

# Application of Hierarchical Temporal Memory to Anomaly Detection of Vital Signs for Ambient Assisted Living

**Benhur Bakhtiari Bastaki**

This thesis is submitted in partial of the requirement of Staffordshire University for the  
degree of *Doctor of Philosophy*

Staffordshire University

UK

October 2019

*Copyright © 2019, Benhur Bakhtiari Bastaki.*

## **Abstract**

This thesis presents the development of a framework for anomaly detection of vital signs for an Ambient Assisted Living (AAL) health monitoring scenario. It is driven by spatiotemporal reasoning of vital signs that Cortical Learning Algorithms (CLA) based on Hierarchical Temporal Memory (HTM) theory undertakes in an AAL health monitoring scenario to detect anomalous data points preceding cardiac arrest.

This thesis begins with a literature review on the existing Ambient intelligence (AmI) paradigm, AAL technologies and anomaly detection algorithms used in a health monitoring scenario. The research revealed the significance of the temporal and spatial reasoning in the vital signs monitoring as the spatiotemporal patterns of vital signs provide a basis to detect irregularities in the health status of elderly people.

The HTM theory is yet to be adequately deployed in an AAL health monitoring scenario. Hence HTM theory, network and core operations of the CLA are explored. Despite the fact that standard implementation of the HTM theory comprises of a single-level hierarchy, multiple vital signs, specifically the correlation between them is not sufficiently considered. This insufficiency is of particular significance considering that vital signs are correlated in time and space, which are used in the health monitoring applications for diagnosis and prognosis tasks.

This research proposes a novel framework consisting of multi-level HTM networks. The lower level consists of four models allocated to the four vital signs, Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP), Heart Rate (HR) and peripheral capillary oxygen saturation (SpO<sub>2</sub>) in order to learn the spatiotemporal patterns of each vital sign. Additionally, a higher level is

introduced to learn spatiotemporal patterns of the anomalous data point detected from the four vital signs. The proposed hierarchical organisation improves the model's performance by using the semantically improved representation of the sensed data because patterns learned at each level of the hierarchy are reused when combined in novel ways at higher levels.

To investigate and evaluate the performance of the proposed framework, several data selection techniques are studied, and accordingly, a total record of 247 elderly patients is extracted from the MIMIC-III clinical database.

The performance of the proposed framework is evaluated and compared against several state-of-the-art anomaly detection algorithms using both online and traditional metrics. The proposed framework achieved 83% NAB score which outperforms the HTM and k-NN algorithms by 15%, the HBOS and INFLO SVD by 16% and the k-NN PCA by 21% while the SVM scored 34%. The results prove that multiple HTM networks can achieve better performance when dealing with multi-dimensional data, i.e. data collected from more than one source/sensor.

## **Acknowledgements**

First and foremost, I would like to express my sincere gratitude to my supervisors Dr. Russell Campion and Dr. Mohamed Sedky for the continuous support of my PhD study, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better supervisors and mentors for my PhD study.

Last but not least, I would like to thank my family: my parents and to my brothers who stood by me and offered me a lot of support through a number of difficult situations (despite continuously asking, “Is it finished yet?”). I would also like to thank my grandmother, your prayer for me was what sustained me thus far.

# Contents

<b>1 Introduction</b>	<b>1</b>
1.1 Motivation .....	2
1.2 Context of the Investigation .....	3
1.3 Aim .....	4
1.4 Objectives .....	4
1.5 Contribution to Knowledge .....	5
1.6 Hypothesis and Research Questions .....	6
1.7 Research Methodology .....	7
1.7.1 Secondary Research .....	8
1.7.1.1 Literature Review.....	8
1.7.1.2 Study of Theory Underpinning this Research .....	9
1.7.2 Primary Research.....	9
1.7.2.1 Proposal, Design and Implementation.....	9
1.7.2.2 Datasets Preparation .....	9
1.7.2.3 Test and Evaluation.....	10
1.8 Structure of the Thesis .....	10
<b>2 Literature Review</b>	<b>12</b>
2.1 Ambient Assisted Living Applications.....	13
2.2 Ambient Intelligence .....	15
2.2.1 Ubiquitous and Pervasive Computing.....	16
2.2.2 Wearable Technologies.....	18
2.2.3 Context-Awareness.....	21
2.2.4 Artificial Intelligence.....	24
2.3 Requirements for Vital Signs Monitoring in AAL.....	27
2.4 Data Mining for Vital Signs Monitoring.....	31
2.4.1 Data Acquisition.....	33
2.4.2 Data Pre-processing .....	34
2.4.3 Segmentation .....	34
2.4.4 Data Transformation.....	36
2.4.5 Data Modelling .....	37
2.4.6 Evaluation .....	37
2.5 Temporal Reasoning for Vital Signs Monitoring .....	41
2.5.1 Time in Health Monitoring .....	42
2.6 Machine Learning in Health Monitoring.....	45
2.6.1 Prediction.....	47
2.6.2 Anomaly Detection .....	47

2.6.3	Supervised Techniques.....	50
2.6.3.1	Classification .....	50
2.6.3.2	Regression.....	51
2.6.4	Supervised Algorithm.....	52
2.6.4.1	Support Vector Machine .....	52
2.6.4.2	k-Nearest Neighbour.....	56
2.6.4.3	Artificial Neural Network.....	57
2.6.5	Unsupervised Techniques.....	58
2.6.5.1	Clustering .....	58
2.6.5.2	Association.....	60
2.6.6	Unsupervised Algorithm .....	62
2.6.6.1	Histogram-Based Outlier Score .....	62
2.6.6.2	Local Outlier Factor .....	62
2.6.6.3	Influence Outlierness.....	63
2.6.6.4	Hierarchical Temporal Memory .....	64
2.7	Anomaly Detection in AAL – Related Work.....	65
2.8	Research Gaps.....	69
2.9	Summary .....	70
<b>3</b>	<b>Hierarchical Temporal Memory</b>	<b>73</b>
3.1	Cortical Facts.....	74
3.2	HTM Theory.....	77
3.2.1	HTM Principles.....	79
3.2.1.1	Hierarchy.....	79
3.2.1.2	Regions .....	80
3.2.1.3	Sparse Distributed Representations.....	81
3.2.1.4	Time.....	82
3.2.2	HTM Neuron Model.....	82
3.2.3	HTM Functions.....	84
3.2.3.1	Learning.....	84
3.2.3.2	Inference .....	85
3.2.3.3	Prediction.....	85
3.3	Cortical Learning Algorithms.....	86
3.3.1	Encoder .....	88
3.3.1.1	Numbers.....	90
3.3.1.2	Categories.....	90
3.3.1.3	Coordinate and Geospatial Coordinate.....	91
3.3.2	Spatial Pooler.....	91
3.3.2.1	Initialization of parameters.....	92
3.3.2.2	Overlap Score.....	93
3.3.2.3	Inhibition .....	94
3.3.2.4	Learning.....	95
3.3.3	Temporal Memory .....	96
3.3.4	Classifiers .....	98
3.3.4.1	SDR Classifier .....	99
3.3.4.2	CLA Classifier .....	101
3.3.5	Anomaly Detection .....	103
3.3.5.1	Anomaly Score.....	103

3.3.5.2 Anomaly Likelihood.....	104
3.4 HTM Applications.....	105
3.5 Summary.....	107
<b>4 Proposed Framework</b>	<b>109</b>
4.1 Challenges.....	110
4.2 Model Scenario.....	111
4.2.1 Temporal Change Detection in Vital Signs.....	113
4.2.2 Medical Condition.....	114
4.2.3 Correlation between Vital Signs.....	116
4.3 Proposed HTM Framework.....	117
4.3.1 Hierarchy.....	120
4.3.2 Regions.....	121
4.3.2.1 Encoding.....	123
4.3.2.2 Spatial Pooler.....	125
4.3.2.3 Temporal Memory.....	127
4.4 Summary.....	128
<b>5 MIMIC-III Database</b>	<b>130</b>
5.1 Database selection.....	131
5.2 MIMIC III.....	134
5.2.1 Data Acquisition and Processing.....	137
5.2.1.1 PATIENTS.....	138
5.2.1.2 ADMISSIONS.....	139
5.2.1.3 ICUSTAYS.....	141
5.2.1.4 TRANSFERS.....	141
5.2.1.5 SERVICES.....	142
5.2.1.6 CHARTEVENTS.....	143
5.2.1.7 D_ITEMS.....	144
5.3 Extracted Dataset.....	145
5.4 Summary.....	146
<b>6 Test and Evaluation</b>	<b>148</b>
6.1 Experimental Setup.....	149
6.1.1 Contexts.....	150
6.1.2 The Patients and Datasets.....	151
6.1.3 Experiment parameters.....	152
6.1.4 Evaluation metrics.....	153
6.1.4.1 NAB Score.....	154
6.1.4.2 Threshold Optimisation.....	154
6.1.4.3 F-measure.....	155
6.2 Results.....	155
6.2.1 HTM.....	156
6.2.1.1 HTM model.....	156
6.2.1.2 Proposed Model.....	157
6.2.2 k-Nearest Neighbour.....	164
6.2.3 Influence Outlierness.....	165

6.2.4 Histogram-Based Outlier Score .....	166
6.2.5 Support Vector Machine .....	167
6.3 Results Analysis and Discussion .....	167
6.4 Summary .....	169
<b>7 Conclusions and Future Work</b>	<b>171</b>
7.1 Contribution to Knowledge .....	177
7.2 Future Work .....	177
<b>8 References</b>	<b>180</b>
<b>9 Appendices</b>	<b>194</b>

# List of Figures

Figure 1-1 Deductive Process .....	7
Figure 2 AAL system for delivering different type of service (Al-Shaqi, Mourshed and Rezgui, 2016).....	14
Figure 2-2 Ambient Intelligence Situated (Kofod-Petersen, 2007) .....	16
Figure 2-3 Wearable Technologies for elderly care (Wang, Yang and Dong, 2017).....	20
Figure 2-4 Context-Aware Power Management subsystem for AlarmNet (Adopted from (Wood <i>et al.</i> , 2008) ).....	23
Figure 2-5 AI Fields, Methods and Techniques (Rech and Althoff, 2004).....	26
Figure 2-6 Data mining tasks including anomaly detection, prediction and clinical diagnosis (Banaee, Ahmed and Loutfi, 2013).....	30
Figure 2-7 A generic architecture of the main data mining approach for raw data (based on Palaniappan and Awang, 2008; Banaee, Ahmed and Loutfi, 2013).....	33
Figure 2-8 Observation, Lead, and Prediction windows (Forkan and Khalil, 2017a) .....	36
Figure 2-9 An example of anomaly windows in stream data (Lavin and Ahmad, 2015) .....	39
Figure 2-10 NAB scoring function (Lavin and Ahmad, 2015).....	40
Figure 2-11 An example of Interval-based approach including the trend and value abstractions (Batal <i>et al.</i> , 2011) .....	44
Figure 2-12 Machine Learning types is based on the availability of the training and testing dataset (adapted from (Goldstein and Uchida, 2016)) .....	46
Figure 2-13 An example of points anomalies in a two-dimensional space (Chandola, Banerjee and Kumar, 2009) .....	48
Figure 2-14 An example of contextual anomaly in a monthly temperature data (Chandola, Banerjee and Kumar, 2009).....	49
Figure 2-15 An example of collective anomaly in a patient ECG (Chandola, Banerjee and Kumar, 2009).....	49
Figure 2-16 An example of Logistic regression for estimation of risk level for various medical conditions (Gennings, Ellis and Ritter, 2012) .....	52
Figure 2-17 Separation of two classes in linear SVM (Amiribesheli, Benmansour and Bouchachia, 2015) .....	53
Figure 2-18 An example of Multilayer Feedforward ANN (Amiribesheli, Benmansour and Bouchachia, 2015) .....	57
Figure 2-19 K-Means clustering used for patient's classification (Liu <i>et al.</i> , 2015) .....	59
Figure 2-20 An example of rules of four clinical events for the multivariate temporal parameters (adapted from (Banaee, Ahmed and Loutfi, 2015)).....	61
Figure 2-21 A comparison between the INFLO algorithm and the LOF algorithm (Goldstein and Uchida, 2016) .....	64

Figure 2-22 Components of HTM (adapted from(Cui, Ahmad and Hawkins, 2017))	65
Figure 2-23 The SmartHabits Expert System (Grguri, Mošmondor and Huljeni, 2019)	68
Figure 3-1 six layers of neocortex (adapted from He <i>et al.</i> , (2017))	75
Figure 3-2 primary components of the neuron (Devineni, 2015)	75
Figure 3-3 Neuronal spike initiation site	77
Figure 3-4 HTM Theory – layers of columns in HTM region	78
Figure 3-5 Information flows up and down hierarchies (Hawkins and Blakeslee, 2005)	80
Figure 3-6 - Implementation of a section of an HTM region comprised of columns of cells	81
Figure 3-7 A Sparse Distributed Representation of an Information in an HTM Region	82
Figure 3-8 A Comparison between a biological neuron and the HTM neuron (Hawkins and Ahmad, 2016)	84
Figure 3-9 Functional steps for using HTM on real-world sequence learning tasks	87
Figure 3-10 - The SDR at column level	88
Figure 3-11 - SDR at the level of individual cells	88
Figure 3-12 Columns in the SP and the potential connections	93
Figure 3-13 Input data to SDR Classifier	99
Figure 3-14 Feedforward Classification Network in SDR Classifier (Bobak, 2017)	100
Figure 3-15 Input data to CLA Classifier	102
Figure 3-16 The primary functional steps for Anomaly Score and Anomaly Likelihood Process (Bobak, 2017)	105
Figure 3-17 Flowchart of custom detector (Mitri <i>et al.</i> , 2017)	106
Figure 3-18 High-level overview of a performance anomaly detection framework for scientific workflows (Rodriguez, Kotagiri and Buyya, 2018)	107
Figure 4-1 Scenario for the proposed framework	112
Figure 4-2 24-hour temporal patterns. (a) SBP, (b) DBP, and (c) HR (adapted from (Morris et al., 2013))	114
Figure 4-3 Time-series graph of 4 vital signs for an elderly patient with hypotension condition	116
Figure 4-4 Proposed Framework	118
Figure 4-5 Hierarchy of regions in proposed framework	120
Figure 4-6 Standard HTM system - One region is allocated for decoding multiple vital signs	122
Figure 4-7 Proposed framework - multiple regions allocated for decoding multiple vital signs	122
Figure 4-8 Sparse Distributed Representation of timestamp in proposed framework	124
Figure 4-9 Example of sparsity ratio in input SDR and output SDR (adapted from (Cui, Ahmad and Hawkins, 2017))	126

Figure 5-1 Overview of the MIMIC-III database (A. E. W. Johnson <i>et al.</i> , 2016).....	136
Figure 5-2 Entity Relationship Diagram of selected Tables.....	138
Figure 6-1 The Experimental setup.....	149
Figure 6-2 An applied period-based approach for temporal abstraction .....	151
Figure 6-3 The NAB Score for the proposed model with different column dimensions .....	159
Figure 6-4 The NAB Score for the proposed model with different Boost Strength values.....	160
Figure 6-5 The NAB Score for the proposed model with different SynPermActiveInc values.....	160
Figure 6-6 The NAB Score for the proposed model with different maxSynapsesPerSegment values .....	161
Figure 6-7 The NAB Score for the proposed model with different synPermInactiveDec values.....	162
Figure 6-8 The NAB Score for the proposed model with different PermanenceDecrement values.....	162
Figure 6-9 k-NN model F-measure results .....	165
Figure 6-10 INFLO model f-measure results for different k-values .....	166
Figure 6-11 The Results for all evaluated models .....	168

# List of Tables

Table 2-1 Common vital signs and symptoms.....	28
Table 2-2 Taxonomy of extracted features in three domains (adapted from (Ni, Hernando and de la Cruz, 2015) ) .....	36
Table 2-3 An Example of representation of input space using kernels (Fischer and Mougeot, 2018) .....	54
Table 2-4 Performance comparison of SVM, J34 and Naive Bayesian ( adapted from (Raji, Jeyasheeli and Jenitha, 2016)) .....	55
Table 2-5 An example of textual representation of the constructed rules (adapted from (Banaee, Ahmed and Loutfi, 2015)) .....	61
Table 3-1 Parameters are used in the SP algorithm for initialization.....	93
Table 3-2 Parameters are used in the SP algorithm for inhibition .....	94
Table 3-3 Parameters are used in the SP algorithm for learning.....	96
Table 4-1 The vital signs and their generalised normal value.....	115
Table 4-2 The vital signs value and medical condition.....	115
Table 5-1 Clinical Database Comparison.....	132
Table 5-2 Characteristics of the MIMIC-III database .....	136
Table 5-3 Detailed description of PATIENTS table .....	139
Table 5-4 Detailed description of ADMISSIONS table .....	140
Table 5-5 Detailed description of ICUSTAYS table .....	141
Table 5-6 Detailed description of TRANSFERS table.....	142
Table 5-7 Detailed description of SERVICES table .....	143
Table 5-8 Detailed description of CHARTEVENTS table .....	144
Table 5-9 Detailed description of D_ITEMS table .....	145
Table 5-10 Characteristics of extracted dataset.....	146
Table 5-11 Example of features in generated dataset.....	146
Table 6-1 The 247 datasets and the corresponding number of records.....	152
Table 6-2 HTM model results .....	157
Table 6-3 Proposed model with different 'numActiveColumnPerInhArea' values .....	159
Table 6-4 Proposed model results .....	163
Table 6-5 k-NN model results.....	165
Table 6-6 INFLO model results .....	166
Table 6-7 HBOS model results.....	167
Table 6-8 SVM model results .....	167
Table 9-1 Description of each service type (A. E. W. Johnson <i>et al.</i> , 2016).....	217

