed models. Section 6.3 discusses some primary analysis about vital sign correlations and clinical events. Section 6.4 illustrates the initial setup of the system description and data preparation. Section 6.5 explains the theories or process of probabilistic model development and Section 6.6 discusses the setup and requirements of the static predictor model. Section 6.7 describes the experimental results and comparisons for both models. Finally, the chapter is summarized in Section 6.8.

### 6.1 Motivations and Contributions

#### 6.1.1 Motivations

The changes in multiple vital signs indicate several symptoms of various chronic disease that can be utilized for early diagnosis [149]. The alterations in vitals are sometime so rapid that even doctors fail to put the causes together quickly which can result in serious clinical emergency, even death. Therefore, the development of techniques for early discovery of knowledge by utilizing intrinsic patterns in a large quantities of vital signs data and scalable power of cloud
computing [81] can enable doctors and care-givers to make accurate and real-time data-driven decisions. This will have an effective impact on patient-care, diagnosis and more importantly will reduce patient morbidity and mortality, and prevent hospital outbreaks. Moreover, there have been limited research about forecasting of various clinical events using multi-parameter data of a large number of patients [76]. Most of previous researchers have done such analysis on a small number of samples. Some models have been developed to predict single parameter such as blood pressure [73] or ECG [204]. These systems cause high number of false alerts with the increment of variety in data. The analysis in this work improves the generalizing ability of learned model by considering multi-parameter data of many patients.

6.1.2 Contributions

The contributions of this paper are the following.

- We developed a probabilistic estimator using Hidden Markov Model (HMM) which can detect the clinical state of a patient using current and past data of multiple vital signs. Most of the existing systems can recognize anomalies in a single vital parameter but neglect the temporal interdependence of many vitals. The system we developed has the unique ability of employing the correlations of many vital signs to detect abnormalities. To the best of our knowledge, this method has never been applied to predict abnormalities by utilizing the knowledge of multiple bio-signals.

- Our developed probabilistic model is suitable for real-time and personalized remote monitoring in assisted living. Multiple HMMs are trained using various clinical events. The clinical events are identified and separated by PCA method. In personalised monitoring, the maximum likelihood probability of the occurrence of a clinical event in future time is continuously computed from real-time multi-parameter data. Moreover, the system is expandable in the sense that a new clinical case can be added with a new HMM training. The model is more adaptive, because HMM parameters can be updated with a new set of observations that maximize its probability. No remote monitoring system exists with such unique set of capabilities.

- We developed a static predictor model by utilizing various features that extracted from raw data of multiple vital signs from a large number of patients. Here we enhance the early clinical abnormality prediction capability by (i) efficiently extracting smart features using a combination of wavelet transform, Pearson’s correlation coefficients and short-length
statistics in all vital signs, and then (ii) applying different data mining techniques such as J48 Decision Tree (J48), Random Forest (RF), and Sequential Minimal Optimization (SMO). The best model is chosen based on accuracy and model building time. There is no previous work that attempts to aim or build such system using large scale data and discriminative features.

- We use cloud computing frameworks for learning and classification. We have tested the suitability and feasibility of both models in the cloud through experimental evaluations and showed that the system does not impose any additional penalty in terms of time and resources for learning from big data. Moreover, the utilization of cloud includes the flexibility of learning from many vital signs data of a very large number of patients with various clinical conditions. Therefore, our system has an exclusive ability of handling a large group of patients simultaneously using the cloud computing technology.

### 6.2 Related work

The interest in analysing biomedical data has grown over the last decade. One particular focus is the analysis of correlations among multiple bio-signals generated by wearable sensors for future abnormality prediction. Several studies have given emphasis on finding correlations among different biological signals [72, 74, 205, 206] such as ECG, blood pressure, heart rate, respiration and oxygen saturation. Some of these systems can identify future anomalies [72, 74, 207] in a specific vital sign. These studies are mostly at theoretical level and still far behind to implement and apply them at application level. One of the practical example is BioSign device [208] that uses stochastic process to model the multi-parameter data by fusing information of five vital signs. However, this does not contain predictive capabilities and can only minimize the time of occurrence of critical clinical situation.

Data mining techniques are already widely adopted for designing predictive and forecasting model to find health-related abnormalities and to detect symptoms of various chronic diseases [77] using various daily monitoring data such as bio-signals, and activities. Examples of such model include Support Vector Machine (SVM) [76, 78, 209], Hidden Markov Model (HMM) [88, 180, 210], Artificial Neural Network (ANN) such as Multi Layer Perceptron (MLP) [211] and Feed Forward (FF) [19], Topic Model [212], and Naive Bayes (NB) [192]. These systems can detect abnormalities and predict future behaviours in one or more vital signs. HMM models are widely used to detect abnormalities using physiological data such as blood glucose level.
estimation [213], motif discovery using ECG and accelerometer sensor data [214]. In general, HMM is used for anomaly detection rather than any other tasks and has not been applied to multi-parameter analysis.

Some recent research works show the pathway of the use of cloud platform in bio-signal processing and health data analysis [204, 215]. The novelty of using cloud for context-aware monitoring and abnormality detection was described and developed in some recent works [91, 204]. Identification of appropriate features is another challenge in biomedical data analysis to estimate future abnormality accurately. Several studies have focus on using correlations among different biological signals [72, 74, 75, 205, 206] such as ECG, blood pressure, heart rate, and respiration. Some of these systems are capable of finding future abnormalities [72, 74] and irregularities in a specific vital parameter [19, 185, 216] using specific set of features. These studies mostly use either time-domain or frequency domain related features and mostly work for limited number of bio-signals. Very few studies are involved in work that deals with various features of multiple bio-signals.

All these contributions have inspired us to develop this learning models that uses hybrid features from a large collection of vital sign data, cloud-based model, and data mining techniques which can be used to predict multiple clinical events in real-time patient monitoring.

6.3 Preliminary analysis

This section describes some preliminary analysis that has been done about vital sign correlations, clinical events and PCA method. The analyses are illustrated as follows.

6.3.1 Vital sign correlations

The progressiveness of different correlations in vital parameters may contain valuable information that can be utilized to avoid future clinical emergencies [76]. For example, when a patient is experiencing a medical health crisis, such as hemorrhage, stroke, or heart attack, the heart beat may slow down due to the failure of the heart muscle to contract or too much blood has been lost. Dangerous cardiac arrhythmias may cause long pauses in the contraction of the heart muscle and blood pressure (BP) drops rapidly causing the patient to faint. In such emergency cases, heart rate (HR) is directly dependent on the BP and vice versa. In some cases both HR and BP can rise at the same time. For example, physical activity such as exercise increases the cardiac output and hence in BP and HR. But after certain period both
goes back to normal rate. Generally, a healthy heart will support HR and BP independently and the two measurements do not correlate with each other.

It is difficult to tell about patient’s actual clinical situation without testing all vital signs. Therefore, patient’s multiple vital signs should be monitored continuously under a doctor’s remote supervision. An example of strong correlation between HR and BP is shown in Figure 6.1 and 6.2. BP value increase in Figure 6.1(a) and decrease in Figure 6.1(b) with the increment in HR. From the bottom scatter plots we can see that there is a strong positive correlation between BP and HR in (a) and negative correlation in (b). Figure 6.2 scatter plot of correlations between HR-SBP after combing data of many similar patients. Figure 6.3 shows a 3D correlation among HR, mean BP (MBP) and respiratory rate (RR) values of a single patient. Here all 3 vitals goes high and low at the same time in most of the cases.

Figure 6.1: High correlation between HR and BP. The top 2 plots show the time-series, changes in HR, Systolic BP (SBP), Diastolic BP (DBP) and mean BP (MBP) with time (in minutes). The bottom 2 plots show the correlation between HR and MBP from corresponding upper plots. Here strong (a) positive and (b) negative correlation is shown between HR and MBP.

Figure 6.2: Scatter plot of correlations between HR-SBP and HR-MBP pairs after combing data of many similar patients. Warmer color indicate higher density.


Figure 6.3: Correlations among HR, MBP and RR in a single patient. Low value in colour bar indicate all vitals have high value and vice versa. Depending on correlations among 3 vitals patients can be in different clinical states.

6.3.2 Clinical events

As described in Table 5.1 of previous chapter, here we also used numerical trend data of six vital signs (HR, SBP, DBP, MBP, RR and SPO$_2$). To decide about BP value information is combined from either SBP, DBP combination or from MBP. The normal ranges and the threshold values of abnormalities for adults in different vital signs described by general medical practice are shown in Table 6.1 [148, 149]. The clinical name of different abnormalities are also listed along with their numerical ranges.

<table>
<thead>
<tr>
<th>Table 6.1: Threshold values of various clinical condition (CC)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CC</strong></td>
</tr>
<tr>
<td>Normal HR</td>
</tr>
<tr>
<td>Tachycardia</td>
</tr>
<tr>
<td>Bradycardia</td>
</tr>
<tr>
<td>Normal BP</td>
</tr>
<tr>
<td>Hypertension</td>
</tr>
<tr>
<td>Hypotension</td>
</tr>
<tr>
<td>Normal RR</td>
</tr>
<tr>
<td>Tachypena</td>
</tr>
<tr>
<td>Bradypena</td>
</tr>
<tr>
<td>Normal SPO$_2$</td>
</tr>
<tr>
<td>Hypoxia</td>
</tr>
</tbody>
</table>

When, the clinical conditions described in Table 6.1 occur instantly, periodically, or continuously and also simultaneously then these are considered as serious clinical emergency and
Preliminary analysis

thus requires quick attention of clinicians. Concurrent abnormal changes in multiple vitals cause dangerous clinical condition such as respiratory failure, heart attack, and myocardial infarction. For this study, we restricted our focus on simultaneous changes in four health parameters (HR, BP, RR, SPO$_2$) from generalized normal values and developed techniques for prior predictions of those changes. Based on data availability the clinical events considered in our analysis is listed in Table 6.2.

Table 6.2: The occurrence of our targeted clinical events for evaluation, their acronyms and class labels.

<table>
<thead>
<tr>
<th>Clinical event</th>
<th>Acronym</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>All six bio-signals are in normal range</td>
<td>NNNN</td>
<td>0</td>
</tr>
<tr>
<td>Simultaneous Tachycardia, Hypotension, Tachypena and Hypoxia for 30 minutes</td>
<td>TTHH</td>
<td>1</td>
</tr>
<tr>
<td>Simultaneous Bradycardia, Hypotension, Tachypena and Hypoxia for 30 minutes</td>
<td>BHTH</td>
<td>2</td>
</tr>
<tr>
<td>Simultaneous Tachycardia, Hypertension, Tachypena and Hypoxia for 30 minutes</td>
<td>TTTH</td>
<td>3</td>
</tr>
<tr>
<td>Simultaneous Tachycardia, Hypotension, Bradypena and Hypoxia for 30 minutes</td>
<td>THBH</td>
<td>4</td>
</tr>
</tbody>
</table>

In Chapter 3, we proved that the thresholds of vital signs values are specific to patient’s context. However, here our focus is to describe the concepts, developments and application areas of the models. Thus, for simplification we have only used the generalized normal and abnormal ranges of vital signs listed in Table 6.1 to describe normality and abnormality and defined five type of clinical events as in Table 6.2. We also call them clinical state, clinical class or clinical episode. The duration of a clinical episode is considered for 30 minutes using similar concept defined for detecting acute hypotension episode in physionet CinC challenge 2009 [217].

6.3.3 Principle Component Analysis (PCA)

Here we consider Principle Component Analysis (PCA) to capture and validate the variations in multiple vitals and isolate clinical events (or, clinical classes) for supervised learning from a large number of patients. PCA separates normal and various abnormalities in multi-parameter data based on their correlated behaviours using principal component features. This process is shown in Figure 6.4. Continuous monitoring data contain various noises and PCA is expected to provide features that are robust to noise because it retains maximum variance. PCA mainly
use to analyse spatial data. However, here we showed that PCA is also a useful tool to clearly separate temporal data with multiple interdependent signals.

Figure 6.4: Validation of clinical event separation from a large number of patients using PCA

6.4 System setup

Our objective is to build a highly accurate and efficient system using wide range of features and learning models that identify future clinical events of patients with unknown clinical conditions. Before we move forward to actual technical details of learning model development in this section we describe some primary setup of overall system and data preparation for experimental evaluation.

6.4.1 Model description

The concept of this system based on CoCaMAAL model proposed in Chapter 2 is visualized in Figure 6.5. The physiological conditions of a patient in AAL are continuously monitored by remote health monitoring stations. A number of wearable sensors are attached to patient’s body that collect the information of different vital signs in form of bio-signals, and then send the collected data to a portable device (e.g. smart phone). The portable device transmits these physiological data to the cloud in a small batches for processing. The cloud runs different learning algorithms with the help of various frameworks for data cleaning, segmentation, feature extraction, learning and knowledge discovery. The learned models of different clinical events are stored in cloud repositories for future classifications and predictions. The new incoming data from patients are then classified using the stored models. The suitable models are used
for measuring future clinical states. If the system can predict the occurrence of an abnormal clinical event in near future, it sends notifications to the appropriate monitoring service. The monitoring person (e.g. doctor, nurse) can make diagnostic decision based on patient’s clinical situation and send proper warning to the patient.

![Diagram of system setup](image)

**Figure 6.5:** The scenario of the developed system showing its components.

### 6.4.2 Dataset description

For our experimental evaluations we have used MIMIC-II [152] database of MIT Physiobank archive [181]. Version 3 of MIMIC-II database contains records of 23,180 ICU patients, equivalent to 3 TB in size. This database consist of physiological signals and vital signs time-series data captured from patient monitors of different ICUs (medical, surgical, coronary care, and neonatal). A subset of this dataset contains 5,266 numeric records of multiple vital sign time-series data which is actually used for our analysis. We have selected this database because it meets our criteria for learning model development and evaluations. In home-based remote monitoring, data are collected continuously using wearable sensors just like controlled environment in ICU. Moreover, there is no available public database which have such a big collection of vital sign data of varieties of patients with different clinical events.

The numerical trend data of MIMIC-II database contain long-term collection of various vital signs such as blood pressure (systolic, diastolic, and mean), heart rate, cardiac output, carbon dioxide, respiration rate, and oxygen saturation. The records those contain at least 24 hours numerical trend data of six bio-signals (HR, SBP, DBP, MBP, RR and SPO2) fulfill our initial criteria and so they are utilized for analysis. Majority of these records are sampled per minute. Some are sampled per second and those are converted to per minute sampling by taking the numerical mean value in a minute. The data with consecutive missing values
over long period are eliminated and noisy data are filtered using median-pass [194] and nearest-neighbour filter. Finally, 4,893 patient records meet the criteria which contain clean continuous monitoring data of six bio-signals for more than 24 hours. From each selected patient record one or more examples are compiled. For the probabilistic model total 2000 samples are used. For static predictor model nearly 140,000 examples are created for three different values of forecast period (60, 90 and 120 minutes).