# List of Abbreviations

ALS	<b>Assisted Living System</b>
AAL	<b>Ambient Assisted Living</b>
AmI	<b>Ambient Intelligence</b>
AI	<b>Artificial Intelligence</b>
BN	<b>Bayesian Network</b>
BUP	<b>Bottom-UP</b>
CLA	<b>Cortical Learning Algorithms</b>
DBP	<b>Diastolic Blood Pressure</b>
DM	<b>Data Mining</b>
FNSW	<b>Fixed-size Non-overlapping Sliding Window</b>
FOSW	<b>Fixed-size Overlapping Sliding Window</b>
HBOS	<b>Histogram-Based Outlier Score</b>
HCI	<b>Human Computer Interaction</b>
HMM	<b>Hidden Markov Model</b>
HR	<b>Heart Rate</b>
HTM	<b>Hierarchal Temporal Memory</b>
INFLO	<b>INFLuences Outlierness</b>
KDD	<b>Knowledge Discovery in Database</b>
KNN	<b>K-Nearest Neighbours</b>
LOF	<b>Local Outlier Factor</b>
LOCI	<b>LOcal Correlation Integral</b>
ML	<b>Machine Learning</b>
MIMIC-III	<b>Medical Information Mart for Intensive Care</b>
NAB	<b>Numenta Anomaly Benchmark</b>
NN	<b>Neural Network</b>
PCA	<b>Principal Component Analysis</b>

SBP	<b>S</b> ystolic <b>B</b> lood <b>P</b> ressure
SDR	<b>S</b> parse <b>D</b> istributed <b>R</b> epresentations
SP	<b>S</b> patial <b>P</b> ooler
SVM	<b>S</b> upport <b>V</b> ector <b>M</b> achines
SW	<b>S</b> liding <b>W</b> indow
TM	<b>T</b> emporal <b>M</b> emory
ToD	<b>T</b> op- <b>D</b> own
TR	<b>T</b> emporal <b>R</b> easoning
VSW	<b>V</b> ariable-size <b>S</b> liding <b>W</b> indow

## Chapter 1

# INTRODUCTION

Recently, the Ambient Intelligence (AmI) paradigm is used for enabling technologies in different scenarios to empower people's capabilities by means of a digital environment. The AmI technologies have been widely applied in the field of healthcare monitoring and smart homes where an Assisted Living System (ALS) provides healthcare services to older and disabled residents living in their homes or in supported housing. The ALSs that are empowered by AmI technologies also are referred to as Ambient Assisted Living (AAL). A series of recent studies has indicated that AAL is a new philosophy that includes methods, products, services, and AAL systems to enrich the lives of elderly and disabled people. For instance, according to (Lewin *et al.*, 2010) the current use of AAL systems in delivering care to older people are distinguished based on the types of services they provide to the elderly people. These services include telehealth (health monitoring) services, telecare services, wellness services and digital participation services.

One of the major topics of investigation in this field is the AAL system that provides health monitoring services to manage long-term conditions of elderly people such as Chronic Obstructive Pulmonary Disease (COPD), heart failure and diabetes. In this context, the health states of the elderly people are monitored

using multiple vital signs and measurement technologies such as those used for Heart Rate (HR), Blood Pressure (BP), respiration rate, body temperature, pulse oxygenation and blood glucose (Patel *et al.*, 2012).

## 1.1 Motivation

One of the motivations for AAL technologies is the need for health monitoring and assistance especially with a significant worldwide increase in an aging population. In fact, according to a report from the Office for National Statistics<sup>1</sup> in 1948, 11% of the population of Great Britain was aged 65 and older. By 2016, this had risen to 18%. That is an increase from around five million to just over 11.5 million people. Although according to the United Nations<sup>2</sup>, the population of older people across the globe is expected to be doubled by 2025 to around 1.2 billion. Demographic changes in the UK, such as an ageing and expanding population has led to an increase in the number of people with complex healthcare needs. This has contributed to raising healthcare costs. Plum's report (Lewin *et al.*, 2010) has pointed out that the health applications of AAL technologies can reduce the cost of equipping a home for telecare or telehealth substantially, perhaps from £2000 to £200 by 2030. The Department of Health has conducted the Whole System Demonstrator (WSD) programme<sup>3</sup> in 2011, to demonstrate potential benefits of applying AAL technologies for delivering the health and care services. The result from the WSD programme indicates 15% reduction in A&E visits, 14% reduction in bed days, 20% reduction in emergency admissions, 5.8% reduction in tariff costs, 14% reduction in elective admissions and more extraordinarily a 45% reduction in mortality rates (Steventon, Bardsley and Billings, 2012; Celler and Sparks, 2015).

---

<sup>1</sup> <http://www.un.org/en/index.html>

<sup>2</sup> <https://www.ons.gov.uk/>

<sup>3</sup> <https://www.gov.uk/government/news/whole-system-demonstrator-programme-headline-findings-december-2011>

Hence in this study, it was of interest to investigate Machine Learning (ML) techniques, in particular, Hierarchical Temporal Memory (HTM) theory to model the spatiotemporal patterns of vital signs and to detect anomalous data points preceding a health-critical event. The proposed HTM-based framework will improve the quality of the health monitoring service by assisting health professionals in making better decisions in addition to the improvement of response time to a critical health event.

## 1.2 Context of the Investigation

The need for AAL technology for monitoring the health condition of older people to prevent critical events (e.g. cardiac arrest) in this area is obvious by looking at current projects and future demographics. In AAL health monitoring, detection of abnormal patterns of vital signs is made possible by applying ML techniques. However, current ML techniques fall short of reaching the full potential of anomaly detection. This is due to a number of facts that AAL in health monitoring scenario must: (i) be adaptable to the new normal patterns, (ii) apply a contextual anomaly detection approach, to detect anomalous data points, (iii) detect an anomalous point as early as possible preceding a critical health event and (iv) employ a common learning technique that can be used for clinical decision support to discover patient specific anomalies independently.

However, the last decade has witnessed the disruptive success of deep hierarchical machine learning approaches to offer working solutions for a multitude of practical problems. One good example is the Hierarchical Temporal Memory (HTM) theory (Hawkins and Ahmad, 2016), which is inspired by recent discoveries in neuroscience that mimics the attributes and hierarchical structure of neurons in the neocortex. Cortical Learning Algorithms (CLAs) are new biologically inspired machine intelligence technologies based on HTM theory which are scalable, adaptive to different modalities, universal in using the same learning principles, enable continual learning, and recognise new and changed behaviour without manual intervention.

HTM theory, takes into consideration three factors that are crucial for intelligence: time, feedback and the physical hierarchical structure of the brain.

Time in the AAL health monitoring scenario is also an important parameter as it is used by health professionals in diagnosis and prognosis. Hence, these features of HTM theory make it suitable for a health monitoring and assistance scenario where vital signs of patients are monitored periodically (temporal patterns).

### 1.3 Aim

The aim of this thesis is to investigate the application of HTM theory to Ambient Assisted Living and to propose a novel and adaptive HTM-based framework to detect anomalies in vital signs as part of an AAL solution.

The novelty of the proposed framework will be considered by applying multi-level hierarchy of HTM theory, and its adaptability will be achieved by learning the new normal patterns of vital signs collected from elderly patients.

### 1.4 Objectives

This thesis is concerned with establishing a theoretical underpinning for the detection of abnormal patterns of vital signs collected from patients to prevent critical health events in AAL health monitoring scenarios. Hence, the following objectives are defined to achieve the aim of this thesis:

- Provide an intensive research on existing AmI paradigm and AAL applications used for health monitoring and assistance scenarios.
- Explore the anomaly detection literature in the context of AAL health monitoring.
- Investigate data types and characteristics of vital signs collected from older people.
- Study different contextual modelling techniques in particular those used to abstract temporal and spatial patterns in the vital signs.
- Review the use of HTM theory in the context of sensed vital sign data.
- Research and explore different techniques of Data Mining (DM) including the Electronic Health Record (EHR) databases.
- Propose a novel and adaptive framework, motivated by the HTM theory in the context of elderly's health monitoring.

- Evaluate the performance of the proposed framework against the performance of various state-of-the-art anomaly detection algorithms.

## 1.5 Contribution to Knowledge

The novel contributions of this thesis are summarised as follows:

- The primary contribution is a novel and adaptive framework, inspired by the HTM theory, for vital signs monitoring in an AAL system. The novelty of the proposed framework will be considered by applying multi-level hierarchy of HTM theory, and its adaptability will be achieved by learning the new normal patterns of vital signs collected from elderly patients.
- The secondary contribution of this work is a pre-processed subset of the MIMIC-III (Medical Information Mart for Intensive Care) database that includes appropriate vital signs parameters of elderly people with different medical conditions, including Hypotension, Hypertension, Tachycardia and Bradycardia.
- Another secondary contribution is the performance evaluation of the proposed framework using both on-line processing as well as batch processing metrics. The Numenta Anomaly Benchmark (NAB) is used to evaluate the performance of on-line machine learning methods and other performance metrics such as recall, precision and F-measure are used as a traditional evaluation technique for the algorithms that use batch processing.

## 1.6 Hypothesis and Research Questions

Two hypotheses underlie the research in this thesis:

**HYP 1:** The hierarchical and temporal principles of the HTM theory would outperform the accuracy of current state-of-the-art machine learning algorithms in detecting anomalous points preceding a health-critical event.

**HYP 2:** HTM theory can be used in an AAL health monitoring scenario to prevent a health-critical event by modelling the patterns of vital signs.

The following research questions were formulated to investigate these hypotheses:

**RQ1:** Can HTM be applied to detect anomalous points preceding a health-critical event by modelling spatio-temporal patterns of multiple vital signs?

**RQ2:** How does hierarchical organisation of the HTM improve abstraction of semantical information from multiple vital signs?

**RQ3:** What are the ML techniques and algorithms that can be applied in a health monitoring scenario to prevent a critical health event?

**RQ4:** Does the use of temporal information improve the anomaly detection in health monitoring scenarios?

**RQ5:** What are the data mining tasks required to prepare datasets for the evaluation of ML algorithms in the context of AAL health monitoring?

All five research questions attempt to answer different aspects of the first hypothesis. RQ3 and RQ4 specifically examine the second hypothesis. To examine RQ3, the results of RQ4, and RQ5 are essential to focus on the suggested approaches and practices in the AI paradigm.

## 1.7 Research Methodology

The research methodology comprises research strategies and methods related to data collection and analysis in order to achieve the project's objectives. According to Brown, 2006, the research methodology is *the philosophical framework within which the research is conducted or the foundation upon which the research is based*. The research onion model proposed by Saunders et al. (2007) highlights different components of the research methodology which is commonly used by researchers to discuss development of their research methodology. It comprises six layers and the first topic to be clarified in a research methodology is the research philosophy positioned at the outermost layer of the research onion model. It deals with the identification of the research philosophy behind the source, nature and development of any knowledge.

This research adopts positivist philosophy/study which is based on the idea that science is the only way to find about the truth. In positivism studies, the belief is that the research is objective and is conducted with minimum interaction with research participants during the research. Among three research approaches identified in research onion model, the deductive approach is selected. As a set of hypotheses are developed based on HTM theory, hence deductive approach suits better to testify (confirm or reject) these hypotheses during the research process. In a deductive approach, research begins by exploring a known theory and tests if the theory is valid in a given context. Hence findings are usually observable and quantifiable. On the other hand, an inductive approach is opposite to a deductive approach in which research starts from the general to the specific. Figure 1-1 Highlights deductive process conducted in this thesis.



Figure 1-1 Deductive Process

Moreover, the quantitative data collection method is selected based on three main factors. Firstly, that a quantitative research method is usually associated with the positivist philosophy and deductive research approach. Secondly, because in this

thesis a large volume of data from elderly patients is used in the experimentation and thirdly, in the experimentation phase, the hypotheses are tested by applying statistical metrics including NAB score, F-measure, recall and precision. The findings from experimentation and testing the performance of the proposed HTM framework are analysed to evaluate the hypotheses.

Hence this research adopts a positivist view and follows the deductive approach starting from the HTM theory and uses quantitative research method to prove HTM's applicability to detect anomalies in vital signs.

The main phases of this research are summarised below.

### 1.7.1 Secondary Research

The secondary research comprises of two phases:

#### 1.7.1.1 Literature Review

The literature review serves as an overview of relevant work to assist in the design and implementation of a novel framework to detect anomalous data points prior to a critical health event. The study attempts to focus on sources relevant to AI and includes important related studies from fields such as AmI, data mining, anomaly detection, machine learning, temporal reasoning, and AAL health monitoring scenarios.

The search for literature was based on queries with combinations of the following keywords: AI, anomaly detection in health monitoring scenario, assisted living system, hierarchical temporal memory, data mining, temporal reasoning, time in health monitoring, context awareness computing, and machine learning algorithms. Queries began through the ScienceDirect, MDPI, IEEE Xplore Digital Library, and ACM Digital Library. Furthermore, the reference list included in publications from reliable journals and conferences were also included to investigate relevant research. The literature review follows the quantitative view of performance of the anomaly detection algorithms in a health monitoring scenario. Moreover, to train and test the proposed framework, a dedicated research is conducted to select a dataset that represents the real medical record

of the patients, including records of their vital signs. This information is analysed to define the research gap and requirements of the anomaly detection algorithms in AAL health monitoring scenario.

#### 1.7.1.2 Study of Theory Underpinning this Research

Further research on the HTM theory is conducted to gain a better insight on hierarchical organisation of the HTM, this is followed by a study of the main principles of the HTM theory including hierarchy, regions, sparse distributed representation and time. The findings of this phase are analysed to establish the stability and capability of the HTM theory in detecting anomalous data points in the context of health monitoring.

### 1.7.2 Primary Research

The primary research deals with further exploration of the information collected in the literature review phase. In this phase, a more systematic and theoretical analysis of theory underpinning this research is conducted. The outcome of this research phase forms the backbone for the design, implementation and subsequently the performance evaluation of the proposed solution.

#### 1.7.2.1 Proposal, Design and Implementation

A novel and adaptive framework is proposed, designed and implemented for health monitoring and assistance scenarios. The proposed framework is inspired by the HTM theory, and it is built on the CLAs. The novel framework is implemented to monitor the health condition of an older patient by detecting abnormal patterns of several vital signs.

#### 1.7.2.2 Datasets Preparation

Investigating and comparing several medical databases, followed by justification for selecting MIMIC-III (Medical Information Mart for Intensive Care) database. This step is followed by the extraction of a set of datasets that are pre-processed and processed.

### 1.7.2.3 Test and Evaluation

An experiment is conducted to evaluate and compare the performance of the proposed model with several state-of-the-art algorithms (discussed in Chapter 6). The outcome of this phase is used to test the hypothesis that if the HTM theory improves the accuracy of current state-of-the-art algorithms in detecting anomalous points preceding a health-critical event.

## 1.8 Structure of the Thesis

This thesis is organised as follows:

*Chapter 1* introduces the thesis by describing the aim and objectives along with the research contribution to knowledge and research methodology used throughout the remainder of the thesis.

*Chapter 2* provides a survey of the literature on AmI paradigm, AAL health scenario, data mining, and several anomaly detection algorithms. It concludes by setting the direction and scope of the research that is undertaken in the thesis.

*Chapter 3* begins with an exploration of the neocortical theory of what exactly constitutes HTM network followed by its sophisticated biologically-inspired algorithms, the CLA.

*Chapter 4* considers fundamental aspects of the process of developing a novel framework, which is inspired by the HTM theory. It proposes a model scenario and notion formalisation of time based on the model scenario, in order to provide means to represent and reason about the temporal aspect of knowledge. Based upon this a novel and adaptive framework is proposed and implemented in this Chapter.

*Chapter 5* documents investigation and comparison of several medical databases, followed by justification of the selected MIMIC-III database, resulting in a set of extracted datasets which are used in Chapter 6 for experimentation and testing of the proposed framework.

*Chapter 6* examines the performance of the novel model proposed in Chapter 4, followed by comparison of its performance with the standard HTM model and with state-of-the-art anomaly detection algorithms.

*Chapter 7* summarises the results obtained in the thesis and presents a number of possible avenues for fruitful future research.

## Chapter 2

# LITERATURE REVIEW

This Chapter begins with a grounding in the principles underlying AmI paradigms providing the health care services to the older residents who are living in their homes or in supported housing. The study on AAL technologies will be carried out to explore the complexity of diagnosis and prognosis in healthcare where multiple vital sign parameters are obtained from the patients. Several previously-published literatures on the subject will be reviewed including works related to the health care applications for monitoring health status of the patients in Critical Care Units (CCU) scenarios. Furthermore, a set of technologies that should be considered for the success of AAL systems will be investigated. Next, the research will be expanded to study several techniques of context modelling such as spatio and temporal reasoning, followed by inspections of the ML techniques and algorithms that are used for anomaly detection in healthcare.