6.4.3 Data formulation

To perform reliable online classification we developed offline learning methods using a large number of samples. Let, each patient record contains minute-by-minute observation of \( p \) vital signs for several hours. For each patient record we perform a linear search to identify the clinical events with 30 minutes duration as listed in Table 6.2 using described values in Table 6.2. We call this duration, prediction window \( T_P \). From each patient record one or more such 30 minutes examples are extracted. Let, in total we extracted \( r \) examples. Therefore, from all patient records we form a matrix \( X \) with \( n \) rows and \( p \) columns where \( n \) is the number of samples after combining 30 minutes samples of all examples. To avoid uneven class distribution and biased learning \( X \) is formed using same number of examples of each clinical events.

\[
X = \begin{bmatrix}
  x_{11} & x_{12} & \cdots & x_{1p} \\
  x_{21} & x_{22} & \cdots & x_{2p} \\
  x_{31} & x_{32} & \cdots & x_{3p} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{n1} & x_{n2} & \cdots & x_{np}
\end{bmatrix}
\]

If, \( L \) is number of targeted clinical events (which is 5 in our case) and equal number of examples \( s \) are extracted for each sample then \( n = L \times s \times 30 \) and \( r = L \times s \). Similarly, another matrix \( Y \) of dimension \( m \times p \) is constructed using 1 hour (60 minutes) data immediate before each clinical event example considered in \( X \). This period is the observation window \( T \) for HMM-based probabilistic model. Thus, \( m = L \times s \times 60 \). The occurrence of a clinical episode depends on the sequence of past data of that episode. Moreover, this episode is also a result of correlated behaviour of multiple vitals. Therefore, the clinical events and their past data have distinguishable features and they should be separable. To validate these statements we have applied PCA on matrix \( X \) and \( Y \) and interpreted the results using 3D scatter plots.
6.4.4 Analysis using PCA

In data mining, PCA is generally used for dimensionality reduction in a dataset with a large number of attributes. The output of PCA algorithm is a set of artificial variables known as principal components. Generally, first few components contain most of information of all attributes and can be utilized to cluster sample data points in multiple categories. We form matrix $X$ and $Y$ by taking equal number of samples for 4 abnormal clinical events and normal case scenario listed in Table 6.2. Then we apply PCA on $X$ and $Y$.

By applying PCA we first generate the centred input around its column mean. Next, we applied Singular Value Decomposition (SVD) to derive eigenvectors and eigenvalues for the centred matrix, which is then rearranged as a new matrix starting with the eigenvector that corresponds to the highest eigenvalue, and so on. Eigenvector is $p \times p$ matrix (in our case $p$ is number of bio-signals which is 6). Finally score matrix is calculated which produce a matrix of dimension $n \times p$ for $X$ and $m \times p$ for $Y$.

![Figure 6.6: Separation of 5 clinical events by PCA method](image)

To interpret the obtained results we plotted 3D scatter graph by taking values from score matrix using first 3 principal components (PC1, PC2 and PC3). We obtained the result in Figure 6.6 for $X$ and in Figure 6.7 for $Y$. PC1, PC2 and PC3 is taken along X,Y and Z axis. It is clear from Figure 6.6 that different clinical events can be separated using multiple vital sign values which are shown in different colours. Therefore, we used them as separate classes in designing the learning models.

In Figure 6.7 where data is taken immediately before the clinical event, we can see that PCA can clearly separate normality and abnormality. Colours are used to show data of 5 clinical cases. We also observed a small percentage of overlapping cases for $Y$. For this reason,
we need to derive features and train the model so it can easily separate different clinical cases using the past data. These features extracted from data in Y. Thus from our analysis using PCA, we can conclude that data preceding the clinical event contain useful information and can be used as training data in learning model development.

![Figure 6.7: Separation of 5 clinical events using observed data before that event by PCA method](image)

6.4.5 Training and Learning

The proposed models first learn about various clinical events from vital sign data of many patients. The behaviours and correlations of multiple vital signs preceding the targeted clinical events are used as features for model training. In next step, the models are trained in the cloud using large samples of normal and abnormal observations from patients with different clinical cases. The learning process for the probabilistic model is shown in Figure 6.8.

6.4.6 Learning in the cloud

For both models training phases are performed in cloud servers. In case of probabilistic model, training a HMM with a large amount of observation samples is computationally expensive. Training algorithm such as BaumWelch uses expectation maximization (EM) [173] process for maximum probability estimation. The EM learning is slow if it is conducted in a local device. A real-time monitoring system demands quick learning.

Using cloud platforms such as Amazon Web Service [40] it is possible to create and configure high performance virtual machine (Amazon EC2) instances at low cost and store a large amount of data (in Amazon S3 buckets [40]). In our approach for HMM model learning, large
data samples from many patients are stored in Amazon S3 storage buckets. For supervised learning the data with similar clinical conditions are grouped together and stored in Amazon S3. When sufficient amount of samples for each target clinical event become available, the steps of computing features and HMM training are run on high performance virtual machines configured using Amazon EC2. Apache Mahout [218], a scalable machine learning library has procedures for HMM training and classification. For $L$ clinical events $L$ HMMs are trained in parallel in $L$ virtual machines (Amazon EC2 instances). Therefore, we get the estimated parameters of all the $L$ HMM models within a very short time. The learned parameters of all HMMs are stored in Amazon S3. This scenario was described in Figure 6.8. The cloud platforms simplify HMM learning process and make computation faster.

For static predictor model, the training steps are performed using WEKA machine learning toolkit on the cloud. Most WEKA classifiers require the entire dataset to be loaded in main memory for training. When dataset become very large it is difficult to train WEKA classifier on a machine with standard configurations. Therefore, each classifier is trained in parallel in high powerful virtual machines on same training and test data and then the best model is selected.

### 6.4.7 Real-time and personalised monitoring

The proposed techniques can be used in real-time home-based health monitoring system to classify the streaming data of multiple vitals of a particular patient. The continuous data of a new unknown patient are sent to the classification engine inside the cloud and continuous classification results and predictions are obtained over time. The scenario for the probabilistic
The PEACE-Home model

model is shown in Figure 6.9.

Figure 6.9: Real-time clinical abnormality prediction for a patient with unknown clinical condition using multiple vital sign data. Here CE means a clinical event. The current observations are matched with stored HMMs and the probabilities of matching are continuously estimated.

6.5 The PEACE-Home model

We named the probabilistic model as PEACE-Home (Probabilistic Estimation of Abnormal Clinical Events in Home-based monitoring). A probabilistic estimator is developed using HMM to recognize clinical conditions of unknown patients. HMM is a statistical model used very successfully and efficiently in speech processing [86]. This model is used in some areas of bioinformatics such as human activity classification [180], gene prediction [219], and DNA motif discovery [220]. It is a general model for stochastic process and thus be applied to a large variety of biomedical signal processing [221]. One type of stochastic process is Markov chain process which is a finite state machine consisting of a set of \( N \) distinct states spaced by time instances. It may stay same state or switch to another state according to some transition probabilities. Markov chain model is called observable Markov model, since each state corresponds to a unique observable event characterized from bio-signals by the process. That is, here a state can be determined directly from signal characteristics. In HMM, the observations are some probabilistic functions of the states and same observation may be generated from various states
The PEACE-Home model

with different probabilities. Here states are no longer observable and hidden from the observer because it is not possible to determine current state directly from observation like observable Markov process. The steps are described below.

6.5.1 Mapping bio-signals to observations

The first step of constructing an HMM $\lambda$ is to convert $p$ vital sign time-series data before the targeted clinical events (observation window) to a discrete sequence of observations, $O = (O_1, O_2, ..., O_M)$. Let, data of $p$ vitals obtained for $T$ duration before the clinical event. Then $Z$ is the observed data matrix of dimension $T \times p$. If $Z$ is sub-divided into $M$ equally-sized window $w$, then $Z$ can be represented as sequence of $M$ windows $Z_1, Z_2, ..., Z_M$ where each $Z_i$ is a matrix of dimension of $w \times p$ and $w = \frac{T}{M}$. Each $Z_i$ is converted to $k$-dimensional feature vector $F_i = \{f_1, f_2, ..., f_k\}$ by computing various features such as mean, standard deviations, min, max of each $p$ vitals in $Z_i$ and their pairwise correlation coefficients. That is, each $Z_i$ is mapped to a row matrix $F_i$ with $k$-elements. Afterwards, each $k$-element in $F_i$ is normalized to numerical value between 0 to 9. Let, $\bar{F}_i$ is the normalized vector of $F_i$, then an observation $O_i$ becomes a row matrix $\bar{F}_i$. In this way, $Z$ becomes a finite sequence of observations $O = (O_1, O_2, ..., O_M)$.

The process is shown in Figure 6.10 along with HMM model parameters.

6.5.2 Parameters of HMM

If $M$ observations can be described by $N$ distinct states $S_1, S_2, ..., S_N$ then it forms a Markov chain process. In HMM, these states are called the hidden states and are not directly visible. Only the observable outputs ($O$) depended on the states are visible. The change of one state to another depends on state-transition probabilities. The transition probability from state $q_{t-1} = S_i$ at time $t - 1$ to the state $q_t = S_j$ at time $t$ is given by Equation (6.1). All these transition probabilities form matrix $A = \{a_{ij}\}$.

$$a_{ij} = P[q_t = S_j|Z(q_{t-1} = S_i)]$$

$$a_{ij} \geq 0, \sum_{j=1}^{N} a_{ij} = 1, i, j = 1, 2, ..., N$$

In order to determine the complete model the initial states should be known. The probability of being in state $S_i$ at time $t = 1$ is $\pi_i$. All such probabilities form initial probability vector $\Pi = \{\pi_i\}$. The observation $O_j$ when the system is in state $S_i$ at time $t$, depends on a observation probability distribution which form a matrix $B = \{b_{ij}\}$. They are called the
emission probabilities \((b_j)\). Using \(A\), \(B\) and \(\Pi\) it is possible to produce a state transition sequence \(Q=q_1, q_2, ... q_M\) for the given HMM, which can best explain a given observation sequence \(O=O_1, O_2, ..., O_M\) that is formed using multiple vital signs. So, an HMM is described by \(\lambda = (A, B, \Pi)\). To construct an HMM we need to first estimate these model parameters \((A, B, \text{and } \Pi)\).

### 6.5.3 Estimation of model parameters - Training

In PCA method we have used \(s\) number of samples for each of \(L\) clinical cases which formed the matrix \(Y\). In HMM training step all \(s \times L\) samples are converted to a sequence of \(k\)-dimensional vector \(\bar{F}_i\), that is a sequence of observations \(O_i\) as described in mapping step. \(T\) minutes observed data of window size \(w\) becomes \(\frac{T}{w} \times k\) dimensional matrix that represents 1 training example. In this way, we generate total \(s\) training examples for each of \(L\) clinical case. Two-third of the samples are used as training set to train \(L\) HMM models and the rest as validation set. To build an HMM-based recognition system \(L\) HMM models \((\lambda_1, \lambda_2, ..., \lambda_L)\) are trained using training samples. The HMMs represents \(L\) type of clinical cases, with one normal case and \(L-1\) clinical events (in our case \(L\) is 5). As training samples of each model are
independent and already separated in PCA-step, so $L$ models can be trained independently in parallel in the cloud.

Given a training set, the parameters of HMM models can be estimated using these training data. An efficient algorithm namely Baum-Welch algorithm [173, 174] is generally used for estimating model parameters $(A, B, \Pi)$ using a training set. The algorithm starts with a random initial guess $\lambda_0 = (A_0, B_0, \Pi_0)$ and follow an iterative process to converges to the final model $\lambda$ that yields the maximum output probability. However, for making initial guess we also need to know number of hidden states $N$ in the sequence, but it is not known. To estimate number of hidden states, we train the model from 2 to 10 number of hidden states using the same training data and compute the maximum likelihood probability. The complexity of Baum-Welch algorithm increases exponentially with the number of states. Therefore, the value of optimum $N$ is determined experimentally using maximum likelihood probability and model training time value. Since HMM models are not dependent and have separate training data, so $L$ HMM models are trained independently and in parallel in the cloud (as shown in Figure 6.8). These $L$ models are then used in recognition.

6.5.4 Validation, classification and probabilistic prediction

The classifier is formed using $L$ HMM models. The performance of the classifier is evaluated using the validation set and by computing maximum likelihood probability for a given observation. Here, the developed HMM-based classifier serves multiple purpose. First, it can tell which one of possible $L$ clinical event a patient will be in prediction window $T_P$ using $T$-minutes observed data immediately before that clinical event. Second, once it knows the possible class of an observation it can also estimate the most likely state sequence for that observation using Viterbi algorithm [175]. This process also helps to estimate future states for possible observations and thus can detect future abnormalities. Third, it can be used in real-time patient monitoring by estimating probability of every $L$ clinical events in every minute (as in Figure 6.9). At time $t$, observed sequence is generated from the data in $t - T$ continuously and $logP(O|\lambda)$ is computed for each $L$ models which outputs $L$ continuous probability estimations. Thus, at time $t$ we compute the maximum $logP(O|\lambda)$ and continuously classify patient situation in one of possible $L$ clinical events.
6.6 The ViSiBiD model

We named the static predictor model **ViSiBiD** (A learning model for early prediction of clinical events using Vital Sign as Big Data). This model is developed using data mining methods and is trained using larger number of samples and more features. This method also uses same clinical events as class described in Table 6.2. In the PEACE-Home model training data are formed from the observations immediately before the clinical event. However, in this model features are computed from the data 1 to 2 hours before the clinical event. This 1-2 hours data segment between the observation and clinical event is called forecast gap (or forecast period). The objective here is to predict a clinical event using the observed data prior to 1-2 hours of that event. Thus, doctors can get enough time to take any diagnostic action. The detail steps of this method are described below.

6.6.1 Segmentation

All the bio-signals are segmented in 3 time intervals: observation time \((T_o)\), forecast period \((T_f)\) and target or prediction window \((T_p)\) as shown in Figure 6.11. The goal here is to detect the target clinical event using the observed values applicable to some prior time. If the situation cannot be detected early enough before the event, then it is impossible for doctors to take action for it in time. Therefore, we selected the forecast horizon period \((T_f)\) being at least 1 hour as this is sufficient time for a doctor to make a proper decision. We also took the maximum value of forecast window as 2 hours since any observation before that time degrades the model performance.

![Figure 6.11: The process of segmentation to compute the features from observation window and class from target window](image)

6.6.2 Feature Extraction

The observation window \((T_o)\) size is 10 to 30 minutes for computing different features from these bio-signals. The information in observation window serves as the basis of the input pattern of the classifiers. Different observation window were analysed to see if observing time series for
longer observations would improve prediction. We want to keep the observation window size to a minimum level so there is no need to depend on long-term data for better prediction results. The goal here is to extract as much as information from short-term observation windows for accurate predictions. Moreover, at the same time a window size should contain enough samples to calculate the features properly. Therefore, we found window size 10 was suitable for computing statistical time-domain features and 30 for computing wavelet coefficients. Based on 6 vital signs time-series data, 357 features are extracted from the observation window. They are described below.

6.6.2.1 Correlation coefficients

Correlation coefficients indicate linear relationships or some kind of dependence between signals. An example of such correlations was shown in Figure 6.1 and 6.2. We have measured the pair-wise correlations of six signals from 10-minutes observation window using Pearsons correlation coefficient which also used in PEACE-Home model. If $x$ and $y$ are two bio-signals and $n$ is the number of samples, then the correlation coefficients $r$ is defined by (6.2).

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{n \sum x^2 - (\sum x)^2} \sqrt{n \sum y^2 - (\sum y)^2}}$$

(6.2)

6.6.2.2 Wavelet coefficients

Discrete wavelet transforms (DWT) depict signals both in time and frequency (scale) domains. Each bio-signal is decomposed to different ranges of frequency signals that are known as wavelet coefficients. The original signal passes through low-pass and high-pass filters and emerges with approximation and detail coefficients shown in Figure 6.12. The low-pass filter removes the higher frequency components and of a signal and the high-pass filter picks up the remaining parts.