Later in this Chapter, research gaps will be discussed, looking in detail at the major drawback of the existing approaches to the problems of anomaly detection in health monitoring and assistance scenario in the context of elderly people.

The information reviewed in this Chapter will be used to form the basis for the work to be performed, and as an indicator of the structure of the reminder of the thesis.

## 2.1 Ambient Assisted Living Applications

One of the major challenges of the ageing population is the efficient delivery of healthcare services. The wellbeing of the elderly people is of concern to their family members, in case they are alone in their homes and some unusual situations occur, affecting their health states. The alternative such as care in nursing homes or hospitals is costly and cost increases further if specialised care is mobilised to patients' place of residence. With the maturing of enabling technologies for AAL have become a reality. The basis of such applications is the continuous care and most often real-time monitoring of the environment and occupant behaviour, applying an AI solution, through providing a means for monitoring and assessment, and triggering assistance as and when needed (Al-Shaqi, Mourshed and Rezgui, 2016).

A typical AAL system which is applied in an elderly home to monitor the behaviour of a patient through wireless sensor technologies, link the health caregiver and family member to the elderly's home. Hence, family members can use the system to monitor health states of their loved-one online anywhere and anytime, to have a peace of mind that their loved-one is in a good health. In addition, health professionals can use the system for health assessment and illness diagnosis.

According to (Lewin *et al.*, 2010), AAL applications are grouped into heterogeneous categories subject to the type of services they are delivering to the elderly people. These services include:

- Telehealth services - delivering medical care, treatment, or monitoring services. E.g. monitoring vital signs to manage long term condition such as heart failure.
- Telecare services – delivering social care or monitoring services. E.g. a voice memo system to remind elderly patient with short term memory condition to take keys and turn off gas/electric before leaving the home.

- Wellness services – delivering services for healthier lifestyles. E.g. an adaptive lighting can contribute to healthy sleep (Cook, Augusto and Jakkula, 2009).
- Digital participation services – enabling social, educational or entertainment activities.
- Teleworking services – enabling elderly and disabled people to work remotely from home for an employer.

A recent study by Al-Shaqi, Mourshed and Rezgui, (2016) has separated AAL applications into different types based on the target group, provided support, and technology demonstrated in an AAL system. As shown in Figure 2-1, one of the applications of an AAL system is health monitoring where the vital signs (e.g. heart rate, blood pressure and body temperature) of elderly patients can be retrieved for the purpose of health condition monitoring and/or to receive help from designated health professionals when needed.

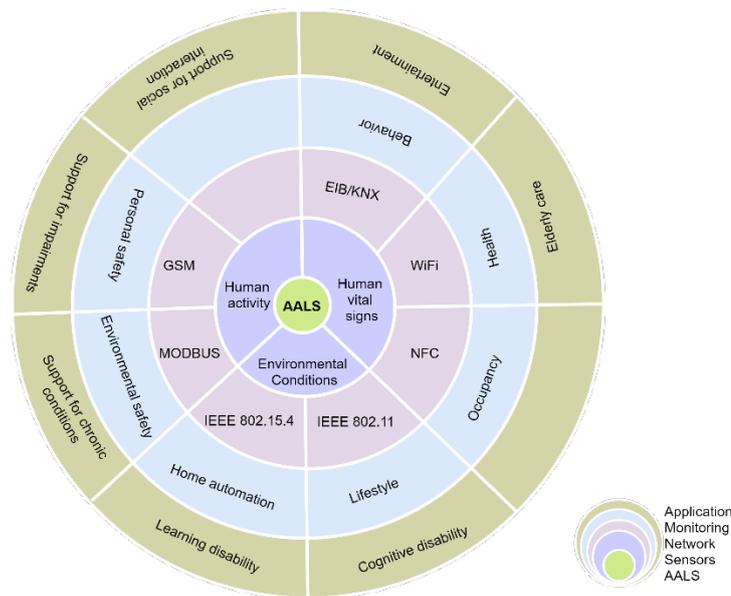


Figure 2-1 AAL system for delivering different type of service (Al-Shaqi, Mourshed and Rezgui, 2016)

The difference between a traditional assisted living system and AAL system is in the type of technologies that are used to add transparency and intelligence to the traditional assisted living system. For an assisted living system, in particular a

health monitoring application, to be Ambient Intelligence (AmI) it is required to be transparent to the surrounding environment, to use and incorporate multiple sensors in order to collect vital signs data from the elderly patient, to take an action according to the patients' current context and to reason and adapt to new changes.

The transparency of an AAL, in particular a health monitoring application, is achieved by the most profound technologies defined as ubiquitous and pervasive computing (discussed in Section 2.2.1). The wireless technologies, in particular wearable technologies, enable the health monitoring application to collect vital signs health data from the elderly (discussed in Section 2.2.2). As discussed earlier in this Chapter, there is a different type of AAL application for delivering different services, hence the AAL health monitoring application must be context-rich (discussed in Section 2.2.3), to have knowledge or reception of a situation. Usually, the vital signs health data collected from the elderly patient are not sufficient for the AAL to reason and adapt to the new changes unless Artificial Intelligence (AI) techniques are used (discussed in Section 2.2.4).

## 2.2 Ambient Intelligence

Ambient Intelligence or in short, AmI, is a new era in Information Technology (IT) in which the “enabling technologies” are used in our surroundings to support us in our daily lives. The European Commission’s Information Society Technologies Advisory Group (ISTAG) introduced the concept of ambient intelligence (Ramos, Augusto and Shapiro, 2008). Computers are already influencing our daily lives, and there are even more projects and researches that are directed to the way that the technologies can be used to help our society. In fact, recent advancements in the technologies, have offered a new set of services and scenarios where people can be surrounded by systems that are intelligent enough to assist them proactively, while not intruding upon people. Whether it is indoor a health monitoring and assistance to monitor the medical condition of an older person, or it is a transport station providing services to the passengers, or an Intensive Care Unit (ICU) in hospital helping to care for a patient. The AmI services are harmonised by “intelligent systems” that integrate the resources

available to provide an “intelligence environment”. This convergence of topics has led to the area of “ambient intelligence”. In other words, as discussed by Kofod-petersen, 2007, the AmI environment is based on a relationship between three computing paradigms: Ubiquitous, Pervasive and Artificial Intelligence (AI). Basically, as illustrated in Figure 2-2, these paradigms can be referred to a set of technologies and principles that should be considered for the success of AmI systems. They bring new features to the AmI system compared to the traditional ones such as: transparent, sensitive, responsive, adaptive and intelligent (Cook, Augusto and Jakkula, 2009).

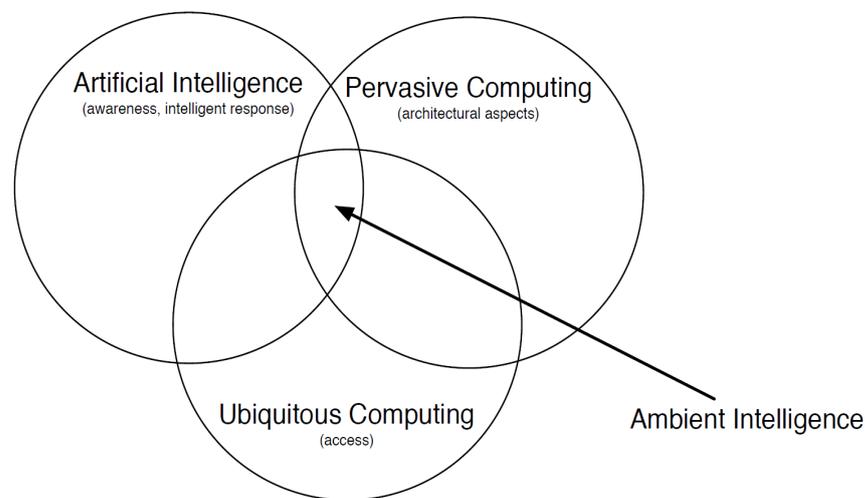


Figure 2-2 Ambient Intelligence Situated (Kofod-Petersen, 2007)

### 2.2.1 Ubiquitous and Pervasive Computing

The terms ‘Ubiquitous computing’ and ‘Pervasive computing’ are often used interchangeably. Mark Weiser a scientific American was the first to envisage in 1991, the ubiquitous computing paradigm in his article “*The Computer for the 21st Century*”, he described Ubiquitous computing as “*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*” (Weiser,1991). He had this vision that in the future, the world will be occupied by a several artefacts empowered with computing capabilities, which would come to be part of our lives and assist us in our daily activities. The fact that computers will start to fade into the background, does not mean they will visually disappeared but they will

operate in such a way that we would stop thinking about the fact that we were using a computer and we would begin to work through them, and they would become an extension of ourselves.

An analogy to this would be cars, when they first came out it was a big deal to have a car and then in the next few generations cars were almost like family pets and now, they are just around us and we don't give much attention to them at all.

Even though the terms ubiquitous computing and pervasive computing often are used interchangeably, the term ubiquitous computing is often preferred when dealing with the original idea of augmenting an ordinary artefact to allow access to the digital world. Whereas, the term pervasive computing is used when investigating systems, architectures, and integration of systems into human surroundings (Satyanarayanan, 2001). The pervasive term often is used when there are multiple sensor devices, collecting different information from the surroundings. In AAL health monitoring, sensed vital signs from multiple sensors should be combined to be pervasive and responsive to the environment's changes (Atallah, Lo and Yang, 2012).

Nowadays, ubiquitous and pervasive technologies are also covered in the *distributed computing* and *mobile computing* research fields, dealing with transparency and invisibility features to the AmI. For instance, wearable sensor technology is available to monitor the activity of individuals. Wearable sensor devices could integrate into Wireless Sensor Networks (WSN) (Zhang *et al.*, 2010), to transfer perceived data to a remote base station using technologies such as Wi-Fi, Bluetooth and ZigBee.

The research community in this field, established a list of objectives for the development of wireless platforms to be small, inexpensive, and to have a low power consumption that can automatically form ad-hoc wireless communication networks when they are deployed in an indoor or outdoor environment (Bose, 2009). The basic hardware design principle for these platforms is to integrate sensors, computations and communication in a single unit. The first generation of these platforms was commercialised by CrossBow Technologies in 1999, it is known as the Berkley Rene. The Rene was based on AT90LS8535 processor with

8 KB of program memory and 0.5 KB of RAM that used a radio with a data rate of 10 Kbps.

Some other popular platforms are: Mica, MicaZ, Telos, TelosB, SHIMMER 3 (Wood *et al.*, 2006). Other works on wireless sensor technologies, include the ‘Smart-Its’ project funded by the European Union’s Disappearing Computer Initiatives. One of the objectives in the Smart-Its project was design of a wireless technology to digitally tag and interconnect to specific everyday objects (such as teacups, keys and toys), or ‘Phidgets’ an innovative sensor and actuation platform developed at University of Calgary (Ko, Lu and Srivastava, 2010), (Kuryloski and Giani, 2009).

Next, the ubiquitous wearable technologies in AmI will be discussed. The wearable technology is a key factor for Weiser’s vision become a reality.

### 2.2.2 Wearable Technologies

At the present time, wireless sensor devices are available in different forms such as portable sensors, wearable sensors, and embedded sensors which are used for monitoring object position or room temperature in smart homes. We are seeing an appearance of implantable medical sensors to frequently measure the health status and vital signs data in AAL health monitoring scenarios (Ko, Lu and Srivastava, 2010). Hence, the wireless sensor technologies play an important role for the AmI concept to become a reality where a network of tiny sensor devices in the AmI environment is observing the surroundings and transmits the perceived data back to a remote base station using technologies such as Wi-Fi, Bluetooth and ZigBee.

For instance, smart-phones or smart-watches are ubiquitously carried by people everywhere and every-day, or smart clothing that incorporates technologies capabilities into existing wear. For example, Heddoko™ (Montreal, Canada) has developed smart shirts that collect human movement data that can be viewed in real time or saved for later playback via a mobile application, or smart shirts by Hexoskin, Cityzen Sciences, Ralph Lauren Polo, and Athos that measure vital

signs such as heart and respiratory rates for monitoring the status of several human physiological properties (Wang, Yang and Dong, 2017).

In addition to the above projects, wearable devices are investigated in several literatures in the context of AAL services for the older people. The wearable devices can be grouped into three categories based on the AAL scenarios. We use Figure 2-3 to describe three AAL scenarios where wearable technologies are deployed as follow:

- Human Activity Recognition (HAR): continuously monitor the body motion of elderly people using wearable sensors such as accelerometers and gyroscopes (Patel *et al.*, 2012). For instance, the accelerometers are used to monitor the elderly person if he/she is lying down, walking, standing and so on.
- Indoor Positioning: continuously monitoring the location of elderly people or objects in an indoor environment. For example, the embedded RFID tags can be used to automatically track the elderly people and objects.
- Health Monitoring and Assistance: continuously monitoring the health status of the elderly people using several vital signs and measurement technologies such as those used for Heart Rate (HR), Blood Pressure (BP), respiration rate, body temperature, pulse oxygenation and blood glucose (Patel *et al.*, 2012).

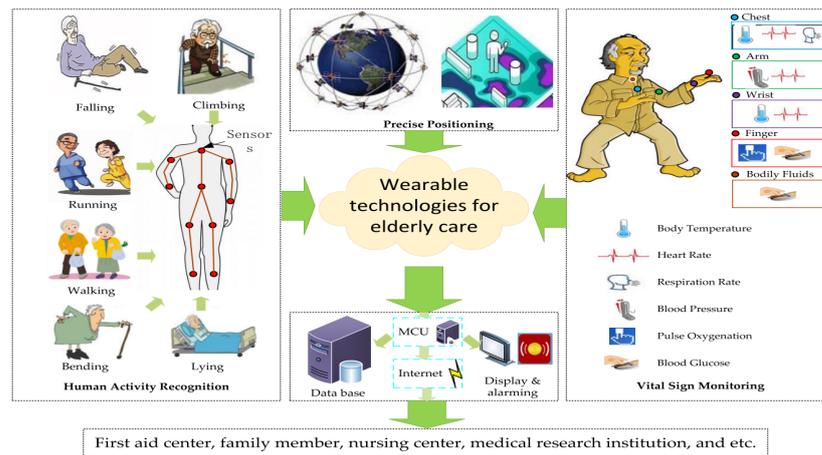


Figure 2-3 Wearable Technologies for elderly care (Wang, Yang and Dong, 2017)

So now that wearable and embedded wireless technologies are available for ubiquitous and pervasive AmI, the perceived data somehow should represent the environment in which they are situated. From a Human Computer Interaction (HCI) perspective, a user interacts with the system through the User Interface (UI). This type of interaction that requires a dialog between the user and a system is formalised as *explicit input* or *explicit interaction* which are common types of interaction in traditional systems. The explicit interactions challenge the idea of invisible computing, pervasive computing, and AmI paradigm. Hence, a new paradigm is proposed by Albrecht Schmidt (Kuryloski *et al.*, 2008; Wang, Yang and Dong, 2017) as implicit Human Computer Interaction (iHCI). In the iHCI paradigm, a new form of input to the system is identified as *implicit input* or *implicit interaction*. An implicit input is defined by Albrecht Schmidt as *actions and behaviour of humans, which are done to achieve a goal and are not primarily regarded as interaction with a computer, but captured, recognised and interpreted by an AAL as input* (Schmidt, 2005).

In AAL health monitoring, explicit interaction is about tasks such as updating medical history of an older person or adding new medical conditions to the system, while implicit interaction consists of vital signs data of an elderly person which are monitored using wearable sensor devices. The ability to appreciate the environment, requires the use of explicit and implicit inputs effectively, within pervasive computing, referred to as being context-aware (Kofod-Petersen, 2007). For the AAL health monitoring is essential to be context-aware as online and ad-

hoc natures of these systems. The contextual information in AAL health monitoring is dynamic and changes more often compared to traditional systems. Therefore, the use of appropriate techniques is helpful for modelling the outside world which leads to reasoning and decision-making capabilities.

### 2.2.3 Context-Awareness

Context-awareness computing is about AAL systems in health monitoring automatically perceiving contextual information from a person and taking an action according to the person's current context and needs (Mshali *et al.*, 2018). Both *context* and *aware* terms have certain meanings. According to Dey, (2001) context represents *any information that can be used to characterize the situation of an entity. An entity includes person, a place or an objective* (Mshali *et al.*, 2018), whereas aware is defined in (Kofod-Petersen, 2007), as *having knowledge or reception of a situation or fact*. In health monitoring, an entity may be an older or disabled person who uses sensor devices for monitoring his/her vital signs.