Wavelet transform can be defined by (6.3). Here, $\psi$ represents wavelet function, $S$ and $P$ are positive parameters representing transform parameters. $C$ represents the coefficients which is a function of scale and position parameters.

$$C(S, P) = \int_{-\infty}^{\infty} f(t) \psi(S, P)dt$$

(6.3)

DWT is an efficient tool for non-stationary signal processing. It can be described by (6.4)
The ViSiBiD model

\[ W(i, j) = \sum_i \sum_j X(i)\psi_{ij}(n) \]  

(6.4)

Where \( W(i, j) \) represents the DWT coefficients, \( i \) and \( j \) are shift transform parameters, and \( \psi_{ij}(n) \) is the wavelet basis time function which can be defined by (6.5)

\[ \psi_{ij}(n) = 2^{-i/2}\psi(2^{-i}n - j) \]  

(6.5)

In previous studies, a Haar wavelet [19] or a Daubechies (DB) wavelet [209] was utilized to select features from a bio-signal such as mean blood pressure. The Haar wavelet is not smooth and so unsuitable for non-stationary signals [222]. It has been proved that forecasting accuracy can be improved by using DB wavelets instead of the Haar [223] wavelets. Therefore, in our study we have chosen Daubechies 2 (db2) wavelet to Level 4. \( A_L(t) \) and \( D_L(t) \) (the approximation and detail coefficients at level \( L \)) of each signal are computed using a 30 minutes observation window. The decomposition is also shown in Figure 6.12.

![Figure 6.12: Discrete wavelet transform and decompositions](image)

\[ \text{Single Signal} \rightarrow \text{Low-pass Filter} \rightarrow A \rightarrow A_2 \rightarrow A_L \]  

\[ \text{High-pass Filter} \rightarrow D \rightarrow D_2 \rightarrow D_L \]

6.6.2.3 Statistical features

The statistical features of 6 bio-signals are extracted from the statistics of a 10 minutes observation window. These features include min, max, mean, standard deviation, skewness, kurtosis, percentiles and inter-percentile ranges. Minimum and maximum contain the information about extreme values; mean and median represent the magnitude of each vital sign; standard deviation describes the variability, and skewness is the third moment of amplitude distribution. Moreover, different percentiles (5th, 10th, 50th, 90th and 95th), IPR (inter-percentile range, which is the difference between 2 percentile values such as 95th and 5th) and kurtosis provide a
snapshot of how the relationship between vital parameters varies over time. We also calculated sequential trends (the number of increasing and decreasing values) and the regression slope in each bio-signal. The slope determines sharp changes in a bio-signal. All these features are easy to calculate. In total, 18 statistics of 6 signals (that is, 108 features) are computed.

### 6.6.3 Training and Learning

Personal experience with IBM’s Watson supercomputer machine learning techniques revealed that data mining algorithms are very efficient and accurate in classifying a particular group of inputs into different classes if they are allowed to learn on a sufficiently large size of training data [224]. In our case, the proposed model first learns about various clinical events from vital sign data of many patients. All extracted features from a large number of patients are combined to build a training and validation set for supervised learning. After data compilation, feature selection and class labelling we get $m$ number of examples with $f$ number of features and $n$ number of possible class values. That is, the training set containing already classified samples is represented by,

$$S = \begin{bmatrix}
    s_{11} & s_{12} & \cdots & s_{1f} & s_{1f+1} \\
    s_{21} & s_{22} & \cdots & s_{2f} & s_{2f+1} \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    s_{m1} & s_{m2} & \cdots & s_{nf} & s_{nf+1}
\end{bmatrix}$$

Each row of $S$ is a $f+1$ dimensional feature vector (also known as attributes) with a known class value in the last column. Thus, the size of training matrix is $m \times f + 1$. In our case $f$ is 357 and when $m$ is sufficiently large, the size of training data can be several gigabytes. The objective here is to balance the accuracy of prediction and the time at which the entire data can be trained for classification. To address the issues of class imbalance and to avoid biased learning the overall data is re-sampled to get a more balanced class distribution.

### 6.6.4 Feature selection

In supervised learning, the goal of feature selection is to find the most relevant features that have a high impact on classification performance. This also reduces the time required for training and storage requirements. While there are many methods for feature selection, in this study we used a simple method which we named “forward feature subset selection”. This technique is independent of any classifier. The metrics used for the evaluation are the accuracy
and building times. We divided all features $f$ into 6 subsets: statistical ($s_f$), correlations ($c_f$) and wavelets of 4 levels ($w_{f1}$ to $w_{f4}$).

The basic idea of this process is that, we start with a subset of features and include another subset of features at each step until we get the combination of subsets with the highest accuracy and the shortest building time. We evaluate the results of this feature selection process on the best classifier along with other strong candidate classifiers.

6.6.5 Classification

The ultimate target is to use the proposed method in real-time home-based health monitoring systems to classify the incoming data of multiple vital signs of a patient with unknown clinical conditions. The continuous data of a new patient are sent to the classification engine inside the cloud and continuous classification results (predicted class values) are obtained over time. If any abnormal clinical event (CE) is predicted by the classifier, a notification is sent to the doctor who makes the final decision as to what (if any) preventive action to take. Therefore, this model can be used as real-time personalized monitoring.

6.7 Experimental results and discussions

To evaluate the performance of proposed learning models we conducted several experiments by using the dataset described in Section 6.4.2. The experimental results and related discussions of both model are described separately in this section.

6.7.1 The PEACE-Home model

As described previously, samples are compiled from 4,893 patient records. For PEACE-home 400 samples from each class are used for evaluation, that is, total 2000 samples are used. To avoid uneven class distribution and biased learning same number of samples are used for each clinical events. The obtained results are described as follows.

6.7.1.1 HMM modelling

For classification we are required to build $L=5$ HMMs ($\lambda_{THTH}$, $\lambda_{BTHH}$, $\lambda_{TTTH}$, $\lambda_{TTHH}$, $\lambda_{NNNN}$) using available samples. Here, number of samples for each class, $s=400$. As we found by PCA method that 60 minutes data before the clinical episode have high influence over the occurrence of that episode, so the features of each sample of each $\lambda_i$ are built using
60 minutes data preceding the targeted clinical event. We considered window size \( w = 10 \) to compute statistical features and correlation coefficients among vitals \([72, 196]\). This results total \( k = 36 \) features for each observation. They are normalized and training sample becomes \( 6 \times 36 \)-dimensional matrix. Then 5 HMM are trained in parallel with \( \binom{2}{3} s = 265 \) samples for each.

To select number of hidden states we build HMM models using 2 to 10 number of hidden states for each of 5 clinical events using 200 training samples. The maximum likelihood probability and model building time (in minutes) of each model is shown as heat map table in Figure 6.13. We want to use the model that maximize the probability as well as has low model building time. Model building time goes high with the number of states. We can see that using 4 to 6 states we can get optimum model because those areas contain maximum probabilities.

To keep model building time minimum level we have chosen 5 hidden state model as optimum and use this for model training.

![Figure 6.13: (a) Change of \( \log P(O|\lambda) \) value, (b) model building time (in minutes) with the change of number of hidden states for 5 HMM models. In (a) Lighter colours indicate higher value in \( \log P(O|\lambda) \) and vice versa. In (b) Darker colours indicate lower value in model building time. The goal is to number of states which have high \( \log P(O|\lambda) \) and low model building time in all 5 models.](image)

### 6.7.1.2 Prediction performance

Prediction performance is evaluated using validation set which contain total \( \frac{1}{3} s = 135 \) samples from each class. To classify an unknown test observation sequence \( O \) into one of the 5 clinical classes (as in Table 6.2), \( \log P(O|\lambda) \) is calculated using forward-backward procedure [161] and compared to find which HMM describes the sequence better. That is, an unknown observation is assigned to \( i \)-th class, such that, \( i = \arg\max_{i} [\log P(O|\lambda_{i})], \ i \in \{THTH, BHTH, TTTH, THBH, NNNN\} \). The confusion matrix from the classification result is shown in Table 6.3.

In this result we found total 23 miss-classifications among 675 samples, that is overall recognition accuracy is 96.6%. Therefore, we can see that the trained HMM models can
Table 6.3: The confusion matrix after performing the classification using validation set of 5 classes. Here 96.6% accuracy is obtained

<table>
<thead>
<tr>
<th></th>
<th>THTH</th>
<th>BHTH</th>
<th>TTTH</th>
<th>THBH</th>
<th>NNNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>THTH</td>
<td>129</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>BHTH</td>
<td>2</td>
<td>125</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>TTTH</td>
<td>0</td>
<td>0</td>
<td>128</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>THBH</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>129</td>
<td>3</td>
</tr>
<tr>
<td>NNNN</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>131</td>
</tr>
</tbody>
</table>

identify different clinical events of unknown observations with a very good accuracy.

6.7.1.3 Probability Prediction in real-time monitoring

We also tested the performance of the classifier in real-time abnormality prediction. 6 patients data are randomly picked for the experiment having a combination of normal and abnormal clinical events. The objective here is to convert the observation window to observation sequence in every minute, feed it to HMM classification engine, and keep getting continuous classification result for next 30 minutes prediction window. In continuous data classification, 5 probability values (\( \log P(O|\lambda) \)) for 5 HMMs are obtained and decision is made by taking the maximum likelihood probability value.

Figure 6.14 shows the result obtained for such an experiment using more 1000 minutes data of one patient. Three types of clinical events (THTH,THTB,NNNN) were observed for this patient. The abnormalities occur after 600 minutes. We can see, there is not much long-term deviation in multiple vitals for first 600 minutes and so most of case \( \log P(O|\lambda) \) values of NNNN is much higher than the others. Also 4 THTH episodes and 4 THBH episodes in the colour plot are found because corresponding predicted log likelihood values go high. This can be verified by abnormal changes in all 4 vitals from the top plot of Figure 6.14. For example, we can see near 700 minutes patient’s HR is high, BP goes low, RR goes low and SPO\(_2\) is low. Thus, this is a THTH event which is detected by HMM-based recognition system by computing likelihood probability and shown in probability colour map. Similar behaviours were observed for other patients data. This proves that, our model can predict the occurrence of abnormal clinical events in real-time to avoid potential clinical danger.
Figure 6.14: The upper plot shows the continuous monitoring values of 6 bio-signals of a patient for 1000 minutes. The lower plot depicts continuous \( \log P(O|\lambda) \) values in a heat-map table obtained for 5 HMMs. The warmer colours indicate higher likelihood probability value.

### 6.7.1.4 Evaluation using cloud frameworks

To evaluate the performance of the system in the cloud the HMM training time for case NNNN is measured in 6 types of Amazon EC2 instance as well as on a local machine. The observation sample size was 100 (each sample was 36-dimensional feature vector). The HMM implementation of Apache mahout framework is used for evaluation in the cloud. The obtained result is shown in Table 6.4.

From the above results we can see that the model training time improves with the increment of the resources in a virtual instance. The training time on a local machine with 8GB RAM took 86 minutes. The lowest training time obtained is 30 minutes for a m3.xlarge virtual machine with 4 CPUs and 8GB RAM. That is, training time decreased almost one third on a
Table 6.4: HMM training time (TT) for case NNNN in different Amazon EC2 instances

<table>
<thead>
<tr>
<th>Type</th>
<th>No of vCPUs</th>
<th>Memory (GB)</th>
<th>TT (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t2.micro</td>
<td>1</td>
<td>1</td>
<td>129</td>
</tr>
<tr>
<td>t2.small</td>
<td>1</td>
<td>2</td>
<td>102</td>
</tr>
<tr>
<td>t2.medium</td>
<td>2</td>
<td>4</td>
<td>67</td>
</tr>
<tr>
<td>m3.large</td>
<td>2</td>
<td>7.5</td>
<td>41</td>
</tr>
<tr>
<td>m3.xlarge</td>
<td>4</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>Local machine</td>
<td>1</td>
<td>8</td>
<td>86</td>
</tr>
</tbody>
</table>

high performance virtual machine. Trivially, the more powerful virtual machine instance will reduce more training time. Therefore, this proves that the cloud implementation for model training improve the model performance.

Moreover, to increase parallelism 5 HMMs are trained in 5 different m3.large virtual instances at the same time and we got the model parameters of all HMMs in short time. In sequential approach on a local machine or on a hospital server it requires to train one model at a time which takes very large time to estimate all the model parameters.

6.7.1.5 Comparison with other models

Table 6.5: Comparison of our model with the results and features obtained in similar works

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of bio-signals</td>
<td>6</td>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Vital signs</td>
<td>HR, BP, RR, SPO₂</td>
<td>HR, BP, RR, SPO₂</td>
<td>BP</td>
<td>HR, BP</td>
</tr>
<tr>
<td>Clinical event</td>
<td>any</td>
<td>any</td>
<td>only Acute hypotension</td>
<td>Hemodynamic instability</td>
</tr>
<tr>
<td>Number of normal samples</td>
<td>700</td>
<td>1370</td>
<td>30</td>
<td>571</td>
</tr>
<tr>
<td>Number of abnormal samples</td>
<td>1720</td>
<td>130</td>
<td>30</td>
<td>116</td>
</tr>
<tr>
<td>Accuracy</td>
<td>97.7%</td>
<td>max 95% (GMM) &amp; 96% (SVM)</td>
<td>94%</td>
<td>ROC max 0.86</td>
</tr>
<tr>
<td>Early prediction capability</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Continuous monitoring support</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Cloud implementation</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Although there are differences in the way of experimental setup, the dataset, platform we have used and the application area we have targeted in comparison with other similar models, it is still possible to draw some performance and feature-wise comparisons. Therefore, we compared our model with the 3 existing works. The work in [76] describes a hospital-focused probabilistic model using Gaussian mixture model (GMM) and one-class support vector machine (SVM) to identify patient deterioration using 4 vital signs (HR, BP, RR, SPO\textsubscript{2}) that we have also used in this study. Others have used fewer bio-signals for predictions. This comparison is presented in Table 6.5.

From these comparisons we can conclude that, PEACE-Home model has more impressive features and results. It is built on large samples and capable of predicting many clinical events more accurately.

6.7.2 The ViSiBiD model

The overall goal here is to be able to predict a clinical event using multiple vitals in most general cases efficiently, by discovering knowledge from big data using fewer features. One of the objectives of this experiment is to compare the performance of different data mining algorithms in terms of model building time, accuracy and forecast gap. Another aim is to find the impacts on these classification results with different subsets of features. For quick learning, all classifiers are trained on high performance virtual machines in parallel. Moreover, to check the consistency of learning model with big data, scalability analysis is performed using MapReduce implementation of the Random Forest classification algorithm. The results are discussed in the following sections.

6.7.2.1 Training and learning

The generated datasets are trained using all 357 features for 3 different forecast horizons. 8 data mining algorithms are trained in parallel using WEKA machine learning toolkit [156]. They are: J48 decision tree (J48), Naive Bayes (NB), Bayes Net (BN), Multi Layer Perceptron (MLP), Sequential Minimum Optimization (SMO), Decision Table (DT), Random Forest (RF) and k-Nearest Neighbour (KNN). All the algorithms are trained using 2 settings: 10-fold cross validation and 66%-split.

For further analysis we kept 3 algorithms J48, SMO and RF, for evaluations. Because the other algorithms performed poorly in terms of model building time (MLP, KNN) and accuracy (NB, BN, DT). We eliminated the classifiers which had model building time of over 5 hours and
recognition accuracy below 90%. Since all 3 training datasets are large (140,000 samples in a 510 MB file) and require a long time for training on a general machine, all data are trained on a m3.2xlarge virtual machine (EC2 instance) provided by Amazon Web Service (AWS) [40].