In AAL generally, there are no standards or guidelines available for building a context-ware system as these systems are usually developed for a specific scenario and purpose, and thus, are focused on finding solutions to a specific problem (Satyanarayanan, 2001). Therefore, how an AAL's knowledge or understanding of a situation is defined is largely subjective (Kofod-Petersen, 2007). For example, to monitor the health status of an elderly person and to ensure that he/she is in good health, the information that should be needed is about the elderly's persons vital signs which may represent Heart Rate (HR), and Blood Pressure (BP) (Patel *et al.*, 2012). In healthcare, readings from these vital signs are used by health professionals to diagnose medical conditions such as hypertension, hypotension or to detect any abnormal readings that can lead to a critical event such as cardiac arrest. In AAL health monitoring, the inputs from sensor devices are referred to as *raw data* or *low level* context which are unprocessed (Mshali *et al.*, 2018). The low-level context does not hold enough information about an entity, hence it needs to be processed before being used by ML algorithms for reasoning and decision making. The processed data is commonly referred to as *context information*. Different models and techniques are used for modelling the

contextual information to make a better understanding of an entity's vital signs patterns. They will be discussed in more detail in Section 2.4. In addition to this, some explicit input from health professionals can be used through the user interface. The explicit information in AAL health monitoring can be the medical history of an older person including his age, gender, or history of any stroke or heart condition (Schmidt, 2005). In health monitoring, in addition to the explicit information, the correlative changes in vital signs data of elderly persons should also be monitored and used with the elderly person's current medical history for diagnosis and prognosis of a symptom.

For instance, Systolic BP over 140 mmHg is a normal rate for a person with a hypertension condition where the same Systolic BP rate is an indicator of an abnormality in the vital sign of a healthy person. The vital signs data may also have a different contextual meaning depending on the time of day that it occurred. For example, the blood pressure normally is higher when a person wakes up in the morning and during the night is descending due to the body's normal circadian rhythm. These changes in temporal patterns of the vital signs in AAL health monitoring are primary for decision making, hence they are modelled and learned to ensure the effectiveness of the system.

A good example of a context-aware AAL system and residential monitoring network is the AlarmNet (Wood *et al.*, 2008) that is developed by the University of Rochester. Various wearable and embedded sensor devices in the Alarm-Net are ubiquitously networked to collect 24 hours of data every day to help the health professionals in monitoring the health condition of residents. One requirement of the Alarm-Net is that power management of every sensor should be adaptively controlled according to its characteristics and context such as the location of sensors. To overcome this requirement, the AlarmNet has combined the Context-Aware Power Management (CAMP) subsystem that is aware of the activity of residents while it ensures the system dynamically provides a reliable sensing service with efficient energy consumption. As shown in Figure 2-4, the CAMP subsystem has five functional components, including sensor drivers, a context-aware power manager, a Circadian Activity Rhythm (CAR) analysis program, a context manager, and an UI.

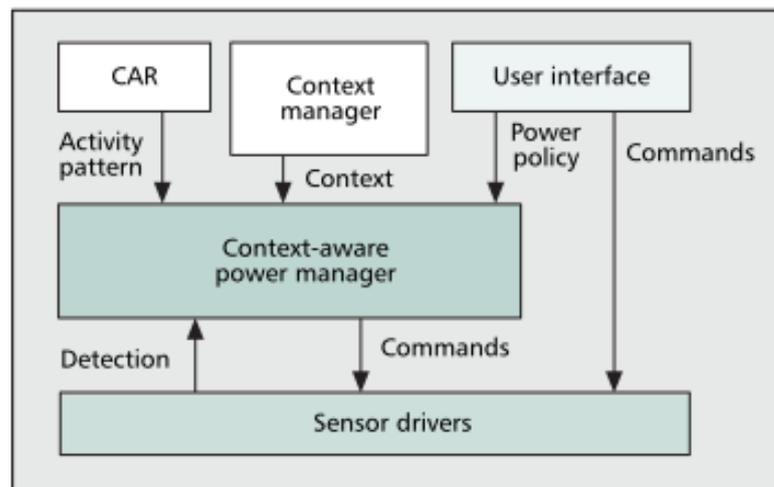


Figure 2-4 Context-Aware Power Management subsystem for AlarmNet (Adopted from (Wood *et al.*, 2008) )

Administrators in Alarm-Net directly control each sensor through the UI (explicit interaction) and they can turn the sensors on/off. Moreover, administrators can use the system to specify context policies for the power management such as "to turn off the sensors in the bedrooms and reducing the temperature sensing rate to an hourly mode in all rooms if the resident is in the living room watching television".

On the other hand, the context manager collects and maintains the current context of residents and the environment from sensor data (implicit interaction). For example, sensors such as Wlan, GSM, compass or air pressure are used in different places to track the location of the resident (Salih and Abraham, 2013). The perceived data from sensors will be used with context policies in order for Alarm-Net to take an action.

Alarm-Net also incorporates the CAR, an artificial intelligence software that learns the patterns of the daily life of the individuals to control the system and network protocols for power management and privacy. The CAR computes reasoning and decision making. It enables advanced power management by anticipating the sensors that should be kept active while temporarily disabling other sensors to conserve power according to the habits of the individual.

For the Alarm-Net to effectively manage the power consumption of each sensor, it needs to be knowledgeable about its surrounding environment. Being knowledgeable is central to being aware, reasoning capabilities or intelligence is applied using the CAR algorithm. Hence, the context-aware paradigm in AAL scenarios plays an important role to bridge the sensing technologies to the reasoning and the decision-making tasks using artificial intelligence software.

#### 2.2.4 Artificial Intelligence

Artificial intelligence (AI) is one of the oldest fields of computer science that strives to understand the essence of intelligence to compose a new intelligent machine that responds in a manner similar to human intelligence. In recent years, AI techniques and methods have been a key factor in the success of AAL, specifically the health monitoring applications. As discussed beforehand, for the AAL to behave intelligently, and to make decisions, there is a necessity for contextual information and intelligence factors. Hence, the AI has resulted in a transformation from a conventional system with limited or no intelligence to the modern systems with a capability of making the right decisions.

The field of AI research was born as an academic discipline in 1956 at the Dartmouth college summer AI conference. This conference was organized by John McCarthy, Marvin Minsky, Nathan Rochester and Claude Shannon. McCarthy in this conference was first to coin the term Artificial Intelligence and defined AI as “The science and engineering of making intelligent machines” (McCarthy, 1998).

Much of the early works in the AI field focused on formal tasks such as: Turing’s test in 1950 developed by Alan Turing, a game of Chess written in 1951 by Dietrich Gunther Prinz a computer science pioneer, the checkers-playing program by Arthur Lee Samuel, an American pioneer in the field of computer gaming and AI in middle 50s and early 60s, and the Logic Theorist created in 1955 by Allen Newell, Herbert Alexander Simon and John Clifford Shaw. The Logic Theorist was among the first attempts to prove mathematical theorems.

A key figure in AI development is Alan Turing, a British mathematician who used computers to break German codes during World War II. After the war, Turing became interested in the area of computing machinery and intelligence. He proposed Turing's test where an interrogator asks a computer terminal questions about a subject for which he or she seeks answers. The interrogator does not know whether the system is linked to a human or if the responses are in fact, generated by a program that is acting intelligently. Hence, when the interrogator cannot tell the difference between the human operator and the computer system, then the system is said to have passed Turing's test and has exhibited intelligent behaviour. This test was advised to determine if a computer has real intelligence or not.

Moreover, a different work on AI was formed in 1959 by Newell, Shaw, and Simon that inspired by a kind of problem-solving that humans do every day, usually called common-sense reasoning. They have built the General Problem Solver (GPS) program as a universal problem solver which did well in solving simple tasks such as in the game of 'Towers of Hanoi'. Nevertheless, GPS had a shortcoming of using a considerable amount of knowledge about a particular problem domain. Consequently, this resulted in the GPS only performing well with small tasks.

It was an assumption among researchers at that time, that computers can perform well at any tasks simply by being fast at searching through many solution paths and subsequently picking the best one. However, this assumption that computers need a little information to perform well has turned out to be wrong since no computer was adequately fast to beat the combinatorial explosion generated by most problems.

Since the birth of AI, theory, techniques and technologies have become more and more mature for handling larger amounts of information. Hence this results in a growth of AI research that now covers various subfields, as shown in Figure 2-5, including robotics, vision and image recognition, natural language understanding, games, machine learning, and expert systems applied in

specialised domains (e.g. medical diagnosis, financial analysis and engineering fault finding).

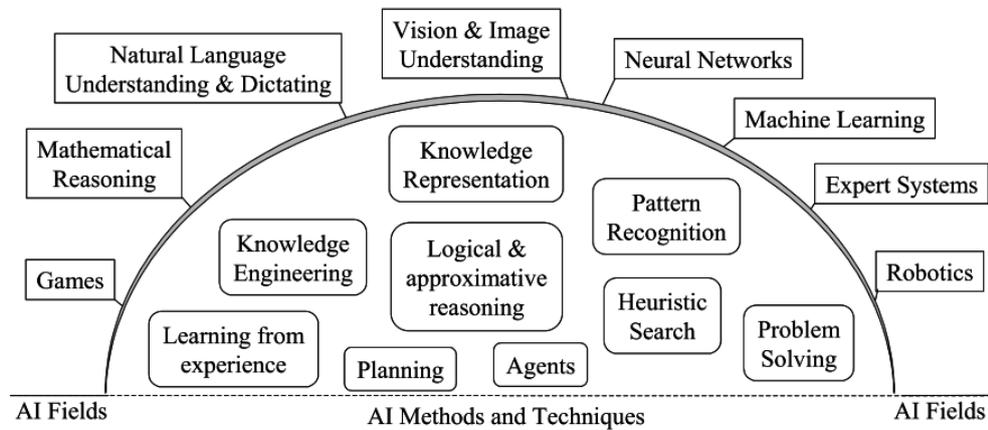


Figure 2-5 AI Fields, Methods and Techniques (Rech and Althoff, 2004)

Machine Learning (ML) as a subfield of AI has received attention from the AI community from the beginning. From a software engineering perspective, ML is an intelligent algorithm software that can be deployed in AmI to enable the learning for reasoning and decision making.

In recent years, emerging technologies such as the AAL system applying ML algorithms have provided relatively inexpensive means to collect the data from surroundings. Therefore, extensive amounts of data are stored in databases which require specialised tools and data processing techniques for data mining (Lavrač, 1999).

As a result, traditional data analysis tools have become inadequate, hence, to satisfy this need, AAL systems considered the use of a new technique developed in the interdisciplinary field of Knowledge Discovery in Database (KDD). The KDD typically has the following steps: understanding the domain, forming the dataset and cleaning the data, extracting regularities hidden in the data and formulating knowledge in the form of patterns or rules (Data Mining), post-processing of discovered knowledge, and exploiting the result.

Data mining forms one of the phases of knowledge discovery where ML algorithms are widely applied as an intelligent data mining technique for data modelling. Traditional systems are capable of functioning based on user's

commands. However, they are not intelligent enough to avoid doing things that the user does not wish for. At the same time, the systems in particular AAL health monitoring applications that are using the ML algorithms capable of learning and modelling of user's behaviour by observing users (explicit interaction and implicit interaction), thus making these systems more acceptable to them (Ramos, Augusto and Shapiro, 2008).

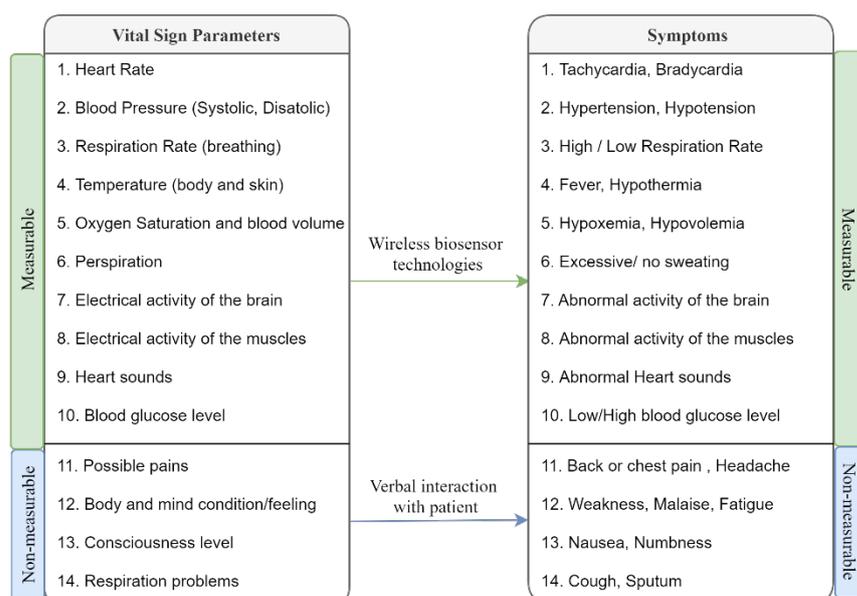
According to Lavrač (1999), ML algorithms, as well as data visualisation, and temporal data abstraction, are techniques of intelligent data mining or methods. Data mining technique is also defined as extracting regularities hidden in the data and formulating knowledge in the form of patterns or rules. Current trends in AAL systems in particular health monitoring show a need for formal reasoning, as well as intelligent data analysis techniques for the extraction of knowledge, regularities, trends and representative cases from patient data stored in the medical record.

ML algorithms can look at patterns and learn from them to adapt behaviour for future incidents, while data mining is typically used as an information source for machine learning to pull from. Although data scientists can set up data mining to automatically look for specific types of data and parameters, it doesn't learn and apply knowledge on its own without human interaction. Data mining also can't automatically see the relationship between existing pieces of data with the same depth that machine learning can.

### **2.3 Requirements for Vital Signs Monitoring in AAL**

In a clinical environment, healthcare professionals use symptoms and vital signs as clues to determine the most likely diagnosis when an illness occurs. Table 2-1 shows the measurable and non-measurable vital signs that are commonly monitored by healthcare professionals for symptom diagnosis and prognosis.

Table 2-1 Common vital signs and symptoms



The temporal pattern of the vital signs has been explored in several studies (Hands *et al.*, 2013; Kyriacos *et al.*, 2014; Mike and Bunker, 2014) to assess the relative importance of the vital signs in diagnosis and prognosis. For instance, the National Early Warning Scores<sup>4</sup> (NEWS) and Cardiac Arrest Risk Triage (CART) scores are two diagnosis systems that are used in general wards and critical care units where their applicability has been well accepted. The NEWS system was launched in 2012 and developed by the NHS in response to a number of national enquiries into patient deaths in hospitals. The NEWS uses six vital sign parameters normally measured in hospitals including heart rate, systolic blood pressure, respiratory rate, temperature and level of consciousness. The NEWS score based on these vital signs assists health professionals to accurately measure a symptom, hence it can be used as a basis for triaging the response to patients. In 2017, an updated version of NEWS, NEWS2, was launched particularly to identify patients at health risk due to sepsis (Royal College of Physicians, 2019). On the other hand, the CART system outlined in (Churpek *et al.*, 2013) predicts the cardiac arrest event by collecting vital signs such as heart rate, diastolic blood pressure, respiratory rate and age.

<sup>4</sup> <https://www.england.nhs.uk/ourwork/clinical-policy/sepsis/nationalearlywarningscore/>

A similar approach in terms of abstracting temporal features of the vital signs is applied in AAL health monitoring for diagnosis and prognosis where continuous and real-time monitoring of patients' vital signs are generated by using a set of biosensors.

Figure 2-6 shows three unique requirements for vital signs monitoring which are generally utilized in AAL health monitoring scenarios. Data mining tasks for health monitoring can be recognized in connection to three views (Banaee, Ahmed and Loutfi, 2013). The first view involves AAL health monitoring scenarios where ML algorithms for anomaly detection or prediction are particularly used to remotely monitor the health condition of the patient, whereas the health application in clinical scenario mainly is applied for symptoms diagnosis. The second view involves scenarios for patients with known medical records, both diagnosis and explicitly the likelihood to raise alerts (alarm) are key undertakings. For health monitoring which commonly incorporates sound people who want to guarantee the upkeep of good wellbeing, prediction and anomaly detection are utilized. The last view outlined considers the three fundamental data mining tasks in connection with how the information is handled. For each one of the three undertakings, information has been attended to both in an online and offline fashion, with more alert-related errands being normally utilized in the context of online and continuous observing.

For an AAL system to detect anomalies in vital signs, it requires certain qualities that enable it to cope with the nature of the data in this domain.

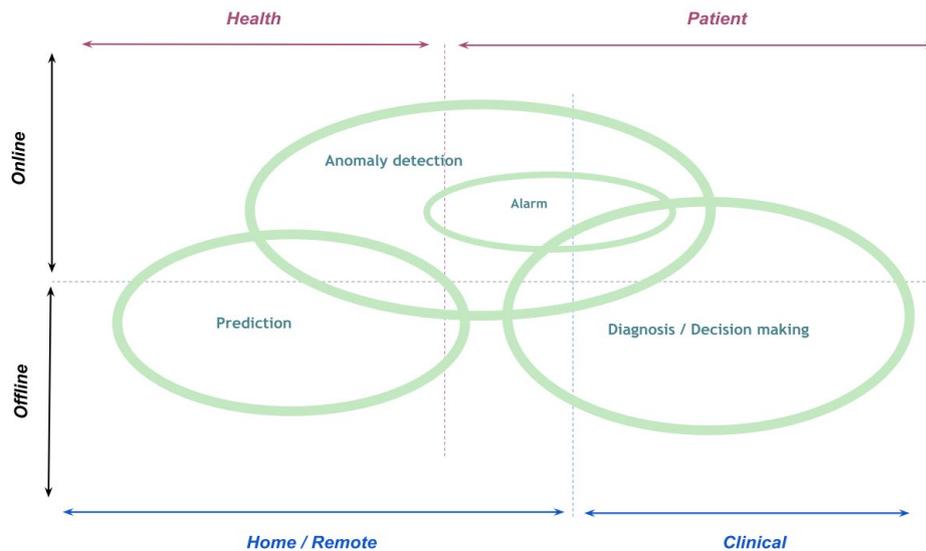


Figure 2-6 Data mining tasks including anomaly detection, prediction and clinical diagnosis (Banaee, Ahmed and Loutfi, 2013)

In AAL health monitoring, vital signs data collected from the patients might contain missing values or might have high frequency noise which require pre-processing of biosensor data. Moreover, vital signs data are generated as a time series of raw data and are affected by several factors over time. These changes are represented as different patterns in raw data which require further processing including data segmentation and transformation (discussed in Section 2.4).

Another requirement of a health monitoring application to detect anomalies in vital signs is to model the temporal patterns, hence temporal patterns of vital signs must undergo a reasoning process (discussed in Section 2.5).

Several studies suggest that anomaly detection algorithms investigated in research or deployed in various industries are typically driven by domain knowledge from expert, data types and scenarios, e.g. real-time, conventional batch or post processing ML methods. An important characteristic of anomaly detection in health monitoring applications is to be capable of adapting to new changes in an online fashion. Moreover, the health monitoring application should be capable of detecting temporal anomalies as early as possible, as late detection of anomalies in the health monitoring scenario can be catastrophic. The use of ML techniques and algorithms in a health monitoring application to detect anomalies in vital signs is discussed in Section 2.6.

## 2.4 Data Mining for Vital Signs Monitoring

Data Mining (DM) techniques in the context of AAL, in particular, a health monitoring application is becoming popular as there is a need for an efficient analytical methodology for detecting unknown and valuable information in health data including raw data and medical condition of the patient (Rafe and Farhoud, 2013). In modern healthcare monitoring, multiple health parameters of patients in AAL are measured continuously in order to make an accurate detection and prediction of health-related abnormalities. Nevertheless, this has introduced new challenges to this field, as integration, and more importantly interpretation of the raw data while considering the state of a patient (from medical history) is a key factor in the success of AAL systems. DM and ML techniques enable AAL applications in diagnosis, patient and treatment monitoring, health monitoring, surveillance of disease and risk factors, and public health investigation and disease control.

In general, KDD, DM and ML are related terms and are used interchangeably, but many researchers assume that these terms are different. According to Lavrač (1999), DM techniques are defined as extracting regularities hidden in the data and formulating knowledge in the form of patterns or rules using ML algorithms. DM is one of the most important stages of the KDD process and it is typically used as an information source for machine learning to pull from. Whereas, ML is defined as an intelligent data mining technique (Lavrač, 1999; Banaee, Ahmed and Loutfi, 2013; Rafe and Farhoud, 2013; Rashidi and Mihailidis, 2013; Aggarwal, 2014), that can look at patterns and learn from them to adapt the behaviour of AAL for future incidents.

Current trends in AAL systems in particular health monitoring show awareness of the requirement to introduce formal reasoning, as well as intelligent data analysis techniques in the extraction of knowledge, regularities, trends and representative cases from patient data stored in the medical record.

Two definitions of DM are from Fayyad *et al.* (1996) and Giudici (2008), which give us a better insight into the DM process and phases:

- Fayyad definition of DM: “a process of nontrivial extraction of implicit, previously unknown and potentially useful information from the data stored in a database” (Palaniappan and Awang, 2008).
- Giudici definition of DM: “a process of selection, exploration and modelling of large quantities of data to discover regularities or relations that are at first unknown with the aim of obtaining clear and useful results for the owner of database” (Palaniappan and Awang, 2008).

The DM consolidates statistical analysis technique, ML algorithm and database technology to extract meaningful information and relationships from sensor data (Lavrač, 1999) to provide the contextual information for anomaly detection, prediction and decision-making tasks. DM techniques that have been applied to sensor data for AAL health monitoring applications have varied, for example in several works in health monitoring, including Batal *et al.* (2011); Khusainov *et al.* (2013); Ni, Hernando and de la Cruz, (2015); Forkan and Khalil (2017b), segmentation is used as part of the DM processes where in some other works the segmentation is not considered (Banaee, Ahmed and Loutfi, 2013). The segmentation is used in several health monitoring applications in particular in vital signs monitoring of the patients are continuously measured periodically for anomaly detection and prediction (Forkan *et al.*, 2015). More details about segmentation and temporal reasoning techniques in the context of AAL health monitoring are discussed further in Section 2.4.3 and Section 2.5.

According to Giudici *et al.*, the knowledge data mining processing for sensor data is structured in various phases, these phases are *data acquisition*, *data pre-processing*, *data segmentation*, *data transformation*, and *data modelling*. Regardless of the data mining technique used, the most standard and widely used approach to mining information from wearable sensors in the context of health monitoring is given in Figure 2-7, and different phases are discussed following.

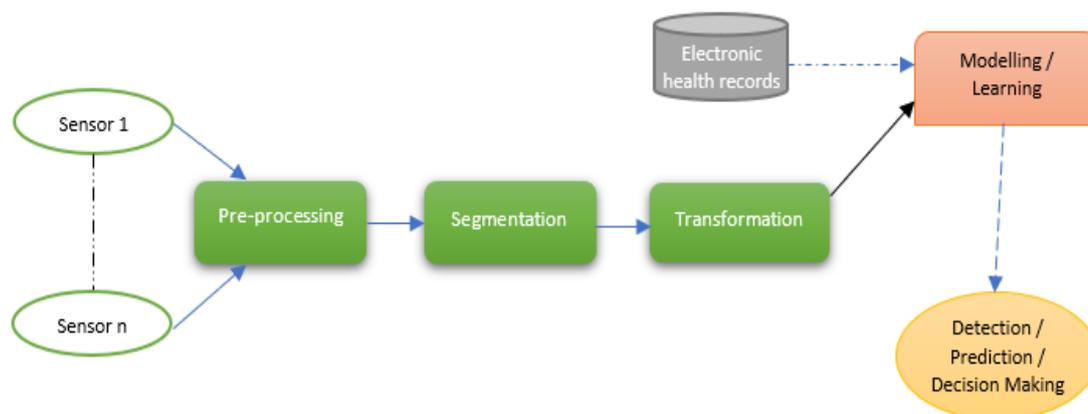


Figure 2-7 A generic architecture of the main data mining approach for raw data (based on Palaniappan and Awang, 2008; Banaee, Ahmed and Loutfi, 2013)

### 2.4.1 Data Acquisition

Data acquisition or data selection in AAL is comprised of processes for collecting data from wearable sensor sources. In AAL health monitoring, multivariant vital signs are usually collected for monitoring the health status of a patient. In the academic literature, three data collection approaches are identified, including experimental wearable sensor data, clinical databases and simulated sensor data (Banaee, Ahmed and Loutfi, 2013). In experimental wearable sensor data scenarios, collected data are obtained based on predefined scenarios. The most common approach in AAL health monitoring for data acquisition is clinical databases from which a wide range of vital sign/physiological data can be accessed. The clinical databases consist of electronic health records (e.g. age, date of birth, gender and medical condition), and temporal measurement of vital signs such as HR, BP and SpO<sub>2</sub>. Some examples of commonly used databases in AAL for anomaly detection and prediction are PhysioNet (Salem *et al.*, 2014), UCI (Sartakhti, Zangoeei and Mozafari, 2012), UCL-THIN database (Blackburn *et al.*, 2018), MIMIC and CPRD-HES Database (Analysis *et al.*, 2017). On the other hand, Simulated sensor data are used for scenarios where there is more focus on the efficiency and robustness of information extraction. Usually, this approach is selected when there is a lack of long-term and large-scale datasets. The clinical databases are investigated and documented in Chapter 5.

### 2.4.2 Data Pre-processing

In the real world, data perceived from sensor devices contain missing values or high-frequency noise due to factors such as sensor errors, disconnections, equipment changes and many other reasons. For instance, data cleaning is used to filter out noisy data features, data interpolation is used to cope with missing values, data normalisation is used to cope with heterogeneous sources, temporal alignment, and data formatting (Banaee, Ahmed and Loutfi, 2013). Examples of techniques that are used in AAL health monitoring to cope with missing vital signs data are such as median-pass, k-nearest neighbour filter and linear interpolation. These techniques are useful to compute a value from available vital signs data to refill one or more missing vital signs data (Forkan and Khalil, 2017a). Furthermore, examples of data cleaning techniques are such as median, Butterworth low-pass, discrete wavelet and Kalman filters (Khusainov *et al.*, 2013).

### 2.4.3 Segmentation

In AAL health monitoring, the vital signs data from patients are continuously measured periodically. In a AAL health monitoring scenario, the vital signs data are generated as a time series of raw data, hence the perceived data are affected by several factors over time, which leads to the representation of different patterns in raw data that are of interest to health professionals. For instance, these patterns can be monitored by health professionals for diagnosis, prognosis and treatment of disease and critical conditions. The segmentation problem is defined by Kusainov *et al.* (2013), as “*given a time series,  $T$  (a finite set of samples characterised by time points), partition  $T$  into segments (windows) of  $t$  consecutive samples between two points  $\mathbf{a}$  and  $\mathbf{b}$  that are internally homogeneous with respect to the application*”. The segmentation task is an important part of the data mining process particularly in health monitoring, where it is used commonly in different temporal reasoning problems. Temporal reasoning paradigm and approaches are detailed in Section 2.5.

The literature (Ni, Hernando and de la Cruz, 2015), (Forkan and Khalil, 2017b), (Khusainov *et al.*, 2013), (Avci *et al.*, 2010) highlighted different Segmentation techniques including: Fixed-size Non-overlapping Sliding Window (FNSW), Fixed-size Overlapping Sliding Window (FOSW), Top-Down (ToD), Bottom-UP (BUP), Sliding Window and Bottom-up (SWAB) and Variable-size Sliding Window (VSW). These segmentation techniques are different based on their selected method of segmenting raw data into smaller blocks. In the literature, the blocks of raw data are referred to as *windows* or *intervals*. FNSW is known as a simple but popular technique for AAL health monitoring systems.

In FNSW the exact size of the window is calculated, and the size of windows is fixed in this approach. One of the deficiencies of the FNSW is that the window sizes can be too short so that they do not cover the duration of a state, or they can be too long so that they may overlap two different unrelated states of temporal data. FOSW is proposed as an improved version of FNSW which resolves some of the FNSW limitations. In FNSW, overlapping of windows can be adapted by giving different overlapping percentages. Value of overlap may vary from 0% to 100%. ToD and BUP are other types of segmentation techniques in which raw data can be segmented into multiple windows, however, they slice raw data in different ways (Avci *et al.*, 2010).

For instance, Forkan and Khalil (2017), has applied FNSW where 24 hours of continuous raw data is segmented into three windows including observation, lead and prediction windows as shown in Figure 2-8. The observation time  $t_o$ , lead or forecast time  $t_l$  and prediction time  $t_p$ . The ML models will use this representation of temporal vital sign data for anomaly detection and/or prediction of health-critical events.

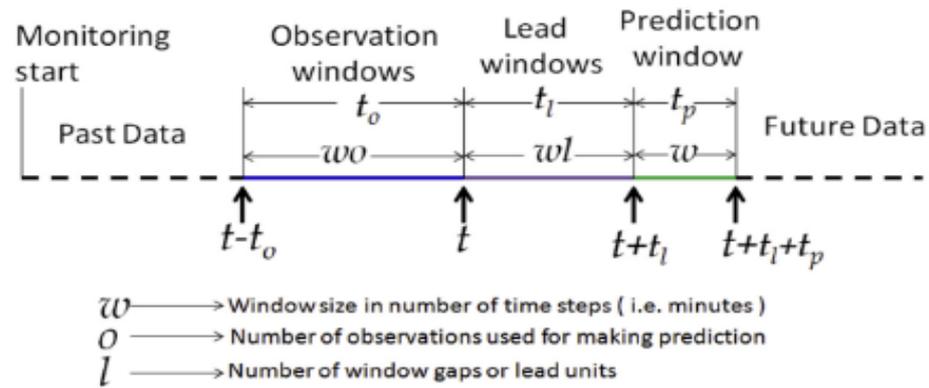


Figure 2-8 Observation, Lead, and Prediction windows (Forkan and Khalil, 2017a)

### 2.4.4 Data Transformation

In literature, the transformation often called feature extraction and selection (Aggarwal, 2014). Feature extraction task refers to the methods of finding a subset of very representative features from raw data which contribute to the performance of ML algorithms. Feature extraction approaches are categorised into three types including: *time domain*, *frequency domain* and *discrete domain*. Table 2-2, summarizes the most commonly used features in each of these three domains. For instance, Mean and Median can be used for temporal raw data to represent the magnitude of each vital sign, or Standard Deviation (STD) can be used to describe the probability distribution of vital sign data (Khusainov *et al.*, 2013).

Table 2-2 Taxonomy of extracted features in three domains (adapted from (Ni, Hernando and de la Cruz, 2015) )

Domain	Extracted Features
Time domain	Mean, Median, Average, Variance, Standard Deviation, Minimum, Maximum, Range, Root Mean, Square (RMS), Correlation, Cross-Correlation, Zero-Correlation, Integration, Differences, Velocity, Signal Magnitude area (SMA), Signal vector magnitude (SVM), Difference, Zero-crossing.
Frequency domain	Wavelet Transformation, Fourier Transform (DC component, Key Coefficients, Coefficients sum, Dominant frequency, Spectrum Energy, Spectrum Entropy, Spectrum centroid)
Discrete domain	Euclidean-based Distances, Dynamic Time Warping, Levenshtein Edit Distance

Once data extraction is completed, associated features are to be selected. The role of data selection is to find the most relevant subset of features that discriminate best between events and classes. The feature selection plays an important role in improving the accuracy of ML algorithms. For instance, classification algorithms including Naïve Bayes, Hidden Markov Model (HMM) have a strong relationship with selected feature (Khusainov *et al.*, 2013). An example of feature selection methods that are used in the literature are like Linear Discriminant Analysis (LDA), Correlation-based Feature Selection (CFS), Independent Component Analysis (ICA), and Principal Component Analysis (PCA). For instance, PCA is used for dimensionality reduction in the dataset. The output of PCA is a set of features that are known as principal component (Forkan and Khalil, 2017b).

#### 2.4.5 Data Modelling

This phase is also referred to as mining (Aggarwal, 2014) and intelligent data analysis (Lavrač, 1999), (Augusto, 2005). In this phase, Machine Learning (ML) algorithms are applied to automate the decision making by processing deeper knowledge representation of raw data. For AAL, in particular a health monitoring application, the intelligent data analysis of raw data is used by ML algorithms for carrying out two main tasks including: prediction and anomaly detection. The ML algorithms must be selected base on the characteristics of context information of the domain (Forkan *et al.*, 2015). The ML techniques and algorithms particularly those that are used in AAL health monitoring, are discussed in Section 2.6.

#### 2.4.6 Evaluation

In the literature, different evaluation techniques are used to evaluate and analyse the performance of the ML algorithms. Three commonly used techniques are *Precision, Recall, F-measure* (F-score), Hammering Loss and Hammering score (Forkan and Khalil, 2017a). According to (Nehmer *et al.*, 2006) and (Haque, Rahman and Aziz, 2015), anomaly detection models that are applied in AAL health monitoring systems should have high recall rate in detecting every real emergency immediately and have high precision rate, in order to reject false emergency detections and alerts. The output of ML algorithms is labelled as positive ( $p$ ) or negative ( $n$ ), and there can be four possible outcomes when the

outcomes of the algorithms are compared to the actual value. The algorithm's outcome when is  $p$  and the actual value is also  $p$ , then it is interpreted as True Positive (TP), whereas if the actual value is  $n$  then it is False Positive (FP). On the other hand, True Negative (TN) is achieved when both the outcome of algorithm and the actual value are  $n$ , and False Negative (FN) is given when the algorithm outcome is  $n$  and the actual value is  $p$ . These notations then can be used to measure the precision, recall and f-measure rate of the algorithm as are defined as follows:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Representation tools such as Receiver Operating Characteristic (ROC) is commonly used in the literature to represent the performance of the ML algorithm on (1), (2), and (3).

The emphasised techniques are almost sufficient to evaluate the performance of the ML algorithms when they are used in a traditional/batched process where the time aspect is not considered in the evaluation process. However, much of the data that are used in AAL for health monitoring scenarios are time series based and measured in an online/real time scenario. In a health monitoring application, it is crucial to detect anomalies in vital signs as early as possible in order to prevent any critical events. For this purpose, Numenta.org (Lavin and Ahmad, 2015) has developed the Numenta Benchmark (NAB) as an open source tool which offers a different technique to evaluate the performance of the algorithms on streaming data an online scenarios. According to (Lavin and Ahmad, 2015), the ultimate goals of real-world anomaly detections must be as follows:

- (a) Detects anomalies in streaming data in on-line manner
- (b) Anomalies should be detected as soon as possible
- (c) False alarms should be refused

The NAB to achieve these ultimate goals, it relies on three components including *anomaly windows*, the *scoring function* and *application profiles*. The NAB evaluates the early detection of anomalies by defining the anomaly windows represented around every anomaly point which is the actual values. The location of the anomaly window must be selected base on the contextual information. As illustrated in Figure 2-9, three anomaly windows (red shaded) are represented around anomalies (marked by red dots). A probationary interval that is shaded in purple is also used which shows the time that algorithms under the test are allowed to learn the data patterns without being evaluated.