6.7.2.2 Clinical event prediction result

The results for the best 3 algorithms are presented in Table 6.6. The building time (BT) and validation time (TT) are reported by WEKA. The actual process running time was around 2 hours for J48 and RF classifier and 9 hours for SMO on average.

Table 6.6: Classification results for different classifiers using different forecast horizons on cloud environment in terms of F-measure, Accuracy, Mean Square Error (MSE), model building and validation times. For all types of forecast horizons, the best result is obtained for Random Forest classifier which is highlighted here.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>F-measure</th>
<th>Accuracy (%)</th>
<th>MSE</th>
<th>BT(s)</th>
<th>TT(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>J48</td>
<td>0.93</td>
<td>92.46</td>
<td>0.031</td>
<td>515.55</td>
<td>5.78</td>
</tr>
<tr>
<td>SMO</td>
<td>0.91</td>
<td>90.72</td>
<td>0.225</td>
<td>9101.61</td>
<td>11.35</td>
</tr>
<tr>
<td>RF</td>
<td>0.96</td>
<td>95.86</td>
<td>0.030</td>
<td>142.59</td>
<td>5.80</td>
</tr>
<tr>
<td>J48</td>
<td>0.92</td>
<td>91.59</td>
<td>0.034</td>
<td>445.98</td>
<td>5.24</td>
</tr>
<tr>
<td>SMO</td>
<td>0.90</td>
<td>90.08</td>
<td>0.225</td>
<td>9161.55</td>
<td>9.59</td>
</tr>
<tr>
<td>RF</td>
<td>0.95</td>
<td>95.35</td>
<td>0.033</td>
<td>140.12</td>
<td>5.69</td>
</tr>
<tr>
<td>J48</td>
<td>0.91</td>
<td>91.30</td>
<td>0.035</td>
<td>482.87</td>
<td>5.41</td>
</tr>
<tr>
<td>SMO</td>
<td>0.89</td>
<td>89.23</td>
<td>0.225</td>
<td>10071.9</td>
<td>7.89</td>
</tr>
<tr>
<td>RF</td>
<td>0.95</td>
<td>95.18</td>
<td>0.035</td>
<td>143.15</td>
<td>6.26</td>
</tr>
<tr>
<td>J48</td>
<td>0.92</td>
<td>92.02</td>
<td>0.033</td>
<td>471.85</td>
<td>0.47</td>
</tr>
<tr>
<td>SMO</td>
<td>0.91</td>
<td>90.69</td>
<td>0.225</td>
<td>4094.04</td>
<td>3.29</td>
</tr>
<tr>
<td>RF</td>
<td>0.95</td>
<td>94.87</td>
<td>0.034</td>
<td>132.78</td>
<td>5.89</td>
</tr>
<tr>
<td>J48</td>
<td>0.90</td>
<td>90.10</td>
<td>0.039</td>
<td>486.79</td>
<td>0.52</td>
</tr>
<tr>
<td>SMO</td>
<td>0.89</td>
<td>88.96</td>
<td>0.225</td>
<td>4215.20</td>
<td>2.66</td>
</tr>
<tr>
<td>RF</td>
<td>0.94</td>
<td>94.00</td>
<td>0.037</td>
<td>129.75</td>
<td>5.79</td>
</tr>
<tr>
<td>J48</td>
<td>0.90</td>
<td>90.18</td>
<td>0.040</td>
<td>461.69</td>
<td>0.55</td>
</tr>
<tr>
<td>SMO</td>
<td>0.88</td>
<td>88.26</td>
<td>0.225</td>
<td>4852.68</td>
<td>2.79</td>
</tr>
<tr>
<td>RF</td>
<td>0.94</td>
<td>93.98</td>
<td>0.040</td>
<td>134.67</td>
<td>5.49</td>
</tr>
</tbody>
</table>

From these results we can see that RF classifier with 10-fold cross-validation performed
the best in all evaluation metrics. Moreover, training accuracy decreased with the increase of forecast horizon for all classifiers but the training time remain almost near (because number of samples were same for all cases). In the case of RF classifier, the recognition accuracy is very good (95.18%) even with 2 hours forecast horizon. That is, this classifier can classify unknown clinical events 2 hours before they could happen with more than 95% accuracy when data is trained using 10-fold cross-validation.

6.7.2.3 Analysis of accuracy with data size

Since, SMO took a very long time to build compared to the others, we kept J48 and RF for further comparative analysis. Figure 6.15 shows the change in accuracy and model building time for the J48 and RF classifiers with the change of training data size. This evaluation is done using training data of 1-hour forecast horizon and 10-fold cross-validation. Each smaller sample data contain a balanced subset of all classes. We can see that both the accuracy and model building time increase with the increment of numbers in training data. That is, the classification algorithms perform better when they are trained with more samples. Here, training time increased with the growth of training data size.

![Figure 6.15: Training time and prediction accuracy for J48 and RF with the change in training data size.](image)

6.7.2.4 Analysis of model building time

To analyse the feasibility of model building time we conducted training using the same data and 10-fold cross-validation for J48 and RF in 4 type EC2 instances [40]. The results are shown in Figure 6.16. From this result we can see that the model training time (in seconds) improves with the increment of the resources in a virtual instance. But training time does not improve a lot by choosing the more costly c3.8xlarge instance. The cost of the more powerful EC2
instance (i.e. c3.8xlarge) is also very high. Therefore, we chose to use m3.2xlarge instance as it has a low model building time and is moderate in cost.

![Figure 6.16: Model building time (in seconds) in different EC2 instances using J48 and RF classifiers.](image.png)

### 6.7.2.5 Scalability analysis

In the above analysis the size of training file was large (510 MB with 140,000 samples) but not massive. We observed that model building time increased with the size of data and we needed to use a high performance virtual machine (VM) for model training. Traditional implementations of data mining algorithms are unsuitable training massive data (e.g. GB, TB in size) even it is trained with very high performance VM instance. To address this issue we have performed a scalability analysis using MapReduce implementation of RF classifier. Apache Mahout provides a classifier named Decision Forest [225] which is a MapReduce implementation of RF classifier on Hadoop 2.0.

![Figure 6.17: The change in model building time (in seconds) and accuracy (%) with the change in number of datanodes in a Hadoop cluster.](image.png)

The size of training data of 60-minutes forecast horizon is increased to nearly 5 GB by copying each sample 10 times, resulting in 1,400,000 training instances. A scalability analysis is then performed using Amazon Elastic MapReduce (EMR) service provided by AWS [40].
The training is performed by varying number of data nodes from 4 to 20 in a cluster. The results of accuracy and training time are shown in Figure 6.17.

We can see that, with more than one million training samples, the training time improves almost linearly with an increase number of task nodes in a cluster. The accuracy does not vary much with massive data in comparison with traditional implementation with smaller size data. For a 40 node cluster, the model training time is 133 seconds and accuracy is 95.02%. This supports our contention that ViSiBiD model perfectly suits to train big biomedical data and maintain high accuracy in predictability.

6.7.2.6 Model performance using selected features

In previous evaluations we have used all 357 features generated from 6 bio-signals. The model building time is much longer with this larger number of features. Therefore, we are interested in to see the performance of the classifiers using fewer features. We applied the Forward Subset Feature Selection by selecting a subset of features from the total features each time and obtained the results of 10-fold cross-validation for 3 classifiers in terms of model building time and accuracy. The observed results are listed in Table 6.7. We found when only time domain statistical features or wavelet features are applied, all classifiers are less accurate. Classification accuracies are almost similar using all wavelet coefficients and considering only Level 4 wavelet coefficients. Generally, Level 4 wavelets contain 12.5% information of total signal. That is, we can observe the maximum information using only Level 4 wavelets. We also noticed poor accuracy when pair-wise correlations are not included as features. That is, correlations have a high impact on predictions.

This observation is obtained again for 60, 90 and 120 minutes forecast horizon and 10-fold cross-validation. We found that the model building time reduced by almost one quarter when we used only correlation and Level 4 wavelets coefficients, which was compromised by 1-2% less accuracy. Therefore, we understand that by applying all correlations and top wavelet coefficients we can get the optimal result. This also significantly reduces data size and training time. Thus, we can conclude that using only correlations and top wavelet coefficients of signals it is possible to build the prediction model within a short time with very good accuracy for clinical event predictions.
Table 6.7: Accuracy and model building time for the best 3 classifiers using different selected feature sets. The optimum model in terms of accuracy, building time and features is highlighted in the table.

<table>
<thead>
<tr>
<th>No of features</th>
<th>Statistical features only</th>
<th>Statistical features and correlations</th>
<th>Wavelets of all 4 levels</th>
<th>Only level 4 wavelets</th>
<th>Statistics, correlations and level 4 wavelets</th>
<th>Correlations and level 4 wavelets</th>
<th>All features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (in %) and model building time (in sec) using 10-fold cross validation and Forecast horizon 60 minutes (1 hour)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classifier:</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
</tr>
<tr>
<td>J4S</td>
<td>79.58</td>
<td>238.31</td>
<td>83.12</td>
<td>261.38</td>
<td>82.46</td>
<td>321.25</td>
<td>82.29</td>
</tr>
<tr>
<td>SMO</td>
<td>74.12</td>
<td>3111.20</td>
<td>77.69</td>
<td>3458.25</td>
<td>76.00</td>
<td>7801.10</td>
<td>76.66</td>
</tr>
<tr>
<td>RF</td>
<td>80.75</td>
<td>47.81</td>
<td>83.25</td>
<td>55.02</td>
<td>81.44</td>
<td>97.22</td>
<td>80.76</td>
</tr>
<tr>
<td>Accuracy (in %) and model building time (in sec) using 10-fold cross validation and Forecast horizon 60 minutes (1.5 hours)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classifier:</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
</tr>
<tr>
<td>J4S</td>
<td>79.58</td>
<td>238.31</td>
<td>83.12</td>
<td>261.38</td>
<td>82.46</td>
<td>321.25</td>
<td>82.29</td>
</tr>
<tr>
<td>SMO</td>
<td>74.12</td>
<td>3111.20</td>
<td>77.69</td>
<td>3458.25</td>
<td>76.00</td>
<td>7801.10</td>
<td>76.66</td>
</tr>
<tr>
<td>RF</td>
<td>80.75</td>
<td>47.81</td>
<td>83.25</td>
<td>55.02</td>
<td>81.44</td>
<td>97.22</td>
<td>80.76</td>
</tr>
<tr>
<td>Accuracy (in %) and model building time (in sec) using 10-fold cross validation and Forecast horizon 120 minutes (2 hours)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classifier:</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
<td>BT(s)</td>
<td>Acc(%)</td>
</tr>
<tr>
<td>J4S</td>
<td>79.58</td>
<td>238.31</td>
<td>83.12</td>
<td>261.38</td>
<td>82.46</td>
<td>321.25</td>
<td>82.29</td>
</tr>
<tr>
<td>SMO</td>
<td>74.12</td>
<td>3111.20</td>
<td>77.69</td>
<td>3458.25</td>
<td>76.00</td>
<td>7801.10</td>
<td>76.66</td>
</tr>
<tr>
<td>RF</td>
<td>80.75</td>
<td>47.81</td>
<td>83.25</td>
<td>55.02</td>
<td>81.44</td>
<td>97.22</td>
<td>80.76</td>
</tr>
</tbody>
</table>

6.7.2.7 Prediction in continuous monitoring

One exciting aspect about the above results is that, they are obtained from serially compiled examples that compromise records entirely. This kind of data preparation simulates real-time monitoring. Continuous monitoring from streaming data is an integral part of clinical decision-making and big data analytics. In our experiment of continuous monitoring, the output class labels are predicted for every minute by increasing the observation window 1 minute at a time and maintaining the fixed gap before target window. To evaluate the performance of real-time monitoring we tested the Random Forest classifier model generated using the features from correlations and level 4 wavelets coefficients, 120 minutes forecast horizon and, over 1000 minutes continuous data of 10 patients. Therefore, for each patient we had total of 850 test samples (by deducting first 30 minutes observation and 120 minutes forecast window and starting from the 151-st minute). The results obtained from this evaluation are presented in Table 6.8. Here by false positive (FP) we considered if a class is incorrectly identified as some other class and by true positive (TP) we mean the number of samples correctly classified among 850 samples.

From this result we can conclude that the developed learning model is perfectly suitable for real-time personalised monitoring in home settings as it shows good accuracy in clinical event predictions using a large forecast gap (2 hours).
Table 6.8: Weighted Average of FP and TP for 5 different classes using level 4 wavelet coefficients and correlation coefficients as features, Random Forest classifier, 2 hours forecast gap and 1000 minutes continuous data of 6 bio-signals for 10 different patients.

<table>
<thead>
<tr>
<th>Patient</th>
<th>Condition</th>
<th>FP rate</th>
<th>TP rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mostly normal</td>
<td>1.29%</td>
<td>95.95%</td>
</tr>
<tr>
<td>2</td>
<td>mostly normal</td>
<td>0.94%</td>
<td>97.18%</td>
</tr>
<tr>
<td>3</td>
<td>mostly normal</td>
<td>1.52%</td>
<td>95.04%</td>
</tr>
<tr>
<td>4</td>
<td>hypotensive and tachycardia</td>
<td>1.76%</td>
<td>94.62%</td>
</tr>
<tr>
<td>5</td>
<td>hypotensive and bradycardia</td>
<td>2.11%</td>
<td>94.00%</td>
</tr>
<tr>
<td>6</td>
<td>hypertensive and tachycardia</td>
<td>1.88%</td>
<td>94.25%</td>
</tr>
<tr>
<td>7</td>
<td>hypertensive and bradypena</td>
<td>1.76%</td>
<td>94.48%</td>
</tr>
<tr>
<td>8</td>
<td>mostly all abnormal</td>
<td>2.47%</td>
<td>93.20%</td>
</tr>
<tr>
<td>9</td>
<td>mostly all abnormal</td>
<td>2.82%</td>
<td>92.16%</td>
</tr>
<tr>
<td>10</td>
<td>mostly hypoxia</td>
<td>2.58%</td>
<td>92.85%</td>
</tr>
</tbody>
</table>

6.7.2.8 Performance comparison

No similar technique is known to exist to compare with our model. Table 6.9 depicts the ability of the proposed learning model using hybrid feature space in comparison with similar methods used in the literature. Most works in the literature stated accuracy as their final results. Therefore, this comparison is also done based on accuracy.

Table 6.9: Comparison of ViSiBiD model with similar methods in literature

<table>
<thead>
<tr>
<th>Method</th>
<th>Signal(s) used</th>
<th>Obs. (mins)</th>
<th>Forecast (mins)</th>
<th>Total patients</th>
<th>No of classes</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViSiBiD model (Using Statistics, correlations, and wavelet coefficients as features and RF as classifier)</td>
<td>HR, SBP, DBP, MAP, RR, SPO₂</td>
<td>10-30</td>
<td>120</td>
<td>4,893</td>
<td>5</td>
<td>95.18</td>
</tr>
<tr>
<td>ViSiBiD model (Using wavelets and correlations as features and RF as classifier)</td>
<td>HR, SBP, DBP, MAP, RR, SPO₂</td>
<td>10-30</td>
<td>120</td>
<td>4,893</td>
<td>5</td>
<td>94.03</td>
</tr>
<tr>
<td>Principal Component Analysis (PCA) as features and MLP classifier [211]</td>
<td>HR, SBP, DBP, MBP</td>
<td>30</td>
<td>60</td>
<td>1,311</td>
<td>2</td>
<td>86.03</td>
</tr>
<tr>
<td>Wavelet transform and SVM classifier [209]</td>
<td>MAP</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>2</td>
<td>92.96</td>
</tr>
<tr>
<td>Wavelet transform and FF Neural Network [19]</td>
<td>MAP</td>
<td>120</td>
<td>60</td>
<td>60</td>
<td>2</td>
<td>94</td>
</tr>
<tr>
<td>Genetic algorithm and SVM [78]</td>
<td>MAP</td>
<td>90</td>
<td>60</td>
<td>60</td>
<td>2</td>
<td>93.65</td>
</tr>
</tbody>
</table>

We can see that existing methods only work on limited number of signals and small size
data. They have not addressed the issues related to scalability in their analysis. In contrast, our method uses many signals, works on massive data, is scalable and has high prediction accuracy even with a large forecast gap.

6.8 Conclusion

In this chapter, we have presented two prediction models “PEACE-Home” and “ViSiBiD”, for early detection of anomalous clinical episodes caused by multiple vital signs. The PEACE-Home model uses HMM for learning and classifying various clinical events from the behaviours of multiple vital signs. This model is also suitable for continuous monitoring of a patient’s health using the information learned from the changes in many physiological parameters. The other model, ViSiBiD, uses hybrid features generated from a large number of patient records. Different data mining algorithms are used for learning and classifying various clinical events caused by changes in multiple vital signs. The advantages of cloud computing are utilized for scalable and fast learning and real-time monitoring. High prediction accuracies and low model building times are maintained by boosting the cluster size with the growth of data, employing a distributed learning algorithm and selecting a subset of significant features. Both models are suitable for finding clinical abnormalities of an unknown patient using the information learned from the changes in multiple physiological parameters of many patients. The proposed work is a valuable contribution to the biomedical informatics and healthcare big data analysis since this systematic prediction model can reduce healthcare cost and chronic disease related deaths. This also simplifies the jobs of healthcare professionals by sending them early warnings and by reducing false predictions.

6.9 Epilogue

The contributions of this chapter answer the first part of fourth research question. Here we considered that patient’s different clinical events are already labelled, as validated by the PCA method. That is, the pre-requisite of these learning models is the availability of labelled data. Both models are capable of handling any number clinical events. But if such labelled events are not available we need a mechanism to separate patient groups based on clinical class. This problem will be solved in the next chapter.
Chapter 7

Patient Clustering by Dynamically Partitioning Patient States

The process of clustering similar patients by recognizing potential patterns in correlated vital sign data can lead to better treatment and more effective monitoring, and potentially can prevent crises such as death or disability. In relation to the second part of fourth research question, this chapter aims to find a proper solution of this patient clustering process.

In a general scenario, doctors use their years of experience and training to interpret the vital sign data of patients and integrate them into groups based on similarities between new patients and representative previous cases. This process takes a long time and is very costly because experienced human resources for doing this task are limited and highly paid. Moreover, such expert rule-based systems with manual classifications increase the possibility of various human errors. Therefore, a long standing goal in the area of health informatics is to automate the process of patients’ similarity analyses.

Some existing techniques utilize prior knowledge about the trends in biomedical data and use that knowledge to build models for patients having similar clinical behaviours with supervised learning [226, 227, 228, 229]. We described such models in the previous chapter. These models can then be used to decide about the clinical situation of an unknown patient. Such learning techniques hold great promise but require the availability of large numbers of clinical examples for model estimation. However, the existence of such large training samples is mostly uncommon. It is not possible to gather knowledge about known clinical conditions without a feasible number of samples. Moreover, in supervised models the rate of misclassification can be high when new types of uncertainties are introduced by closely coupled vital sign data if the
whole model is not trained again. These limitations necessitate the innovation of unsupervised algorithms for this purpose.

The time-series of vital signs are multi-dimensional, high in resolution (e.g. collected per minute), non-stationary and interrelated. They can change rapidly or slowly and may be induced by underlying control systems of the human body. The interactions among multiple vitals are also specific to patients’ clinical natures and other attributes such as age, sex, activities and emotional states. If we just take a snapshot (e.g. 24 hours) of continuous vital sign data of a single patient, we will see many different types of changes within this period which is impossible to interpret merely by observation. A patient can be in different states based on the correlations and changes of these vital signs. Thus, an important aspect of a patient clustering method is to summarize these high-dimensional data to extract useful information that can be utilized for similarity analysis among patients.

Clustering algorithms normally group similar patients based on statistical features of multiple bio-signals. However, the task of finding various statistics in multi-dimensional data requires the prior solving of optimization problems such as identifying the length of a sliding window to extract the best statistical nature of the data and selecting relevant features for clustering. Moreover, the use of such demographic information increases data dimensionality which makes clustering algorithms computationally intensive. In this chapter, we propose a patient-specific dynamic pattern discovery method that converts multi-dimensional biomedical data into single-dimensional states. Our algorithm does not need to depend on statistical behaviour, rather, we utilize entropy values to identify and separate different states in a patient. Entropy normally reflects the amount of disorder in data. Similar correlations and changes in vital signs will output a similar entropy value. We measured the aggregated entropy value of each multi-dimensional instance and utilized only that value to partition different clinical situations within a patient in terms of states. In this way, we get a summarized clinical nature of individual patients. Finally, our proposed patient clustering algorithm combines similar patients based on this summarized information.

The proposed algorithms are completely unsupervised and do not depend on any prior knowledge. These can work on small to large numbers of patients and any number of clinical conditions. Once a set of similar patients is identified by the proposed algorithms, doctors can make some generalized diagnostic decisions for that patient group. This is an integral part of remote patient monitoring.

The rest of this chapter is organized as follows. Section 7.1 describes the motivations and contributions of this chapter. Section 7.2 discusses the related work in the area of patient
7.1 Motivations and Contributions

7.1.1 Motivations

Automations of grouping similar patients without any prior knowledge is a powerful approach to the meaningful utilization of multi-dimensional physiological data. A proper clustering method can identify the common distribution of clinical conditions within a cluster of patients that can be used to solve a wide range of problems in modern healthcare technology. The motivations that drove us to develop this patient clustering model are summarized below.

- The dynamic discovery of patterns based on clinical situation allows the adaptation of rapidly changing medical data. If a new clinical event appears with a particular variation in multiple vital sign data, then the model should determine the clinical class by its patterns and create a new cluster of patients who have that specific pattern.

- For a particular patient cluster, a doctor can track the results for patients who are on different treatments, and can thereby identify which treatment is the most effective or has fewer side effects.

- The successful outcomes from diagnosis and medication for patients in a particular subgroup can be used as a reference in treating a new patient belonging to that subgroup.

- By monitoring the antecedent rates of each patient cluster independently, it is possible to identify potential disease outbreaks.

- Similar patient analysis can simplify the process of personalized decision support.

7.1.2 Contributions

The major contributions of this chapter are listed below.

- We developed an unsupervised and patient-specific pattern discovery method for highly correlated vital sign data. This method partitions a patient’s physiological situation into
a number of discrete states based on observed changes of multiple bio-signals with different distributions over a defined period. The method uses Euclidean distance to detect similarities between two observations of patient data. The multi-dimensional physiological data is converted to a single dimensional matrix by computing an aggregated instance-wise entropy (AIE) value which is further utilized as the partitioning criteria. The number of partitions for each patient is determined dynamically by separating the data as measured by the AIE vector into different cohesive regions. These partitions summarise large multidimensional data onto a number of clusters or states. Later, this summarised information is used to identify similarities between two or more patients. The noisy data-points are eliminated using an average relative density (ARD) factor defined over two separate localised neighbourhood levels.

- We implemented a method of identifying the subgroups of similar patients from a large group of patients. After finding different dynamic states of a patient, statistical features are extracted from each state. Afterwards, distance correlation measure is utilized to find the similarities between state features of two patients. The identified similarity values are then used to perform a hierarchical clustering which partitions the group of similar patients into distinct and well-separated clusters. The same type of correlations in multiple vital signs having similar changes in a time-series mostly have similar consequences, prompting the occurrence of the same clinical event in two patients. This approach of patient clustering can combine patients with similar medical conditions in the same category with no prior knowledge. This can help doctors to make a common treatment plan for such specific patient groups. Moreover, the developed model can be applied to identify the cluster to which a new patient may belong without any previous knowledge about his/her medical condition.

7.2 Related work

There is extensive research works in patient clustering methods for hospital settings that use features from different vital signs to characterize various illnesses and utilize different distance measures for finding similarities. A predictive system is developed in the context of monitoring patients in ICU to categorize patients into two groups based on a particular clinical condition [226]. The method computes top-F wavelet coefficients from physiological signals such as heart rate, SPO\textsubscript{2}, and mean, systolic and diastolic blood pressure. Then patient similarity
Related work

Analysis is performed by using generalized Mahalanobis distance between the selected features of two patients and a regression model. However, this method is goal-oriented as it uses prior knowledge of targeted clinical events to build their models. Such a method may not work for other types of clinical conditions. A similarity-based searching and pattern-matching algorithm is proposed using temporal dynamics of multi-parameter vital sign data [230]. This method uses manually-designed features to capture temporal trends with a Gaussian mixture model for clustering clinical episodes. The results show great success in “search by example” and classification tasks, but the major problem with this algorithm is its dependency on manually-defined features.

Unsupervised discovery of meaningful patterns and features from biomedical data is another popular research direction in the area of patient clustering. A probabilistic clustering method is proposed using a mixture model with an empirical prior distribution vital sign time-series that encourages a degree of smoothness over time for each variable’s measurements [231]. This is dependent on the assumption that data are “missing at random”. K-means clustering is used to separate patient records containing multiple vital signs into five different classes based on the pattern similar to bag-of-word representations [232], where the physiological time-series data is converted to discrete symbols similar to the way documents are modelled in information retrieval tasks. While this approach can capture certain higher level structures of multi-dimensional medical data, it discards important information related to temporal correlations. A topic model, similar to those used in natural language processing, is proposed to find the temporal nature in multi-parameter physiological data [212]. However, this method relies on the availability of complete, high frequency data, which in most cases is not available. In general, the available medical data are uncertain and sparsely sampled.

Besides multi-dimensional vital sign data, various clinical data and patients’ medical histories such as drugs, lab tests, demographics, and ICD codes are used in patient clustering methods using different clustering algorithms [227, 233, 234]. A multivariate Gaussian process regression is proposed to distinguish vital sign trajectories between normal and abnormal patients recovering from gastrointestinal surgery [235]. To investigate the patterns of co-occurrence of medical co-morbidities in autism spectrum disorders, hierarchical clustering was applied to manually-designed features [236]. A fuzzy clustering method is described to divide obese patients into groups of diverse metabolic types [237]. K-means clustering is used to find the likelihood of some diseases based on anticipated likelihood attributes with core attributes of disease in a data point [238]. All these methods mostly rely on manually defined features.

In our method we have used entropy to identify dynamic states of a patient from vital sign
correlations. Entropy is a powerful tool for analysing and clustering multi-dimensional data [239]. It allows a description of the distributions of the possible state of a system, and therefore the information encoded in it. This property makes it a popular measure for unsupervised classification [240, 241]. It is also widely used in biomedical data analysis [242, 243, 244]. A classical entropy measure requires some prior knowledge i.e. the probability distribution associated with time-series data about the system [245]. Different clustering algorithms also assume some prior knowledge and parameter settings to determine the number of partitions in clustering [246], whereas our proposed method avoids any arbitrary definitions of parameters or probabilistic assumptions in dynamic partitioning.

7.3 The patient clustering method

The main objective of the proposed method is to find clusters (or subgroups) of similar patients using correlated multiple vital signs data to simplify the task of healthcare professionals for making clinical decisions. For a target patient the method uses an entropy criteria to dynamically identify different situational “states” of that patient based on interactions and changes of multiple vital signs without any prior knowledge about his/her clinical conditions. Our process presumes that a long-term multi-dimensional raw physiological data of many patients are available for analysis inside the cloud storage. Our algorithm runs in parallel on high powerful virtual machines in the cloud. For each patient a snapshot of overall data is used and it is summarized to a single-dimensional matrix of states using our proposed algorithm. Since the conversion is independently done for each patient and it totally depends on the correlations among physiological data of that particular snapshot, we call this process “dynamic pattern discovery”. Afterwards, similarity analysis is performed using the statistical features of states in each patient to identify multiple subgroups of patients. This workflow is illustrated in Figure 7.1.

The target is to identify $k$ subgroups/clusters ($C_1, C_2, ..., C_k$) from $m$ patients ($P_1, P_2, ..., P_i, ..., P_m$) based on the similarities of changes that occur in multiple vital signs. We consider that each of $m$ patient record contains the observations of $p$ vital signs with sampling interval $t$ (e.g. $t = 1$ minute). For a single patient, $Z$ is the data snapshot (e.g. 24 hours) containing $n$ instances of the overall observation which is utilized to identify the dynamic states of that patient based on the correlations of $p$ vitals. Therefore, $Z$ is a data matrix of dimension $n \times p$. Let, $Z_l$ is the observed data for patient $P_l$ where $1 \leq l \leq m$. First the noisy instances (i.e. outliers) are eliminated and then each multi-dimensional matrix, $Z_l$ is converted to a
The patient clustering method

Large number of patients with unknown clinical nature

Highly-correlated multiple vital sign data of each patient

Dynamic partitioning of patient’s states using entropy criteria

Individualized and unsupervised pattern discovery

Feature extraction and similarity analysis

Hierarchical clustering using similarity pattern

Patient subgroups identification

Patient Subgroup-1

Patient Subgroup-2

Patient Subgroup-3

Figure 7.1: The workflow of the proposed patient clustering method. First, different states of multiple patients with unknown medical conditions are identified dynamically in parallel from their multiple vital signs using similarity analysis based on entropy values. Then, a hierarchical clustering is used on features of different patient’s states to identify subgroups of similar patients.

single-dimensional vector $E_l$ by computing collective entropy value of each instance $z_{li}$ in $Z_l$. Afterwards, dynamic states $\{\Pi_l\}$ for each patient $P_l$ are computed using $E_l$. In the next step, features are extracted from each state of each $P_l$ which are then used to find $k$ subgroups. All steps toward the patient clustering method are described in the following.

7.3.1 Data normalization

A wide range of variations exist in vital sign values of different patients. Each vital sign has a specific physiological range. Moreover, the upper and lower bounds also vary in different vital signs. Therefore, data need to be transformed to a standard normalized value. Here, the raw values in each $Z$ are normalized to between 0 and 1 which result the matrix $Y = \|Z\|$. The values in $Y$ are computed first by normalizing to a standard score value as given in Equation (7.1) and then all values are transformed into range $[0, 1]$ using feature scaling as given in Equation (7.2).

$$Z' = \frac{|Z - \bar{Z}|}{\sqrt{\frac{1}{n} \sum_{j=1}^{n} (Z - \bar{Z})}}$$  \hspace{1cm} (7.1)$$

$$Y = \|Z\| = \frac{Z' - Z'_{\text{min}}}{Z'_{\text{max}} - Z'_{\text{min}}}$$  \hspace{1cm} (7.2)
7.3.2 Outlier detection

The abnormal values can occur in physiological data due to faulty measurements. The outliers do not follow the usual changes in data pattern. Therefore, we need to exclude those data instances to extract the correct summarized information from the data. To detect these outliers, average relative densities (ARD) are calculated for all data points in $Y$. The effectiveness of using ARD for detecting anomalies in a wireless sensor network domain was discussed in [247].

The following definition is used for calculating the ARD.