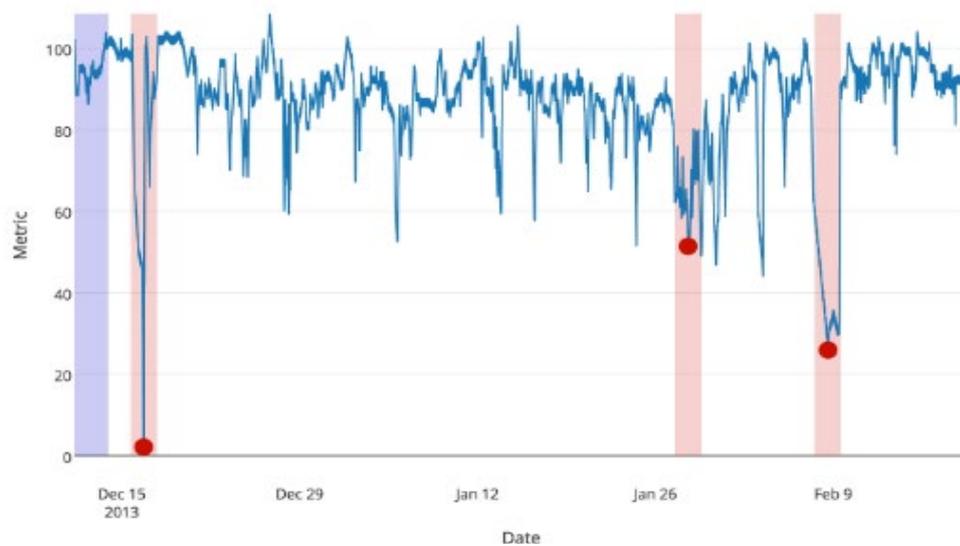


Figure 2-9 An example of anomaly windows in stream data (Lavin and Ahmad, 2015)

In order to define the length of anomaly windows, it is recommended to be 10% of the whole length of the data file divided by the number of anomalies, and they are usually labelled manually in data files. The next component in the NAB, the scoring function uses the anomaly windows to evaluate the performance of the algorithm under the test. Figure 2-10, illustrates an example of multiple anomalies are comprised of red crosses in the scoring function mechanism, which are outside the window, whereas the anomaly points inside the anomaly window are represented by green crosses. The scoring function uses the window to identify and weight TP(s), FP(s), and FN(s). The first anomaly point in Figure 2-10, is given a negative weight (FN) by NAB as it is not detected correctly by the algorithm under test. The second anomaly has been given highest positive scores as it is detected as earliest a point as possible in the window whereas the second

anomaly within the window has been ignored as the NAB is interested in the anomalies that are detected as early as possible. The fourth and fifth anomalies have been given the negative scores as they are detected outside the window.

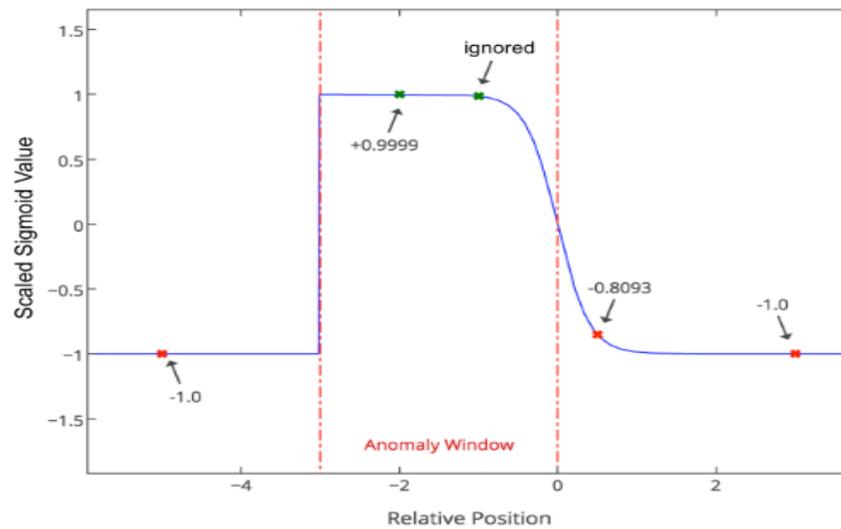


Figure 2-10 NAB scoring function (Lavin and Ahmad, 2015)

The initial idea of the next component, the application profile is to customise the scoring methodology in order to make the NAB applicable to the different application's scenarios. As highlighted beforehand, the performance of the application can be evaluated by adjusting the weight to the false positives and/or false negatives scores. Hence, three application profiles are proposed by the NAB to apply different weights to the detected anomalies that are associated with three different scenarios (also called profile in NAB framework). These three profiles are highlighted as:

- **Standard Profile:** in this setting, equal weights are appointed for TP(s), FP(s) and FN(s)
- **Reward low false positives profile:** it assigns more weight and penalties for FP(s)
- **Reward low false negative profile:** it gives more weight and penalties for missed anomalies (FN(s))

## 2.5 Temporal Reasoning for Vital Signs Monitoring

The problem of representing temporal knowledge and temporal reasoning has a vital role to play in many areas, specifically from areas of AI, Medical Informatic (MI), computer science, linguistics, and psychology. The accurate detection and prediction of health-related abnormalities in health monitoring applications rely on temporal patterns of the monitored vital signs beside the medical history of the patient (Forkan and Khalil, 2017b).

In AAL health monitoring systems, the vital signs data from patients are generated as a sequence (time series) and collected continuously. The vital parameters are correlated in time and space and have a temporal dependency. Such a temporal and correlation attributes of vital signs like BP, HR, SpO<sub>2</sub> and ECG measurements (numeric) are used in AAL health monitoring scenarios for decision support in diagnosis, prognosis and treatment (Augusto, 2005). For instance, vital signs such as BP can be measured to diagnose a medical condition of patients (e.g. hypertension, hypotension). On the other hand, multivariant vital signs in prognosis scenario can be used to predict a critical event such as a stroke or cardiac arrest. Another example of mining temporal property of data can be used in activity recognition applications/scenarios where punctual occurrences of *preparing drinks, sleeping, walking* are recorded for estimating hazardous situations such as falling.

A key step for data mining in AAL in particular health monitoring scenario is to define a language that can adequately represent the temporal dimension of data and temporal correlations among multiple time series data. According to (Avci *et al.*, 2010), Temporal Reasoning (TR) comprises of formalizing the notion of time and providing means to represent and reason about the temporal aspect of knowledge.

The suggested formalisms on the nature and representation of time is full of disputes and contradictory theories, hence there are many perspectives and options to consider when trying to define what the basic characteristics of a temporal system should be (Pani and Bhattacharjee, 2001).

The literature has considered temporal knowledge as part of the program to characterise the notion of intelligence. According to (Pani and Bhattacharjee, 2001), reasoning and problem-solving in most AAL health monitoring applications are logic-based. Three main logic-based approaches for representing temporal knowledge are First-order logic (FOL), Modal Temporal Logic (MTL) and Reified Temporal Logic (RTL) (Pani and Bhattacharjee, 2001). The reason behind the projection of these logics is to find the best approach to represent different aspects of time for analysing temporal discourse (Pani and Bhattacharjee, 2001).

A lot of work has been done to formalise temporal representation and reasoning, some of the most important formalisms of temporal reasoning in AI including:

- The Situation Calculus (SC) developed by McCarthy and Hayes
- Chronos by Bruce
- Time specialist by Kahn and Gorry
- McDermott's temporal logic
- Allen's interval algebra
- Event calculus developed by Kowalsky and Sergot
- Temporal Logic proposed by Shoham

### 2.5.1 Time in Health Monitoring

One key challenge in TR paradigm is to decide what sort of information is subject to change, in other words, what kind of concepts are considered in the theory beyond the time itself. Temporal reasoning because it is involved in problem-solving in a dynamic environment, hence there is a need to represent what properties the objects of the world can have or do have at each meaningful temporal reference. The vital sign, or in general clinical variables are measured at different time moments in numeric or integer formats and are irregularly sampled in time (Batal *et al.*, 2011). The idea of adding a *time-stamp* to the vital signs has been proposed by the MI community as early as the 1970s to recover the information later in a useful context.

The problem of deciding the kind of references should be considered the basic one has been subject to intense debate. These characterize aspects of temporal data such as HR, BP, and SpO<sub>2</sub> of each patient/object acknowledged in the proposed scenarios.

In several academic literatures, two different perspectives to time are also highlighted in order to obtain a temporal description of data and to transform the time series in vital signs to a high-level qualitative form which known as *temporal abstraction* (Batal *et al.*, 2011). These two approaches are known as *instant-based* view of time, and a *period-based* approach. There are varying terminologies which are used by authors but usually, *instants* and *time points* are notations to refer to *punctual occurrences*, *anchored* and *absolute* references. For instance, some symptoms are described as occurring on a particular day such as “the symptoms started on the 30<sup>th</sup> of March”. On the other hand, *periods* and *intervals* are used to refer to *durative temporal*, *unanchored*, and *relative* temporal references, for instance, “He already had high blood pressure for four days”.

For instance, in a work by Batal *et al.* (2011), a novel temporal pattern mining for classifying health data is proposed. In this work in order to present an interval-based approach, the trend and value abstractions on time series of a platelet count of a patient are used. The interval representation of a time series of a platelet count of a patient is useful for detecting medical conditions. For instance, if he/she has low platelet count which can proceed to bloody vomit event.

As shown in Figure 2-11, the trend abstraction represented in solid red lines, shows platelet counts of patients as increase and decrease intervals. On the other hand, value abstraction is represented as the solid green lines, and it is defined as 5 *states* including a value below the 5<sup>th</sup> percentile is very low, a value between 5<sup>th</sup> and 25<sup>th</sup> percentiles are low and so on.

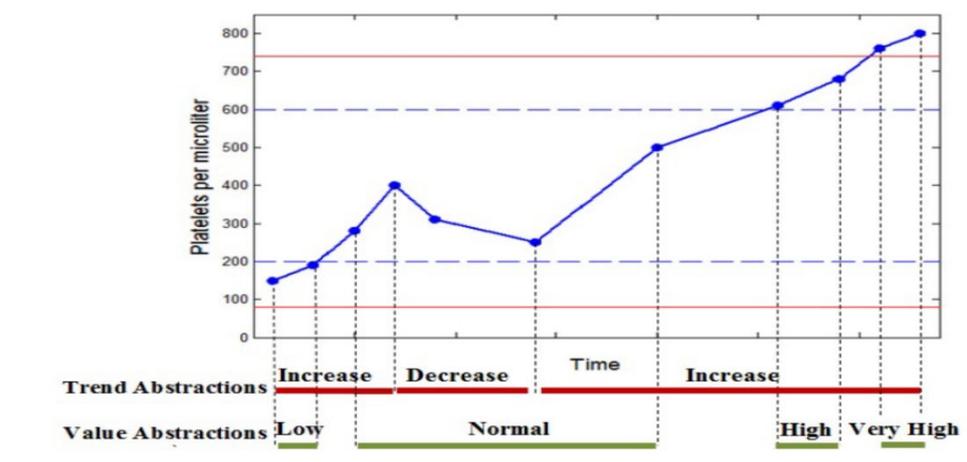


Figure 2-11 An example of Interval-based approach including the trend and value abstractions (Batal *et al.*, 2011)

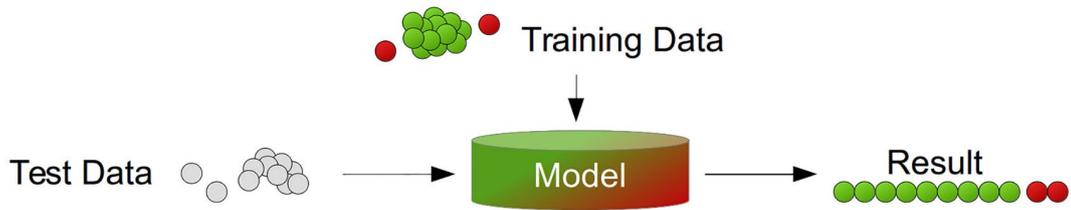
There are several technical terms and notations that usually appear in literature, some of the frequently used terms are defined here, which are also mostly related to the health monitoring scenarios. According to (J.F.Allen, 1984), *events* are defined as a description of an activity that involves a product or outcome, in other words, events are ‘things that happen’ and they usually represent the dynamic aspect of the world (Pani and Bhattacharjee, 2001). On the other hand, a *state* in health monitoring scenario can be defined as medical conditions such a hypertension and hypotension when the BP readings are too high or too low for a period of time. Furthermore, *events* can be defined as a critical event such as stroke, heart attack or cardiac arrest. Grouping events into classes may be useful to prepare the system to react appropriately when critical events occurred (Augusto, 2005).

In another work by Batal *et al.* (2011), *state* of a variable (object) is defined as an abstraction for a specific variable which it can represent a decrease and increase trend in the values of temporal data. To acquire temporal descriptions of the data, basic states are combined using temporal relations to form the temporal patterns. In this work also, a *state interval* is defined as a state of a variable that holds during an interval which has a specific start time and end time. Furthermore, *state sequence* is defined as a series of state intervals, where the state intervals are ordered according to their start times.

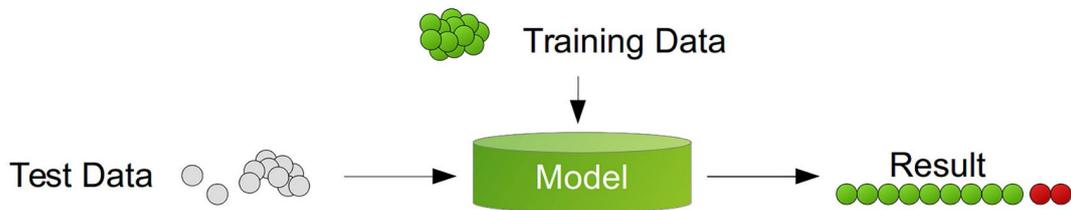
## 2.6 Machine Learning in Health Monitoring

As discussed beforehand, Machine Learning (ML) is a subfield of AI discipline that is used in AAL health monitoring applications for learning, reasoning and decision-making tasks. In literature, ML techniques and algorithms are categorised into different types base on the availability of data labels. As shown in Figure 2-12, the ML algorithms can be categorised into three types:

- a) *Supervised*: Supervised ML techniques require labels to create anomaly or prediction models. The label is not the only term used in literature and other terms such as *notation* and *ground truth* also are used. The dataset for supervised ML is partitioned into learning/training and testing portions, usually 80 percent of the data set is labelled for training purposes for identifying normal and abnormal data points, whereas the other 20 percent of dataset is used for testing.
- b) *Semi-supervised*: the ML technique in semi-supervised scenario use a combination of the supervised and unsupervised learning scheme, typically available training data consists of normal records and the deviation from these normal records is considered as an anomaly.
- c) *Unsupervised*: unsupervised or descriptive ML techniques reflect uncontrolled situations for specifically experimental dataset. In unsupervised learning scenarios, the datasets are not label and no training dataset is required.



a) Supervised Modelling



b) Semi-supervised Modelling



c) Unsupervised Modelling

Figure 2-12 Machine Learning types is based on the availability of the training and testing dataset (adapted from (Goldstein and Uchida, 2016))

In addition to the above classification of the ML models on a basis of availability of labelled data, they are also classified to parametric and non-parametric ML models (Ayadi *et al.*, 2017). The basic idea behind these two classifications of ML models is that learning in parametric models is relayed on predefined map functions and fixed size parameters (e.g. the weight coefficient) which is used for computing probability distribution among data points. Algorithms such as

logistic regression, linear Support Vector Machines (SVM)s are an example of a parametric model.

On the other hand, non-parametric models unlike parametric models, they do not tie to the probability of data distribution. In the following Section, ML algorithms that are used in parametric and non-parametric models will be discussed. Algorithms such as k-Nearest Neighbours (k-NN), Histogram-based Outlier Score (HBOS) and RBF kernel SVM are considered as non-parametric ML algorithms.

### 2.6.1 Prediction

A prediction is an approach that is broadly used in the data mining field to identify events that have not yet occurred. It is getting more attention from the research community for the development of AAL systems to help with prognosis by predicting syndromes, medical conditions, and critical events. For instance, predictive models are used for predicting blood glucose level, further stress levels of patient, medical condition such as hypertension, the survival of cardiac surgery patients (Alotaibi and Sasi, 2015), or predicting lung cancer (Zhao *et al.*, 2015). The predictive models are also referred to as supervised learning models. According to the literature, the suitability of predictive models in AAL health monitoring applications is criticized due to the way that they are handling the temporal patterns of vital signs data, and also, their potential high-risk impact on patient care (Banaee, Ahmed and Loutfi, 2013).

### 2.6.2 Anomaly Detection

Anomaly detection refers to the tasks and techniques of finding unusual patterns or anomalies in data that do not correspond to the expected pattern of the data. The anomaly detection algorithms in AAL health monitoring applications are used to detect anomalies in a single or multivariate vital signs data by learning the patterns of vital signs data. As discussed in Section 2.5, the anomalies of an individual vital sign can be detected by abstracting its temporal patterns. In addition to this, in a scenario where multivariate vital signs are collected for detection of anomalies, then the spatio-temporal patterns of the vital signs must

be considered as the changes in the pattern of one vital sign data can influence temporal data patterns of its correlated vital signs.