**Definition 7.1. Average Relative Density (ARD).** Let, $Y_i$ be a single $p$-dimensional instance of normalized matrix $Y$, that is, $Y_i=(y_{i1}, y_{i2},..., y_{ip})$. The ARD of $Y_i$ is defined based on two radius $\Lambda$ and $\lambda$ as given in Equation (7.3).

$$D_{ar}(Y_i, \Lambda, \lambda) = \frac{d(Y_i, \lambda)}{\sum_{Y_j \in N(Y_i, \Lambda)} d(Y_j, \lambda)} \neq \frac{d(Y_i, \Lambda)}{\sum_{Y_j \in N(Y_i, \Lambda)} d(Y_j, \lambda)}$$

(7.3)

$N(Y_i, \rho)$ is the set of instances $\{Y_j\}$ located within radius $\rho$ centred in $Y_i$. That is, $N(Y_i, \rho) = \{Y_j \in Y | \text{eucl}(Y_j, Y_i) \leq \rho\}$ where $\text{eucl}(Y_j, Y_i)$ is the Euclidean distance between instance $Y_i$ and $Y_j$. The density of instance $Y_i$ is, $d(Y_i, \rho) = |N(Y_i, \rho)|$ which is the number of instances in $N(Y_i, \rho)$. $\lambda$ is called counting neighbourhood over which density is estimated for each instance. The radius $\Lambda$ is called sampling neighbourhood which is used to collect sample instances for computing average.

According to the work in [247] we picked $\Lambda=0.75$ and $\lambda=0.1$ where all values in $Y$ are normalized within range $[0, 1]$. The density computed using $\lambda$ value detects the small variations within data while $\Lambda$ value recognizes variations in a much broader range. Thus, the ARD measure provides an approximation where an instance can be clearly separated from its nearby instances and can be uniquely utilized to detect and eliminate outliers in the normalized data. After computing ARD for all instances $Y_i$ in $Y$ we get a single-dimensional ARD vector $A_Y=[A_{Y_1}, A_{Y_2}, ..., A_{Y_1}, ..., A_{Y_n}]$ of length $n$ where $i$-th value $A_{Y_i} = D_{ar}(Y_i, \Lambda, \lambda)$. The next task is to find the outliers using the values in $A_Y$. A threshold is defined to identify those anomalous instances and thus we have the following definition.

**Definition 7.2. Outlier Factor** The outliers are the set of instances $\{Y_o\}$ in $Y$ who’s average relative density values are higher than a threshold value $T_o$ which we call as outlier factor. This is defined by the set, $O(Y, T_o) = \{Y_o \in Y | A_{Y_o} > T_o\}$. Here,
In Equation (7.4), the first part is the mean ($\mu_{AY}$) and the second part is the standard deviation ($\sigma_{AY}$) of vector $AY$. That is, if ARD value of any instance is higher than mean plus one standard deviation value computed from ARD values of all instances then that instance is considered as outlier. One standard deviation is taken as sufficient to reflect outlying data due to the high cohesiveness in the normalized data that is analysed.

Thus, if $o$ number of outliers are detected by this method then these $o$ instances are removed from $Y$ and we get a matrix $X$ of dimension $q = (n - o) \times p$. The next step is to convert the values in $X$ to a vector of dimension $q \times 1$ using an entropy criteria.

### 7.3.3 Unsupervised pattern discovery

Once the sparsely distributed noisy instances are removed the next task is to separate the data in $X$ into $s$ number of partitions. The value of $s$ is unknown and can be different for different patients because it is based on interrelations and changes of multiple vital signs data in $X$. Therefore, the goal is to find the unknown parameter $s$ and the distributions of the partitions in $X$. We call it “unsupervised pattern discovery”. The steps are described as follows.

#### 7.3.3.1 Similarity measure

Given a dataset $X$ to be separated into $s$ partitions, the most common way to find partitions is to determine the similarities among the elements in $X$. Usually, such similarities are established in terms of proximity through a metric or distance function. Many distance measures are described in literature to solve various clustering solutions. In this work, we used the Euclidean distance as it is an effective measure of proximity and easy to compute. The Euclidean distance between two instances $x_i$ and $x_j$ of $X$ is given by Equation 7.5.

$$d_{ij} = \sqrt{\sum_{u=1}^{q} (x_{iu} - x_{ju})^2}$$  \hspace{1cm} (7.5)

We then define the similarity between two instances $x_i$ and $x_j$ by,

$$S_{ij} = \exp^{-\alpha_i D_{ij}}$$  \hspace{1cm} (7.6)
The patient clustering method

Here, $\alpha_i = -\ln(0.5\bar{D}_i)$. $\alpha$ is calculated adaptively for every $i$-th observation. It depends on the current distribution of data and taken as the corresponding value when the similarity measure is substituted by 0.5 similarity, and $\bar{D}_i$ which is the mean distance between all instances of $i$-th observation. After calculating all $S_{ij}$ where $1 \leq i, j \leq q$ we get a square matrix $S$ of dimension $q \times q$ where all $S_{ii} = 1$ and $S_{ij}$ may not be equal to $S_{ji}$.

7.3.3.2 Transform multi-dimensional observation into single-dimensional vector

The next step is to transfer the information in an $p$-dimensional instance $x_i$ to a single value using an entropy measure to simplify the process of partitioning. In vital sign data each element may contain unpredictability because of the existence of uncertainty and disorder in overall data. The entropy value can measure the uncertainty or unpredictability of a random variable as well as a data element. According to the definition of Shannon’s Entropy [248], it is an evaluation of the information content of a random variable $R$ with possible values \{r$_1$, r$_2$, ..., r$_q$\}. From a statistical viewpoint, the information of the event ($R = r_i$) is inversely proportional to its likelihood. This information is denoted by $I(r_i)$, which can be expressed as, $I(r_i) = \log(\frac{1}{p(r_i)}) = -\log(p(r_i))$.

In our method, to determine the entropy $e_{ij}$ between two instances $x_i$ and $x_j$ using similarity measure $S_{ij}$ we used Shannon’s logarithmic information measure [248, 249] which is given by,

$$e_{ij} = -[S_{ij} \log S_{ij} + (1 - S_{ij} \log(1 - S_{ij}))]$$ (7.7)

After calculating all $e_{ij}$ from $S_{ij}$ where $1 \leq i, j \leq q$, we get another square matrix $E_S$. Here, first we calculate the entropy values for each observation $i$ relative to all other observations within a concentrated data distribution. Afterwards, we calculate aggregated instance-wise entropy (AIE) for each observation $i$ which is given by,

$$e_i = -\sum_{j \neq i, j=1}^{q} [S_{ij} \log S_{ij} + (1 - S_{ij} \log(1 - S_{ij}))]$$ (7.8)

Here, the disorder in each instance is calculated through aggregating its entropy value with all other instances within the observation. After calculating all $e_i$ (where $1 \leq i \leq q$) we get a single-dimensional vector $E$ with $q$ elements. This $E$ provides a single value criteria for the dynamic partitioning step. The AIE measure is adaptive to dynamic data distribution changes and captures data characteristics that are not limited to second order statistics. Any $e_i$ value can be zero when the outcome is certain. That is, those instances have no impact on dynamic
The patient clustering method

behaviour of a patient’s states. Therefore, we ignored the instances having \( e_i = 0 \) during partitioning.

### 7.3.3.3 Dynamic partitioning

In this step, the vector \( E \) is partitioned into a number of clusters which we call “states”. Here, the number of partitions (or states) based on similarities is determined algorithmically without any prior knowledge about the clinical nature of the patient.

**Definition 7.3. State.** If \( X = \{X_1, X_2, ..., X_i, ..., X_q\} \) is the observed data segment of a patient containing \( q \) observations and \( p \) vitals with each \( X_i = (x_{i1}, x_{i2}, ..., x_{ip}) \), then a set of states (or partitions), \( \Pi = \{\Pi_1, \Pi_2, ..., \Pi_i, ..., \Pi_s\} \) are defined over spatial distribution of \( X \) based on the similarity values observed in \( q \)-length AIE vector \( E \). That is, \( q \) instances of \( X \) are partitioned into \( s \) states.

Let, \( E_i \) be the AIE vector of all elements in partition \( \Pi_i \). There is no common element between two partition \( \Pi_i \) and \( \Pi_j \). That is, \( E_i \cap E_j = \emptyset, \forall i, j \). Let, \( \sigma_{E_i} \) is the standard deviation of \( E_i \). Then each state \( \Pi_i \) has a maximum similarity difference \( \sigma_{E_i} \) on AIE values between the elements of \( E_i \). That is, \( E_i(\max_e) - E_i(\min_e) \leq (1)\sigma_{E_i} \) where \( 1 \leq i \leq s \). \( E_i(\max_e) \) and \( E_i(\min_e) \) is the maximum and minimum AIE value observed in \( E_i \).

Based on the above definition the process of dynamic partitioning is described in Algorithm 7.1. This procedure takes AIE vector \( E \) and similarity matrix \( S \) as input and gives a set of partitions \( \Pi \) as output. The process utilizes aggregated similarity vector \( SS \) and its standard deviation \( SD \) for partitioning. After eliminating instances with zero-entropy values, the process finds the instance with the least entropy value. This instance is used as the centroid of \( i \)-th partition around which the region of that partition will be defined. A threshold value \( th \) is used on total standard deviation of the similarity value matrix to define the partition boundary. After this, the instances within that partition boundary are identified and removed from \( E \). Then, once again the next partition is defined based on the next observed least entropy value in remaining \( E \). This process is repeated until no instances are left and the data distribution is completely partitioned on to disjoint states with similar AIE values. The number of states varies in patients based on underlying vital sign correlations.

This partitioning process converts multi-dimensional matrix \( X \) to a sequence of state indices, \( f(x) = i \), when \( x \in \Pi_i \). This process of clustering does not require any predefined parameter (i.e. estimated number of clusters) like k-means or fuzzy c-means clustering [246]. Here, the number of clusters is determined algorithmically to fit data variability in a dynamic
Algorithm 7.1. Dynamic partitioning of patient’s state

1: **Input:** $E$
2: **Input:** $S$
3: **Output:** $\Pi = \{\Pi_1, \Pi_2, ..., \Pi_s\}$
4: $th \leftarrow 0.3$
5: **Procedure** DynamicPartitioning()
6: **begin**
7: $SS \leftarrow \sum(S)$
8: $\sigma \leftarrow \text{std}(S)$
9: $\Pi \leftarrow \emptyset$
10: $i \leftarrow 1$
11: $ZeroI \leftarrow \text{find}(E = 0)$
12: $E \leftarrow E - E(ZeroI)$
13: **while** $E \neq \emptyset$ **do**
14: $minE \leftarrow \min(E)$
15: $I \leftarrow \text{find}(E = minE)$
16: $CS \leftarrow SS(I)$
17: $\Pi_i \leftarrow \text{find}\left(\left((CS - th \cdot SD) \leq SS \leq (CS + th \cdot SD)\right)\right)$
18: $E \leftarrow E - E(\Pi_i)$
19: $SS \leftarrow SS - SS(\Pi_i)$
20: $\Pi \leftarrow \Pi \cup \{\Pi_i\}$
21: $i \leftarrow i + 1$
22: **return** $\Pi$
23: **end while**
24: **end**

manner. The number of partitions can be controlled by changing the threshold value $th$. A high value of $th$ means a larger partition boundary that can have more instances which reduces the number of partitions and vice versa. The use of this algorithm have several advantages in our problem domain. Firstly, it will generate a higher number of states for an unstable patient who’s vital sign values vary a lot and that’s why we will observe large variations in entropy values. On the other hand, for a stable patient we will observe a fewer number of states. Thus, this process can easily distinguish between stable and unstable patients. Secondly, this will generate nearly identical state distributions for patients having similar natures and thus they can be grouped easily. Finally, the proposed algorithm is scalable. This method can run in parallel for hundreds or thousands of patients in the cloud and can find individual dynamic patterns for each patient.
7.3.3.4 Cluster analysis

As our original data are multi-dimensional having more than three dimensions, to visualize the distribution of clusters identified by dynamic partitioning in three or two-dimensional space we have used Principal Component Analysis (PCA). PCA is a statistical procedure that uses orthogonal transformation to convert a set of correlated variables into a set of values of linearly uncorrelated variables called principal components. It is generally used for dimensionality reduction [18]. After applying PCA on dataset $X$ it generates a score matrix $score$ of same dimension. The way the score matrix is generated guarantees that more than 95% data is represented by first few columns. Therefore, we considered first two and three columns of the score matrix (also called principal components) to explore the clusters in 2D and 3D space respectively. This analysis is discussed in Section 7.4.

7.3.4 State feature extraction

After completing the above steps for a single patient we get a partition vector $\{\Pi_1, \Pi_2, \ldots, \Pi_i, \ldots, \Pi_s\}$ and corresponding AIE vector $\{E_1, E_2, \ldots, E_i, \ldots, E_s\}$. For each partition $\Pi_i$ we extract some statistical features of AIE values from $E_i$. They are $min$, $max$, $mean$ and $standard deviation$ relating to the entropy distribution on each single partition. $min$ is the centroid of the partition, $max$ contains the information of boundary point, $mean$ and $standard deviation$ contain the information about the distribution of the overall partition. Therefore, each state is converted to a four-element feature vector. For each patient, we get a feature matrix $F$ with dimension $4 \times s$ where each column contains the feature vector of a state. As described earlier, according to our method here $s$-value is different for different patients. Therefore we need a mechanism to find similarities among these uneven matrices.

7.3.5 Identify similar subgroups

The final step of our method is to find similar patients based on extracted features of states and to categorize them onto the same cluster.

**Definition 7.4. Subgroup** Let, $\{P_1, P_2, \ldots, P_i, \ldots, P_m\}$ be a list of $m$ patients and $\{F_1, F_2, \ldots, F_i, \ldots, F_m\}$ are corresponding state feature matrices. A feature matrix $F_i$ of patient $P_i$ has $s_i$ number of states. A patient subgroup or cluster $C_h=(P_{h_1}, P_{h_2}, \ldots, P_{h_r})$ with $r$-number of members is defined based on the similarity score observed from pairwise distance correlation measure $dCorr(F_i, F_j), i \neq j$ over all $F_i$ ($1 \leq l, i, j \leq m$).
Distance correlation is a measure of statistical dependence between two random vectors of arbitrary, not necessarily equal dimension [250]. It is measured from their distance covariance \( dCov \) and distance standard deviations (square root of the distance variance \( dVar \)) as given by,

\[
dCorr(F_i, F_j) = \frac{dCov(F_i, F_j)}{\sqrt{dVar(F_i)dVar(F_j)}}
\]  

(7.9)

After computing distance correlations for each pair in \( F_i \) we get a square correlation matrix \( CR \) of dimension \( m \times m \) where \( CR_{ij} = CR_{ji}, CR_{ii} = 1 \). This correlation matrix is then used as input of the hierarchical clustering algorithm for finding subgroups of patients. The whole process is described in Algorithm 7.2. As in our analysis the number of states varies in patients so it generates feature vector with different dimensions. In this context, distance correlation is an appropriate measure because it can identify correlations between two uneven vectors.

**Algorithm 7.2.** Identify patient subgroups using distance correlation and hierarchical clustering

1: **Input:** \( \{F_1, F_2, ..., F_i, ..., F_m\} \)
2: **Output:** \( C = \{C_1, C_2, ..., C_k\} \)
3: **Procedure** PatientCLustering()
4: begin
5: \( i \leftarrow 1 \)
6: \( j \leftarrow 1 \)
7: \( CR \leftarrow \emptyset \)
8: for each pair \( (F_i, F_j), i, j \) to \( m \) do
9: \( \text{if } i = j \text{ then} \)
10: \( CR(i, j) \leftarrow 1 \)
11: \( \text{else} \)
12: \( CR(i, j) \leftarrow dCorr(F_i, F_j) \)
13: \( CR(j, i) \leftarrow dCorr(F_i, F_j) \)
14: \( \text{end if} \)
15: \( \text{end for} \)
16: \( DS \leftarrow 1 - CR \)
17: \( D \leftarrow \text{euclideanDist}(DS) \)
18: Run hierarchical clustering using dissimilarity matrix \( D \)
19: \( T \leftarrow \text{hierarchical cluster tree} \)
20: \( C \leftarrow \text{cluster}(T) \)
21: return \( C \)
22: end

We have used hierarchical clustering to identify patient subgroups. The hierarchical clus-
tering method groups data over a variety of scales by creating a cluster-tree or dendrogram. The tree has a multilevel hierarchy where clusters of one level are combined as clusters at the next level. This allows for fine-grained determination on the level of clustering in a manner that is more appropriate for the application under consideration. In order to decide which observations should be combined, a measure of dissimilarity between the sets of observations is required.

The $CR$ matrix, observed by distance correlations, provides the information about similarities between all pairs where $0 \leq cr_{ij} \leq 1$. We will get dissimilarity information by simply deducting all values in $CR$ from 1 which produces the dissimilarity matrix $DS$. Then the Euclidean distance measure is used to assign the correlation values to be distances ($D$). Once these proximities between members in the data are computed, a hierarchical clustering algorithm, which expects a dissimilarity matrix, is used to determine how $m$ patients should be grouped into clusters. After applying a cut-off criteria on the hierarchical cluster tree $t$, it divides the overall patient population into $k$ number of clusters ($C_1, C_2, ..., C_k$). The criteria is chosen experimentally which is briefly described in the experimental evaluation.

### 7.4 Experimental results

The main goal of the experimental evaluation is to validate the ability of finding patient-specific states and identifying similar patients by applying the proposed patient clustering method over highly correlated bio-signals. The performance assessment of any unsupervised learning process is difficult without using any ground truth or expert opinions. Unsupervised learning methods are mainly used in exploratory analysis where the ground truth is unknown. In our case, no ground truth was available about the dynamic states of patients. For this reason, the performance of the proposed dynamic state partitioning process is validated by Silhouette value [251] and Principal Component Analysis.

For patients’ subgroups identification method we used labelled data to measure the performance. Moreover, the performance of the hierarchical clustering is validated by using an inconsistency index and comparing with k-means clustering. Overall, the model is evaluated using data from many patients having different clinical conditions and different patterns in vital sign values. The nature of the data and the results are discussed in the following sections.
7.4.1 Data Sets

As described in previous chapters, here we have also used multiple vital sign data of several patients from MIMIC [193] and MIMIC-II [152] numeric dataset of MIT physiobank [181] archive to validate the proposed patient clustering approach. These datasets were preferred because it fulfilled the criteria to validate our approach. We have used 2 types of datasets. Dataset-1 is from MIMIC and is labelled with clinical class for different patients [252]. This dataset contains 121 records. Here we have also used only those records which have clean continuous monitoring data of six vital signs (HR, SBP, DBP, MBP, RR, and SPO$_2$) for at least 24 hours. Each of six vital signs are sampled per minute. We only consider first 24 hours data of each patient as a snapshot. The longer data segments increase the time of computation. Moreover, in real-life situations we will not have such long-term data for many patients. For this reason we considered only the snapshot of first 24 hours as it would be enough to differentiate the patient situation using our analysis. Finally, a total of 35 patient records satisfied our criteria. In summary, for this dataset we have $p=6$ vital signs, $m=35$ patients and for each patient the number of instances in the data is $n=1440$ (24 hours = 1440 minutes). We observed a total of 12 different clinical classes (including 2 unknowns) in these 35 patients. The clinical class distribution is shown in Figure 7.2.

Dataset-2 is gathered from the MIMIC-II database. Different clinical situations can occur due to individual changes in the six vital signs. This was listed in Table 6.1 of Chapter 6. Based on this, we consider the same 5 types of clinical classes described in Table 6.2 of Chapter 6. In MIMIC-II database we searched for these events and extracted in total 34 samples having a different number of observations in overall data. The sample size is based on the availability of that situation in patient record with a data segment of significant size (e.g. 600). The index values of this labelled data is shown in Table 7.1.

Figure 7.2: The distribution of 12 clinical classes in 35 patients of Dataset-1 used in our evaluation. This is used as ground truth for the patient clustering experiment.

Dataset-2 is gathered from the MIMIC-II database. Different clinical situations can occur due to individual changes in the six vital signs. This was listed in Table 6.1 of Chapter 6. Based on this, we consider the same 5 types of clinical classes described in Table 6.2 of Chapter 6. In MIMIC-II database we searched for these events and extracted in total 34 samples having a different number of observations in overall data. The sample size is based on the availability of that situation in patient record with a data segment of significant size (e.g. 600). The index values of this labelled data is shown in Table 7.1.

Now the objective of our experiment is, to find the dynamic states of each sample using Algorithm 7.1 based on the given snapshot of data (e.g. 24 hours for Dataset-1) with six correlated vital signs. Then using these dynamic states and applying Algorithm 7.2 get the
Experimental results

Table 7.1: The occurrence of clinical event targeted for the evaluation and their sample size and index number

<table>
<thead>
<tr>
<th>Clinical event</th>
<th>Samples</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNNN</td>
<td>20</td>
<td>1-20</td>
</tr>
<tr>
<td>THTH</td>
<td>7</td>
<td>21-27</td>
</tr>
<tr>
<td>BHTH</td>
<td>4</td>
<td>28-31</td>
</tr>
<tr>
<td>TTTTH</td>
<td>2</td>
<td>32-33</td>
</tr>
<tr>
<td>THBH</td>
<td>1</td>
<td>34</td>
</tr>
</tbody>
</table>

subgroups of these samples and compare the distribution of clinical classes situated upon the described baseline of these 2 datasets. The experimental results are discussed below.

7.4.2 Outlier detections

In the first step, the target is to detect the instances which are anomalous or unusual in overall data distributions. As described earlier, this kind of anomaly can occur due to faulty measurements and can cause misleading information if they are considered for patients’ state identification and needs to be segregated first. We detected such outliers using average relative density (ARD) measure as shown in Figure 7.3 for a single patient.

Some outliers are easy to identify by visual inspection. As in top plot of Figure 7.3, at time 8 and 122 abnormal changes in BP values are observed. In the middle plot of Figure 7.3 we found that according to the density measure 11 outliers are detected in this data segment. For example, in 4 instances near time 180 abnormal drop in RR values is observed. We call an instance as outlier when it has higher ARD value than others which we can verify from the bottom plot of Figure 7.3. We detect such outliers in the overall data segment for each sample. Those instances are not used in the entropy-driven dynamic state estimation step.

7.4.3 Dynamic pattern discovery

After eliminating the outliers the rest of the instances are used to identify the dynamic states of a patient. We applied Algorithm 7.1 to all samples of described datasets. Thus, six correlated vital signs are converted to a number of states for each patient sample. As described earlier, the number of partitions vary in different patients, even two patients having a similar clinical conditions can have different pattern of partitions. As we have in total six signals, so to visualize the partitions in 3-dimensions we took first 3 principal component values after applying PCA. Generally, first 3 principal components contain most of information of all signals [18]. An
Figure 7.3: Outlier detection using average relative density measure. The top plot shows the time-series of six vital signs of a patient for 185 minutes. In the middle plot, the detected outliers are shown in vertical lines. The bottom plot shows the changes in the density measure over time computed from six vital sign values of the top plot.

Example of such dynamic pattern discovery for three patients (4, 5 and 6) in Dataset-1 having CHF/Pulmonary Edema is shown in Figure 7.4. For first 2 patients the number of identified states is 7 and it is 9 for the third patient. The result of dynamic states for patient-4 using PCA is visualized more precisely in Figure 7.5(a) and (b) in 3D and 2D. This also shows the centroids of each partition. The number of dynamic states identified for 35 patient samples in Dataset-1 and 34 patient samples in Dataset-2 is summarized in Figure 7.6.

7.4.4 Cluster Analysis

Different patients exhibit different dynamic patterns in terms of states discovered by the entropy measure. Figure 7.7 shows the distribution of clusters for 9 patients (1,2,6,9,10,12,15,25,27)
Experimental results

The distribution of identified dynamic states using first three principal components (PC) as X, Y and Z axis value

Figure 7.4: Dynamic pattern discovery from six correlated vital signs. The plots in the left side show the time-series of six correlated bio-signals for 3 patients having CHF/Pulmonary Edema in first 24 hours. The plots in the right side show the patterns of states identified using an entropy-driven clustering method on 3 principal component axes. We can observe that the number of discovered states are different in 3 patients.

in Dataset-1 using 2 principal components. These results suggest that, the distribution of states (based on the correlations of six vital signs) for these patients are different from one another. It is impossible to visually identify any kind of similarity using these patterns. Therefore, we need to find the correlation between a pair of such patterns to identify any existence of similarity and use that to group similar patients.

The transformation to states from multi-dimensional vital sign data is a piece-wise function that represents a patient’s clinical situation as a sequence of state transitions. As in Figure 7.8 we can observe the transitions of 7 different states for patient-4 of Dataset-1. Therefore, we can have a simplified summary of patient’s clinical situation using the proposed state partition process.

There is no ground truth available to evaluate the accuracy of the state partitioning process. Therefore, the quality of dynamic state partitioning is validated by an internal performance measure named Silhouette [251]. Silhouette values validate consistency within clusters of data. The silhouette value for each instance is a measure of how similar that instance is to instances
Experimental results

Figure 7.5: Visualization of state distributions for patient 4 in (a) 3D and (b) 2D using PCA. Here PC-1, PC-2 and PC-3 are first 3 principal components computed from score matrix of original data.

in its own cluster, when compared to instances in other clusters. The silhouette value for the $i$-th point, $S_i$ is $S_i = (b_i - a_i) / \max(a_i, b_i)$, where $a_i$ is the average distance from the $i$-th point to the other points in the same cluster as $i$, and $b_i$ is the minimum average distance from the $i$-th point to points in a different cluster, minimized over clusters. Normally, the silhouette value ranges from $-1$ to $+1$. A high silhouette value (i.e. high $b_i$ and low $a_i$ value) indicates that it is well-matched to its own cluster, and poorly-matched to neighbouring clusters. If
Figure 7.6: Number of discovered dynamic states of all patient samples for two experimental datasets.

Figure 7.7: Cluster analysis for 9 patients of Dataset-1. From the labelled information we know that 3 of these patients (6,7 and 9) belongs to the same clinical class and others are in different classes.

most points have a high silhouette value, then the clustering solution is appropriate.

Figure 7.9 shows the Silhouette plots obtained from the discovered dynamic states of 12 different patients of Dataset-1. Here we can see that in most of the cases high positive silhouette
Experimental results

Figure 7.8: Transition of states in 24 hours data for a single patient.

values are identified and they are very close within its own cluster. This confirms that our
dynamic partitioning algorithm can generate well-separated patient states using an entropy
measure.

Figure 7.9: Consistency validation of dynamic partitioning method using Silhouette plots of
12 patients in Dataset-1.

7.4.5 Patient classification

After identifying the states of all patients in two experimental datasets and extracting the
statistical features of aggregated instance-wise entropy values of those states we applied Algo-
rithm 7.2. Figure 7.10 shows the pair-wise correlations coefficients in heatmap for 2 datasets.
This correlation matrix is used for a hierarchical clustering. We can visualize and extract
some information from these heatmaps. For example, in Dataset-2 we can observe that first 20 patients are highly correlated (high colour value in heatmap), which is also true for the benchmark data. Therefore, such correlation matrix is ideal to solve our problem using a hierarchical clustering.

![Patient correlation matrix for Dataset-1 containing 35 patients](image1)

![Patient correlation matrix for Dataset-2 containing 34 patients](image2)

Figure 7.10: Correlation map among all patients for 2 experimental datasets.

The hierarchical binary cluster tree, created after applying the Euclidean dissimilarity and shortest distance linkage between each pair, is viewed graphically using dendrogram plot in Figure 7.11 for Dataset-1. In this figure, members along X-axis represents patient indices. The links between patients are represented as upside-down U-shaped lines. The height of the U indicates the distance between the patients. For example, the link containing patient 6 and 24 has height 0.02. The tree consistency is verified using cophenetic correlation coefficient which measures the correlation between tree input and output. This was 91.66% for our case.

The hierarchical cluster tree naturally divides the data into distinct, well-separated clusters. This can be particularly evident in a dendrogram diagram created from data where groups of objects are densely packed in certain areas and not in others. We have used inconsistency coefficient threshold of 0.9 which generates 12 cluster as shown in different colours below
Experimental results

Figure 7.11: Hierarchical cluster tree for Dataset-1 containing 35 patients along X-axis. The original distribution of patients are shown at top and identified subgroups are shown at bottom in different colours.

dendrogram plot in Figure 7.11. This value is chosen from experimentally so that it can generate the desired number of clusters. An inconsistency coefficient determines where to partition data set into clusters. We observe that, this clustering method mostly combines similar patients in the same cluster according to the baseline as shown in top bar of Figure 7.11. Here we identified a total of 7 misclassifications out of 35. They are 13, 14, 22, 3, 12, 34 and 24. As in Figure 7.2 class of 11 and 24 was unknown. Here, 24 is categorized as Sepsis with 16. 11 is grouped along with one sample of respiratory failure (22). Considering a classification of 24 that is correct we have a total of 6 wrong classifications. Thus, the accuracy of clustering here is 82.85%. For some patients data could be very unstable and sometime they show a nearly similar pattern of another clinical class. This can be the reason for some misclassifications.

Dataset-2 contains 20 stable and 14 unstable clinical samples. The unstable samples have 4 different types where multiple vitals goes below or above the normal range. This dataset is not specific to particular patient rather specific to particular clinical situation where multiple variation can occur for the same patient. The objective of using this dataset is to show the advantage of our technique in separating different clinical situations based on the correlations of multiple vitals. Here we found a total of 3 wrong classifications. Sample 24 is classified as normal. Also 30 and 31 are classified as THTH which should be BHTH. Therefore, for this dataset the accuracy is 91.17%.

We also applied k-means clustering to Dataset-1 by using the same correlation coefficient matrix generated previously from AIE values as input and $k$ value as 12. The result is illustrated in Figure 7.13. We can see that, for some clinical conditions we get a similar result in k-means like hierarchical clustering. But in case of k-means we get a higher number of
Figure 7.12: Hierarchical cluster tree and similarity matrix for Dataset-2 containing 34 patients along X-axis. The identified subgroups are shown at bottom in different colours.