In ML, the nature of the anomalies is playing an important role in anomaly detection. According to (Chandola, Banerjee and Kumar, 2009), anomalies are categorised into three types base on the nature of the anomalies:

- **Point anomalies:** when a data point/instance is considered as abnormal from the rest of the data. For instance, in Figure 2-13, data points that are situated in  $N_1$  region and  $N_2$  region are interpreted as normal data points, whereas, the data points  $O_1$ ,  $O_2$  and  $O_3$  that are situated far from the normal regions are considered as point anomalies. An example of the point anomalies in AAL health monitoring scenario can be when a Blood Pressure (BP) of a patient monitored to diagnose the health condition of a patient. If the reading of BP instance is higher than a normal range/threshold, then it can be an indication of the patient has high blood pressure (hypertension).

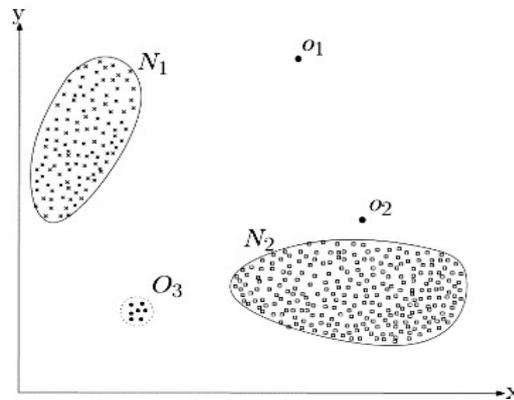


Figure 2-13 An example of points anomalies in a two-dimensional space (Chandola, Banerjee and Kumar, 2009)

- **Contextual anomalies:** when a data point on its own is not considered as anomalous, however, when it is analysed in a specific context then it is regarded as anomalous. For instance, Figure 2-14 illustrates a temporal pattern of monthly temperature. The temperature at data points  $t_1$  and  $t_2$  is same, but  $t_2$  is considered an anomaly as it is occurred in a different season (different context).

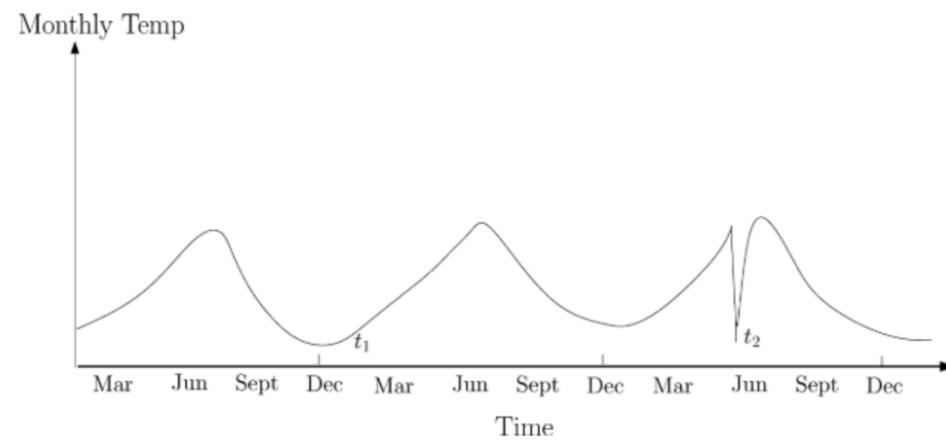


Figure 2-14 An example of contextual anomaly in a monthly temperature data (Chandola, Banerjee and Kumar, 2009)

- **Collective anomalies:** when a collection of data points is anomalous compared to the rest of data. As illustrated in Figure 2-15, the value of electrocardiogram (ECG) is considered anomalous neither on its own nor in a different context, but the repetition of a value in an interval is regarded as anomalous.

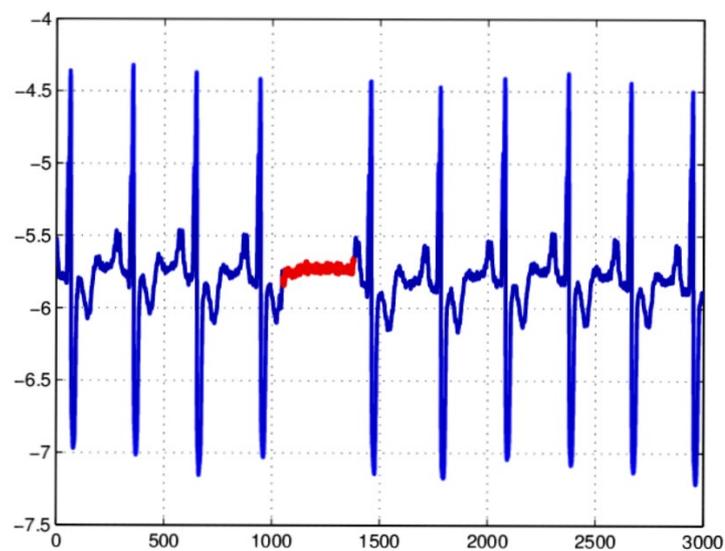


Figure 2-15 An example of collective anomaly in a patient ECG (Chandola, Banerjee and Kumar, 2009)

In literature, the unsupervised anomaly detection algorithms are further split into global and local methods. In local methods, an anomalous data point is referred to a data point that has a value within a normal range for the entire dataset, but if it is compared with values of surrounding points, it is usually high or low. Some

of the popular local algorithms are the Local Outlier Factor (LOF) and Local Correlation Integral algorithm (LOCI). In global methods, the value of a data point deviates significantly from the rest of the data points in the dataset. Examples of global method are such as KNN and Cluster-Based Local Outlier Factor (CBLOF). It is highlighted by Goldstein & Uchida (2016) that local anomaly detection models are not ideal for scenarios where datasets containing global anomalies as they will generate many false positives, however, it seems that global methods perform at least average on local scenarios.

### 2.6.3 Supervised Techniques

Two common types of supervised techniques are classification and regression. The nature of data in input space determines which one of these methods should be used. According to the work by Rafe & Farhoud (2013), Naïve Bayes, SVM and logistic regression are among the algorithms that are suitable for modelling the categorical data where linear regression or regression model using SVM are preferred approach for continuous numerical data.

#### 2.6.3.1 Classification

Classification technique is applied in different applications in the healthcare field. the Classification techniques map data into target classes/groups where the classes are defined based on data attributes and their values. There are two forms of classification technique, binary and multilevel. In binary classification, there are only two possible classes, for instance, two classes can model the health state of the patients such as “diabetic” or “non-diabetic”. On the other hand, the multiclass classification model is suitable for scenarios where datasets consist of more than two classes. For instance, in the health monitoring scenario where three degrees of risk level is labelled as “Low”, “Medium” and “High”. In classification modelling, the datasets are partitioned into training and testing datasets, where the training dataset is used for training the classifier model and testing dataset are used for testing and correcting the classifier. For instance, Bestsimas *et al.* (2008) has used a dataset of 3 years data collected from the insurance companies to evaluate the performance of his proposed "classification tree" approach for predicting the cost of healthcare. The first two years of data in

dataset are used for the training and the last year of data to test the predicted outcome of the classifier. The classification trees can be used for prediction task, for instance, the classification trees are derived from the commonly known ID3 recursive partitioning algorithm to partition the patients into ever smaller classes until creating the classes with all patients corresponding to the same class.

#### 2.6.3.2 Regression

The Regression model is known as a statistical model which mathematically explains the relationships and correlations among the variables. The variables in regression models are commonly distinguished into two types, *dependent variable* usually represented using 'Y', and *independent/explanatory variable* usually represented using 'X'. There is always one dependent variable for one or multiple independent variables. In regression also an ordinary least square method is applied to draw a line of best fit, which is often regarded as *fitting* the model (Harper, 2005).

In literature, the regression models are further divided into two types, linear regression and non-linear regression. Linear regression models are used when target data is numerical data as it is not capable of dealing with categorical data. To overcome this limitation, non-linear regression such as logistic regression can be used where logit transformation function is used to predict forthcoming values. Two types of logistic regression are as Binomial and multinomial. Binomial is used when there are only two status, for example, if a dependent variable has two states, '*healthy*' or '*non-healthy*', while multinomial is used when the dependent variable has more than two status. For instance, a patient either has a medical condition of '*hypertension*', '*hypotension*', '*healthy*' and '*bradycardia*'. Figure 2-16 shows the use of logistic regression to estimate the relative risk for different medical conditions such as Stroke, Thyroid, Diabetes and so on (Gennings, Ellis and Ritter, 2012).

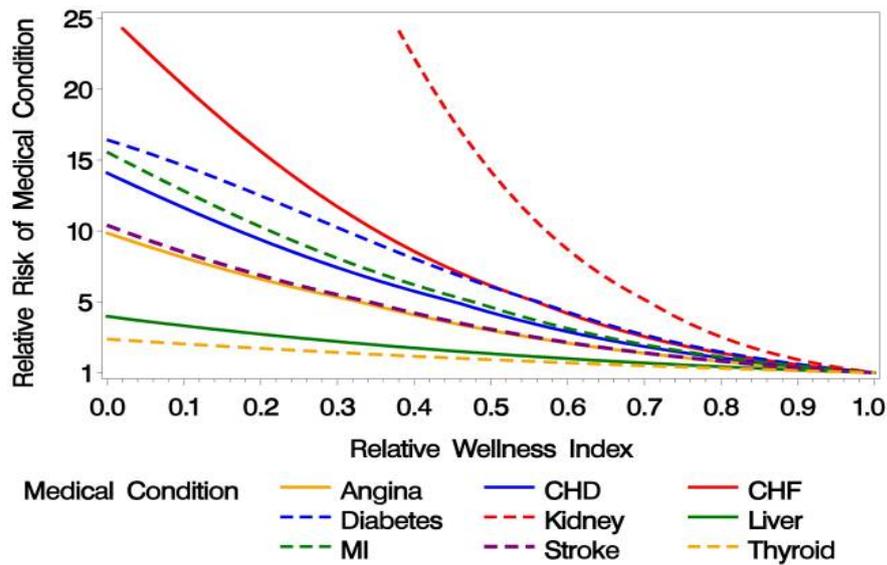


Figure 2-16 An example of Logistic regression for estimation of risk level for various medical conditions (Gennings, Ellis and Ritter, 2012)

## 2.6.4 Supervised Algorithm

In here, some of the common classification algorithms that are used in healthcare fields will be discussed.

### 2.6.4.1 Support Vector Machine

Support Vector Machines (SVM)s are based on statistical learning methods that are used for classification of both linear and non-linear data. SVM forms one or multiple hyperplanes to separate the data points into two classes. The distance between the hyperplane and closest input point (support vector) is optimised by a margin. Figure 2-17 shows hyperplane, margin and support vectors in a linear SVM. According to Kumari & Godara (2011), the classification error of SVM models can be improved by optimising the margin separation of two classes.

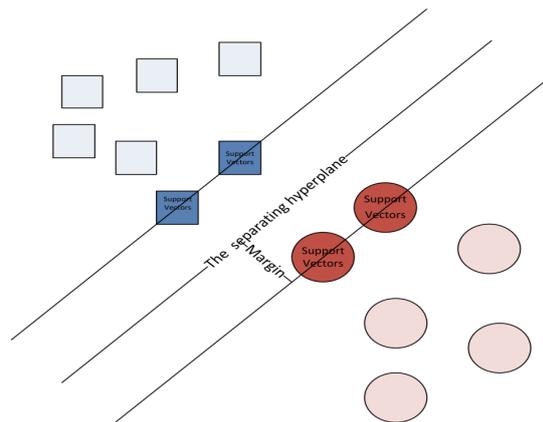
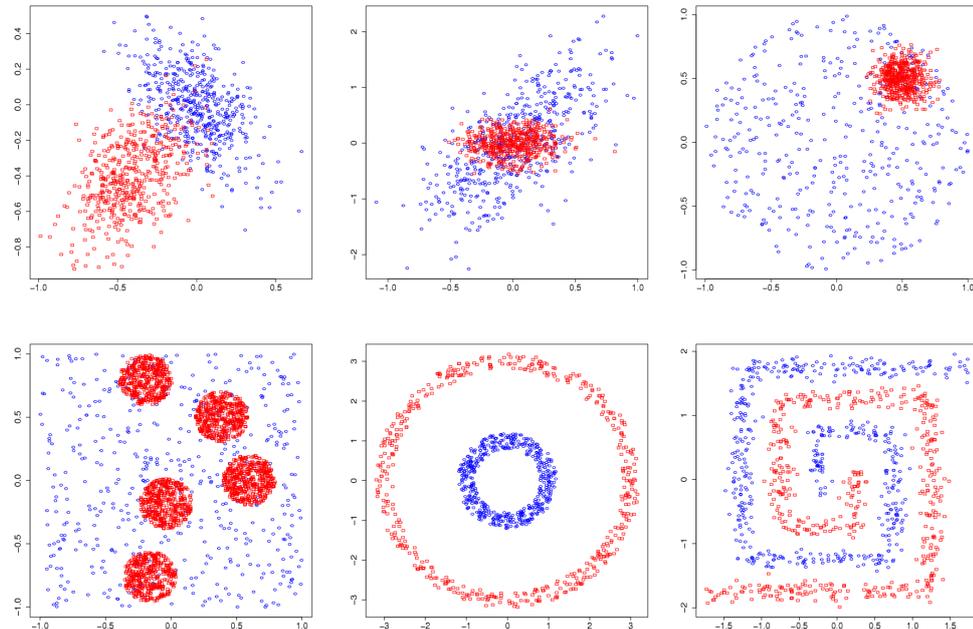


Figure 2-17 Separation of two classes in linear SVM (Amiribesheli, Benmansour and Bouchachia, 2015)

SVM models also use kernel functions (kernel trick) to transform the non-linearly separable data from input space into a high dimension representation where data becomes linearly separable (Amiribesheli, Benmansour and Bouchachia, 2015). Some examples of kernel functions include Radial Basis Function (RBF), Gaussian, polynomial and Sigmoid. For instance, Radial Basis Function (RBF) kernel is used in Kim *et al.* (2016), for non-linear SVM classifier to predict acute hypotensive episodes in Intensive Care Unit (ICU).

For instance, Table 2-3 shows six examples simulated to illustrate the classification performance, 1000 observations are used and classified into two classes. The first four graphs (from left to right, from up to bottom) show mixtures of uniform or Gaussian distribution, and the last two graphs illustrate respectively two concentric circles and two interlocked spirals (Fischer and Mougeot, 2018).

Table 2-3 An Example of representation of input space using kernels (Fischer and Mougeot, 2018)



In work by Raji, Jeyasheeli and Jenitha (2016), the performance of a linear SVM model is compared to J48 and Naïve Bayesian classification performance. In this work, a chronic disease monitoring system for an older person is proposed, where three classification models are trained to classify health condition of patients to *healthy* or *unhealthy*. Several vital signs including temperature, HR, SBP and DBP are collected from 500 patients, and performance of the algorithms under the test are evaluated for the individual vital sign using precision, Recall and F-Measure. As Table 2-4 shows, the SVM and J48 models have given better results.

Table 2-4 Performance comparison of SVM, J34 and Naive Bayesian ( adapted from (Raji, Jeyasheeli and Jenitha, 2016))

Vital Signs	Classification Algorithms	Precision	Recall	F-Measure
Temperature	Naïve Bayesian	0.824	0.750	0.747
	J48	0.922	0.958	0.939
	SVM	0.922	0.958	0.939
Heart Rate	Naïve Bayesian	0.807	0.667	0.628
	J48	0.962	0.958	0.958
	SVM	0.962	0.958	0.958
Systolic Blood Pressure	Naïve Bayesian	1.000	0.857	0.921
	J48	1.000	0.952	0.975
	SVM	1.000	0.952	0.975
Diastolic Blood Pressure	Naïve Bayesian	0.964	0.958	0.958
	J48	1.000	1.000	1.000
	SVM	1.000	1.000	1.000

Different types of SVM model are utilised in the healthcare applications such as hepatitis disease diagnosis using a novel hybrid method based on SVM and simulated annealing (Sartakhti, Zangooui and Mozafari, 2012), automated health alerts for embedded health assessment (Skubic, Guevara and Rantz, 2015), cardiovascular disease prediction (Kumari and Godara, 2011). Different applications of SVM models also are highlighted in a work by Rafe and Farhoud (2013) including:

- Practical Swarm Optimization SVM (PSO-SVM) approach for analysing arrhythmia cordis.
- Hybrid SVM based strategy to build a predictive model for breast cancer diagnosis.
- Genetic SVM classifier for analysing the heart valve disease.

- PSO based SVM model for identifying erythemato-squamous diseases.

#### 2.6.4.2 k-Nearest Neighbour

The k-NN algorithm is known as the simplest machine learning model, it is also referred to as a lazy learner as it is sensitive to noisy, irrelevant data and large-scale dataset.  $k$  in k-NN is a parameter that refers to the number of nearest neighbours (data points), it is used by majority voting function where a new data point is classified by majority votes from its nearest neighbours.