Figure 7.13: Clustering result for Dataset-1 using k-means clustering

7.5 Conclusion

In this chapter, a patient clustering method is demonstrated by discovering patient-specific dynamic patterns using an entropy measure on correlated and multi-dimensional vital sign data. A two-step unsupervised approach is proposed for this clustering purpose. In the first step, the multi-dimensional physiological data of each patient are converted to a single dimension by computing collective entropy of each observation instance. Then this data is partitioned into different cohesive regions to represent patient-specific situations in terms of the number of dynamic states. The outlier and misleading data points are eliminated by computing average relative density over two distance neighbourhood levels. The parameters for dynamic partitioning are derived adaptively for each patient. This process enables individualized knowledge discovery for a large number of patients independently and in parallel. Principal Component
Conclusion

Analysis is adopted to visualize patients’ dynamic states and the quality of this clustering method is verified using silhouette plots. In the second step, the features of dynamic states are directly utilized to compare the similarities between two patients. Distance correlation measures are used to identify the dissimilarities between two uneven vectors of states. This information is further used by a hierarchical clustering algorithm to find the actual patient clusters. Experiments were conducted on two types of publicly available vital sign data with labelled clinical conditions. Experimental results indicate high recognition accuracies for identifying patient subgroups with different clinical conditions. This process can provide personalized decision intelligence to clinicians and can help them in the diagnostic decision-making process.
Chapter 8

Conclusion

The aim of the research was to build a scalable context management system with predictive capability for personalized and assisted healthcare. From the outset we identified that when biomedical and contextual data are continuously collected and accumulated from many patients, the quality of context-aware services can be improved by solving certain issues. Firstly, correlating the information from different context sources and aggregating them to a single information for making the actual decision. Secondly, discovering patient-specific knowledge from the correlations among different contexts in past data and utilizing that knowledge to predict future abnormalities only for that particular patient. Thirdly, detecting long-term behavioural change using context histories and their inter-relationships. Fourthly, making the learning models adaptive and continuous so they can adapt based on the changes in contexts and can still predict future events continuously with good accuracy. Fifthly, predicting the future event of an unknown patient based on the learned knowledge for that event from available data of many similar patients. And finally, identifying similar patients in an unsupervised manner when no prior information about patient similarity is available.

The research presented in this thesis has therefore focused on the development of a cloud-based solution for simultaneous support of a large number of patients in an assisted living environment. It explored different machine-learning solutions for large scale biomedical data analysis. Various cloud resources are utilized for heavy computational tasks such as context processing, storage, management, training and knowledge discovery. Most of the experiments are carried out in the cloud environment. The developed techniques reduce the rate of false predictions and assist health professionals in remote monitoring centres through early discovery of patient-specific abnormal clinical events by using various contexts of a patient.
In addressing four research questions, this thesis proposes a set of solutions to achieve reliable and personalized context-aware decision support for patients in ambient assisted living (AAL) systems and to simplify the job of healthcare professionals. The major contributions of this research were described in Section 1.6 which emphasise the limitations in existing literature mentioned in Section 1.5. A cloud-based model was described in Chapter 2. Chapters 3 and 4 focused on the solutions for abnormality and long-term behavioural change detection in a patient by discovering his/her personalized knowledge and correlating the information in different patient context domains. Chapter 5 addressed the issue regarding continuous prediction of patient-specific future clinical abnormalities. Chapter 6 described solutions for abnormality detection and prediction for an unknown patient using knowledge based on data of similar patients, while Chapter 7 discussed the process of clustering those similar patients.

The first research question was addressed in Chapter 2. The main objective of this chapter was the development of a scalable platform where heterogeneous patients equipped with versatile wearable sensors in different assisted living systems can be easily monitored by using a common framework through context-aware services. It was identified that traditional solutions rely on small hand-held devices for biomedical data processing, targeted to support only some specific services and which are thus incapable of handling a large number of patients through personalized services. Therefore as a solution, a cloud-orientated context-aware middleware “CoCaMAAL” was proposed to simplify the complex context management task of a large number of AAL users. The context processing tasks were distributed in different cloud components and corresponding functionalities were described. A unified context modelling approach was proposed which is easily modifiable, reusable, adaptable and extendable. The model was prototyped and tested for different patient scenarios for detecting abnormalities in an ongoing context of a patient and matching appropriate context-aware services with 96.34% accuracy. The performance of the model was evaluated using queueing theory and a short response time was obtained while supporting a large number of patients in parallel. Overall, this model enabled a qualitative remote patient monitoring support by taking the leverage of cloud computing technology.

Regarding second research question, a personalized knowledge discovery framework was illustrated in Chapter 3 and a change detection model was developed in Chapter 4. The CoCaMAAL model was extended to a new model, “BDCaM”, and was described in Chapter 3. The BDCaM model had the functionalities for knowledge discovery process to find patient-specific anomalies using large historical data. Algorithms were developed based on MapReduce and Apriori to identify personalized association rules for a patient, such as the thresholds
values of different vital signs with the change of user activities and disease symptoms. The discovered rules were validated using different data mining algorithms over a large amount of contextual data. The framework was evaluated by implementing a case study that discovered the knowledge of abnormal conditions of patients having variations in blood pressure and heart rate values. The obtained results proved the effectiveness of the proposed method for detecting proper anomalous situations of different patients using personalized rules. In Chapter 4, a set of pattern recognitions models were proposed for detecting future changes in the patient’s activities, daily routines and vital signs. The change detection models that were developed and evaluated were: a Hidden Markov Model (HMM) based approach detecting abnormalities in daily activities, a process of identifying irregularity in routine behaviours from statistical histories and an exponential smoothing technique to predict future changes in various vital signs. These changes were correlated using a fuzzy rule-based system and 95.10% accuracy was observed for abnormal change detection. The experiments were conducted using simulated data. The results indicated the usefulness of the proposed techniques to accurately interpret the future changes in a patient before any potentially dangerous health situation arises, in order to send proper alerts to the healthcare service providers.

A learning model for clinical decision support system (CDSS) using multi-label classification (MLC) was proposed in Chapter 5, concerning the solution of the third research question. The system was developed to have the capability of predicting patient-specific multiple vital sign values using their correlations. In the experimental settings, different statistical features and correlation coefficients of six vital signs were formulated as MLC problem. Eight MLC algorithms along with three fundamental machine learning algorithms were tested on the cloud environment using data from 85 patients. Different MLC evaluation measures: Hamming score, F1-micro average and accuracy were used for interpreting the prediction performance. More than 90% Hamming scores were obtained for all 85 patients by using different multi-label classifiers. The model was also made adaptive and situation-aware by using an incremental learning process. The results were compared with single-label classifiers and without considering the correlations among the vital signs. The evaluation results reveal that multi-label classification techniques using the correlations among multiple vital signs were effective ways for early estimation of future values of those vital signs.

The fourth research question was addressed in Chapters 6 and 7. Two predictive models for future abnormality detection of an unknown patient using prior knowledge of many similar patients were developed in Chapter 6. The first model, “PEACE-Home”, is capable of making probabilistic predictions of different clinical events using recent past data of multiple vital
signs from many patients. This model is also suitable for making a probabilistic estimation of clinical events for a continuously monitored patient. Principal Component Analysis (PCA) was utilized for separating the identified clinical events using the temporal behaviours of six bio-signals and the Hidden Markov Model (HMM) was then adopted for predicting those clinical events. The HMM models are trained by using samples from 1023 patient records and 96.6% recognition accuracy was obtained for classification. The second model, “ViSiBiD”, is a static predictor model that utilized different features obtained from 1-2 hours prior data of 4,893 patient records by computing discrete wavelet transform, short-term statistics and correlation coefficients of six bio-signals. Different data mining algorithms were used for learning and classifying various clinical events caused by changes in multiple vital signs, and best accuracy (95.85%) was achieved for Random Forest classifier using all features. The effectiveness of both models was also demonstrated in the cloud platforms through comparative evaluations.

An innovative algorithm for identifying similar patients was proposed in Chapter 7. Two clustering algorithms were developed. In the first one, different physiological states in a patient were dynamically computed using aggregated instance-wise entropy value to transform multi-dimensional physiological data to single dimension. Then dynamic partitioning was applied based on minimum entropy value to discover the dynamics states of each patient. In the second algorithm, similarity analysis was performed among dynamic states of patients using distance correlation and then hierarchical clustering was used to separate patient subgroups based on similarity. The accuracy of clustering for the two benchmarked data was 82.85% and 91.17%. This can help doctors to identify patients with similar problems and to make a common treatment plan for them.

Overall, the significance of this research work is to provide a comprehensive care to the patients in ambient assisted living and to simplify the workload of healthcare professionals. The developed prediction models can significantly reduce chronic disease-related deaths by early predictions of emergencies. These techniques also can improve lifestyle pattern of home-monitored patients by prior detection of future behavioural changes. Using our proposed model, heterogeneous patients can be monitored using a common platform and thus it can reduce admissions to hospital. The adoption of our proposed model will also play a key role in reducing the healthcare-related budget. Home monitoring systems are easy to build with low cost sensors and devices. The utilization of cloud servers for computational and storage facilities will reduce the cost even more. Therefore, the development of a cloud-based context-aware system for ambient assisted living will have a significant impact on the world economy.
Limitations of the Study and Recommendation for Future Works

In this thesis, problem-specific efficient learning algorithms have been developed that can be applied for context-aware monitoring in an assisted living environment by utilizing cloud platforms. This research mostly utilized existing data mining techniques in an innovative way to solve problems regarding abnormality detection, prediction and knowledge discovery. The proposed solutions also addressed the scalability issues by building a cloud-based model which can handle big data effectively and accurately. While the proposed techniques outperform related schemes, there is still room for improvements of these approaches. Here we will briefly discuss the limitations of our study and suggest some future work based on these limitations.

We did not set up any real-life testbed for data collection: rather, we considered an abstraction of real-life AAL systems which still produced meaningful results. We mostly relied on simulated prototypes and focused on learning model development, validation and their performance evaluation in the cloud environment. We used publicly available data of patients monitored in hospital beds and presumed a similar nature for home monitoring data when collected in a controlled manner from wearable sensors in an AAL system. In future, researchers can collaborate with hospitals and doctors to set up a testbed environment for collecting long-term real-time data of patients having various chronic conditions and evaluate the performance in real-life scenarios.

In our research we ignored other autonomous functional requirements of real-time home-based monitoring system such as low level infrastructure of sensors, sensor failures, reliability of communication between sensors and mobile device, noises in sensor data, context quality assessment, network traffic performance for data transmission to the cloud, cloud orchestrations, and security and privacy in patient data. Such requirements require an independent research investigation. We can say that our proposed model is a basic foundation that expands the scope of multiple research directions.

In Chapter 2, we emphasized building a cloud-based context-aware system and integrating it with body sensor network in ambient assisted living. The model can infer the context of a single user but cannot associate contexts of multiple users: this would be an interesting extension to identify different social contexts involving multiple users. The model described a service-oriented architecture but skipped services regarding data security. Moreover, the model is evaluated with a limited number of case studies where context and services are generated synthetically. Therefore, it is significantly important to ensure privacy in context information before processing and further justify the model with a wide range of case studies from real-life
In Chapter 3, the personalized knowledge discovery framework showed encouraging results for the implemented case study using synthesized data. However, the model is not verified for more complicated case studies using more context domains such as the environmental situation, disease history and emotional state of the patient. The model performed well for large data but we could not test the model with actual big data (e.g. petabytes) due to data unavailability and lack of resources. Moreover, we have used Apriori as an association rule mining algorithm for our approach because it suited perfectly for MapReduce operations. However, other state-of-the-art methods for association rule mining was not verified as their MapReduce conversion processes are still ongoing.

In Chapter 4, an integrated system was proposed to detect deviations in behaviour using multi-context information and to predict the outcome by combining all the contextual knowledge. We observed promising results by combining information from multiple contexts such as activity, daily routine and vital signs. However, much other contextual information such as sensor status, ambient condition and medical history were not used in information fusion. Separate pattern recognition model can be developed to gather intelligent information from those context domains. The model is validated only using fuzzy rule-based model but was not tested for other rule-based systems. Moreover, the Hidden Markov Model (HMM) is used for detecting change in activity and location patterns. Although it had a good accuracy in computation, HMM has a disadvantage in high computational cost. This model is also evaluated only using simulated data and thus there is scope to improve this technique by validating it with real-time data.

In Chapter 5, we proposed a new approach to utilize multi-label classification algorithms to forecast person-specific vital sign values and related abnormalities using the correlations among multiple vital signs. However, there is no feedback mechanism in the process that lets the method know whether an abnormal clinical event has actually occurred. This method will not perform well if a patient exhibits a slow deterioration. The process may consider this gradual worsening as a new normal situation and so fail to detect an abnormality. Moreover, only six vital signs were used for model evaluation and correlations of other important vital signs such as body temperature and ECG were not considered. The proposed model also used all 123 features computed from six vital signs. The size of window for computing features is chosen experimentally. Thus, as a recommendation for future consideration, we suggest implementing a process of selecting an optimal window size and evaluating different feature selection algorithms to identify set of features that have actual impact on model performance.
Moreover, the model is required to be validated with more vital signs and more multi-label classifiers. Also, a feedback control system can be added for continuous validation of the model’s performance.

In Chapter 6 two different models were described for predicting future clinical events using observed data of many similar patients. By using a large amount of data for training and for evaluation we ensured that our models were exposed to many different types of patients during training and testing. These models also used data from six vital signs and a limited number of clinical events. They need to be validated with more vital signs, clinical events and features. In this analysis, clinical events are defined using generalized normal values. However, what is normal in one patient may not be the same for another. The described methods also consider the clinical episode persists for a fixed period. However, rationally the period should vary for different clinical events. Here we also need an appropriate feature selection process to select only the relevant features which can minimize model training time. The training time of the HMM model can be improved by employing a fast learning algorithm such as Extreme Learning Machine. Both of our developed models are required to train with large historical data from many patients through offline analysis. However, a learning model which is already induced is difficult to update with new samples. The entire training process is required to be repeated with new data to adapt the system with new knowledge. Therefore, as a future solution we recommend to improve this process using data stream mining algorithms such as AdaptiveHoffding tree or Very Fast Decision Tree (VFDT).

In Chapter 7, we described a patient clustering method by identifying different dynamic states in a patient using entropy criteria. However, if both small change and large change occur within short period entropy could not determine the small changes in data. In the biomedical domain, such small changes are sometimes highly significant. Moreover, this model assumes the availability of a large collection of data from many patients which may be difficult to obtain. The dynamic model is generated only using a small snapshot of long-term data. However, for some patients vital sign data may not vary within that limit and so our method will map every instance to the same cluster. A possible solution can be to increase the size of observation but this also increases computational costs of calculating entropy in every instance. The method uses euclidean distance as the similarity measure but is not tested using other distance measures. The obtained results from our initial evaluation should be further validated by using a larger patient population and many different types of clinical cases.
Bibliography


[96] J. Bardram. The java context awareness framework (JCAF)—a service infrastructure and
programming framework for context-aware applications. *Pervasive Computing*, pages
98–115, 2005. **Cited on page 30.**

and execution of context-aware component applications. In *ICPS06: IEEE International

2002. **Cited on page 30.**

[99] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in
page 29.**

[100] G. Van Den Broek, F. Cavallo, and C. Wehrmann. *AALIANCE ambient assisted living

[101] R. Woolrych and A. Sixsmith. Challenges of user-centred research in the development
of ambient assisted living systems. *Impact Analysis of Solutions for Chronic Disease

[102] Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B.K. AlShebli, M. Caccamo, C.A. Gunter,
In *Systems, Man and Cybernetics, 2006. SMC’06. IEEE International Conference on,

projects.com/euprojects/amigo/, 2008. **Cited on page 29.**


framework. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth


[123] Liang Wang, Tao Gu, Hanhua Chen, Xianping Tao, and Jian Lu. Real-time activity recognition in wireless body sensor networks: from simple gestures to complex activities.


BIBLIOGRAPHY


[204] Xiaoliang Wang, Qiong Gui, Bingwei Liu, Zhanpeng Jin, and Yu Chen. Enabling smart personalized healthcare: a hybrid mobile-cloud approach for ecg telemonitoring. *IEEE*