The value of  $k$  has an impact on the accuracy of the algorithm, in the literature the process of selecting the right value for  $k$  parameter is referred to as 'parameter tuning' process. Usually, a larger value is selected for  $k$  to reduce the effect of noise on the classification, however larger values for  $k$  can cause some processing issues as it takes longer for the algorithm to complete a process (Rajkumar and Reena, 2010). Two options are highlighted in literature for selecting the right value for  $k$ , first is usually to use square root of  $n$  where  $n$  is the total number of data points, and second is to use an odd number for value of  $k$  when the total number of data points is an even number by adding one or subtracting one from it.

Usually, the Euclidean distance formula is computed to calculate the distance between two data points where the value of Euclidean has been used to measure similarity or dissimilarity between two data points. A new data point when is supplied into the algorithm, the Euclidean distance formula calculates the distance of inputted data from all data points in the dataset, and then use the  $k$  value to find the nearest data points. In a work by Rajkumar and Reena (2010) applied the k-NN for diagnosis of heart disease. The performance of k-NN is compared with Naïve Bayes and Decision List algorithms using a training dataset consisting of 3000 instances with 14 different attributes. In this work, two evaluation metrics are used including computation speed and accuracy of the algorithm. In both cases, the k-NN did not perform well and it has been outperformed by Naïve Bayes and Decision List algorithms respectively. Different use of the k-NN-based anomaly detection algorithm is proposed by Xie *et al.* (2013) which is based on hyper-grid that has achieved lower computational

complexity compared to original k-NN. More information is discussed in a work by Salem *et al.* (2014).

### 2.6.4.3 Artificial Neural Network

Artificial Neural Network (ANN) is an analytical technique which is inspired by the networks of neurons in the neocortex of mammalian brains. One of the earliest ANNs is known as *perceptron*, invented by Rosenblatt (Rich and Knight, 1991). The ANN is formed by a series of *neurons* or *nodes* that are grouped into layers. The ANN consists of an input layer, one or multiple hidden layers and an output layer. Figure 2-18 illustrates a three layers feedforward neural network that receives a vector of  $n$  inputs and provides a vector with  $m$  outputs. The input layer receives inputs such as age, gender, HR, BP, SpO<sub>2</sub>, and outputs can be used for diagnosis, prognosis and decision-making tasks (Amato *et al.*, 2013). The way that neurons are connected and communicate with one another is an important characteristic of the ANN. Neurons can inhibit other neurons in the same layer also known as *lateral inhibition/competition*, and neurons also can form cycles (feedback) connections and acyclic (feedforward) connections.

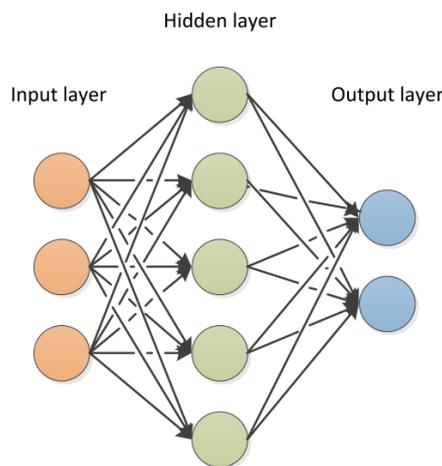


Figure 2-18 An example of Multilayer Feedforward ANN (Amiribesheli, Benmansour and Bouchachia, 2015)

The ANN has been proposed as an appropriate algorithm for a wide range of decision making and anomaly detection in healthcare (Srinivas, Rani and Govrdhan, 2010). According to Amato *et al.* (2013), ANNs have very good performance in detection of Cardiovascular diseases (CVDs) with classification

accuracies higher than 90%. For instance, Uguz *et al.* (2012) used the ANN in a biomedical decision support system to detect the heart valve faults. In this work the record of the heart sound is collected from 120 patients, and Principal Component Analysis (PCA) is also used for dimension reduction of obtained features. The ANN model classifies the heart valve faults into *normal*, *pulmonary* and *mitral* stenosis heart valve diseases with an accuracy of 95%.

Multilayer feed-forward neural networks are another example of ANN model consists of multiple layer of perceptrons/neurons that are most often applied for forecasting and classification applications. In a work by Li and Clifford (2012), Multi-Layer Perceptron (MLP) neural network is applied for assessing the clinical utility of PPG traces which has achieved performance accuracy of higher than 95%. PPG signals are usually used as a cost-effective method for detecting blood volume changes, blood pressure and cardiac output estimation. In this model, several PPG signal features are used as inputs to the MLP, and the results are outputted as two classes of *good* and *bad* quality pulses to observe the quality of the pulses in PPG (Banaee, Ahmed and Loutfi, 2013).

## 2.6.5 Unsupervised Techniques

There are several unsupervised techniques available for scenarios where the datasets are not labelled. In here, two widely used techniques including clustering and association will be discussed.

### 2.6.5.1 Clustering

The clustering techniques are similar to classification except the clusters are not predefined (not labelled), but rather defined by the data points. In clustering techniques, input space is partitioned based on similarities between data points. The data points within a cluster have greater similarity when comparing them to the data points in other clusters. Euclidean distance is one of the formulas that is used for measuring the distance between two data points.

The clustering techniques are grouped into three types: partitioned, hierarchical and density-based clustering. In partitioned clustering techniques, K-means and K-Medoids are two most popular methods that partition the  $n$  number of data

points within input space into ‘ $k$ ’ cluster. One of the challenging tasks in clustering input space, is that the number of clusters (called  $k$ ) must be defined in advance.  $k$  value usually is selected based on assumptions, experimentation and prior knowledge (Hamerly and Elkan, 2004; Liu *et al.*, 2015). For instance, in K-Means method when the number of  $k$  cluster is defined, then the distance between each data points and  $k$  centroids is calculated. Next, each data points will be assigned to the cluster centre whose centre yields the least with cluster mean. The latest mean is calculated and repeated each time to accommodating a newly arrived data points, or until the centres do not change. On the other hand, K-Medoids is very similar to K-Means except that it uses ‘medoids’ instead of ‘mean’ for partitioning the data inputs, where the centre of a cluster is defined as one of the data points (medoids). For instance, in a work by (Liu *et al.*, 2015) a K-Means method is used for monitoring vital signs including breathing and heart rates during sleep to assess the health condition of a patient. In this work, patient status is grouped into two clusters based on two dimensional features including: Power spectral density (PSD) and breathing rate. Figure 2-19, illustrates the k-means method that is used to partition two-dimensional data inputs into two clusters including person 1 and person 2.

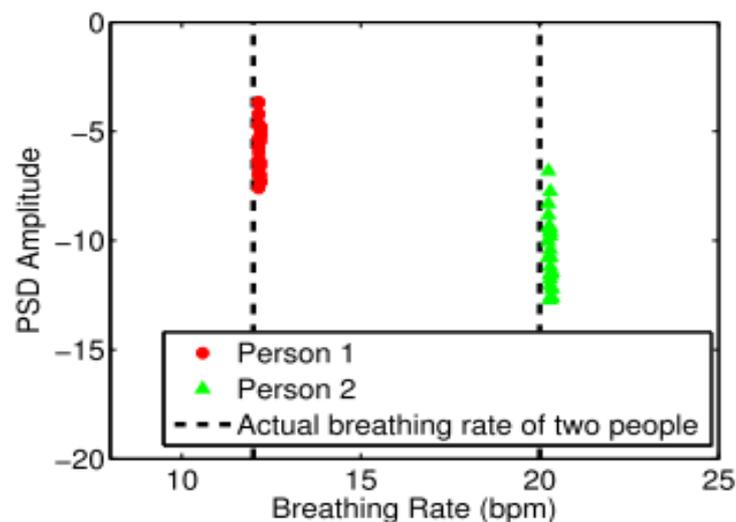


Figure 2-19 K-Means clustering used for patient’s classification (Liu *et al.*, 2015)

Another example of K-Means clustering is used by Rafe & Fahoud (2013) where health condition of a patient is grouped into different clusters (high risk and low risk) base on high blood pressure and cholesterol level.

#### 2.6.5.2 Association

The Association rules model is commonly used in the healthcare field to identify and abstract temporal dependencies of multivariate vital signs by constructing a set of rules to define these temporal relationships.

For instance, a novel application of Intertransaction Association Rule (IAR) model has been proposed by Sebastian, West and Venkatesh (2007) in AAL scenario to detect abnormal behaviour of the elderly patient in order to take an action accordingly. In this approach, the rule sets are constructed to capture associative and, non-sequential correlation of patient's activities while some of the temporal patterns of the occurred events are retained. One of the challenges that is tackled in this work, was the nature of the data in AAL systems in particular HAR as the medical datasets contain a frequent occurrence of a large number of events over relatively short intervals which can result is a large volume of rules.

Another interesting use of association rules model is highlighted by Banaee, Ahmed and Loutfi (2015) to identify the correlation between two pairs of vital signs for nine clinical conditions. The abstracted information is used for diagnosing different clinical conditions, where vital signs data including Heart Rate (HR) with Blood Pressure (BP), and HR with Respiration Rate (RR) are collected from MIMIC database. In this approach, the algorithm looks at impact that temporal changes in HR have on the temporal behaviour of second vital sign (BP or RR). Figure 2-20 illustrates the visual representation of rules for four clinical conditions including: Myocardial Infraction (MI), Coronary Artery Bypass Graft (CABG), Angina and Resp. failure.

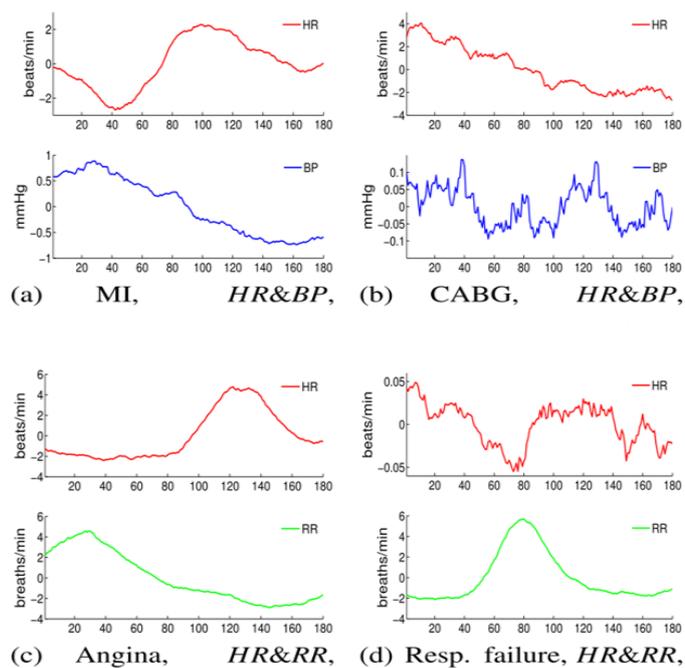


Figure 2-20 An example of rules of four clinical events for the multivariate temporal parameters (adapted from (Banaee, Ahmed and Loutfi, 2015))

Table 2-5 illustrates a selection of outputs for constructed rules in Figure 2-20 and a description of the rules are used to identify different medical conditions.

Table 2-5 An example of textual representation of the constructed rules (adapted from (Banaee, Ahmed and Loutfi, 2015))

Rule	Output text
Rule 1, (a)	In MI condition, most of the time, when heart rate first suddenly increases (5 beats) and then steadily decreases (2 beats), blood pressure steadily reduces (2 units).
Rule 2, (b)	In post-op CABG condition, commonly, if heart rate steadily decreases (8 beats), then blood pressure fluctuates in a very small range.
Rule 3, (c)	In Angina condition, sometimes, when heart rate first sharply rises (7 beats) and then steadily falls (6 beats), respiration rate steadily decreases (9 breaths).
Rule 4, (d)	In Respiratory failure condition, most of the time, after heart rate fluctuates in a very small range, respiration rate first steadily rises (8 breaths) and then steadily falls (7 breaths).

## 2.6.6 Unsupervised Algorithm

In this section, the unsupervised algorithm will be discussed.

### 2.6.6.1 Histogram-Based Outlier Score

Histogram-Based Outlier Score or in other term HBOS is known as statistical, non-parametric algorithms (Goldstein and Uchida, 2016; Ayadi *et al.*, 2017), that is commonly used for anomaly detection task. The assumption in HBOS algorithm is to build a separate univariate histogram for every variable in input space in order to calculate anomalies score. This is done in two modes including, static bin-width or dynamic bin-width. In the static mode, a fixed size bin-width over the input value range is specified, whereas in the dynamic mode, the bin-width can vary. The height of each bin in HBOS represent the density estimation of each data point. Although the nature of input data in HBOS is defined as independent features, hence, several works have recommended the HBOS algorithm for large or high dimension datasets (Goldstein and Dengel, 2012; Goldstein and Uchida, 2016; Ayadi *et al.*, 2017).

For instance, in a work by Goldstein and Dengel (2012) the HBOS performance is compared with 10 other algorithms including k-NN, INFLO, CBLOF. The UCI dataset consists of breast cancer data is also used to evaluate performance of the algorithms with reference to local and global anomalous data points. The HBOS performance was not satisfactory for detecting local anomalies, as HBOS cannot model local anomaly with their density estimation. On the other hand, it had competitive performance compare to rest of the algorithms on global anomaly scenario. The performance of HBOS has outperformed other algorithms with computation speed on large datasets (Goldstein and Dengel, 2012).

### 2.6.6.2 Local Outlier Factor

The Local Outlier Factor (LOF), is one of the first algorithms that has introduced the idea of the local anomalies, and it has been used commonly for local anomaly problems and is based on nearest neighbour technique. A study by Ayadi *et al.* (2017) indicated that LOF algorithm compare to the HBOS is more suitable for

local anomalies detection, because the HBOS computes maximum points of anomalies where LOF computes only the most deviated extreme points as anomalous data.

The LOF compute the local anomalies in three steps:

1. The  $k$ -NN is calculated for each data points.
2. The output (k-NN) from first step is used in Local Reachability Density (LRD) function to calculate the local density of an input data point ( $x$ ) and an object  $o$ .

$$LDR_k(x) = 1 / \left( \frac{\sum_{o \in N_k(x)} d_k(x,o)}{|N_k(x)|} \right) \quad (4)$$

3. As a last point, the LOF score is calculated, by comparing the LRD value of a data point with the LRD value of its  $k$ -nearest neighbours.

$$LOF(x) = \frac{\sum_{o \in N_k(x)} \frac{LRD_k(o)}{LRD_k(x)}}{|N_k(x)|} \quad (5)$$

Putting it differently, the LOF score is the ratio of the local densities, hence, data point is normal when its density is similar to the density of its neighbours, and the anomaly score will be about 1.0. The LOF score for the data point will have larger value based on how different is its density value compared with its neighbours.

However, LOF has a deficiency in handling the data points with varying densities that are close to each other (Rafe and Farhoud, 2013). This shortcoming is fixed by its new arrival, commonly known as Influences Outlierness (INFLO). The INFLO algorithm will be discussed next.

### 2.6.6.3 Influence Outlierness

The INFLO is a density-based algorithm. It is an improved version of LOF algorithm which can handle the local anomaly points where the clusters with different density are near to each other. The INFLO in addition to three steps that is explained previously for the LOF algorithm, it also computes reverse nearest

neighbours set of data points. One important difference between reverse nearest neighbours and k-NN is that the set of k-NNs typically have  $k$  neighbours data point were in reverse nearest neighbourhood set may have any amount depending on the data. For instance, Figure 2-21 which illustrates two clusters with varying densities, the red instance will be detected as anomaly by the LOF algorithm as it takes only the five nearest neighbours (the grey area) into account which have higher local density (Benetis *et al.*, 2006). The INFLO will additionally take the blue instances into account by computing reverse nearest neighbours, hence blue instances become reverse nearest neighbours of red instance which make the red instance less likely to be detected as anomaly by the INFLO.

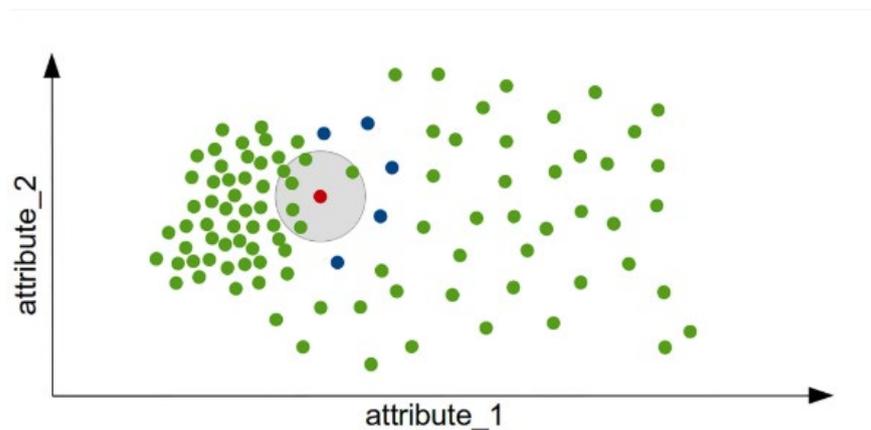


Figure 2-21 A comparison between the INFLO algorithm and the LOF algorithm (Goldstein and Uchida, 2016)

#### 2.6.6.4 Hierarchical Temporal Memory

Hierarchical Temporal Memory (HTM) is a sophisticated version of Neural Networks algorithms that is inspired by recent advances in neuroscience and the interaction of pyramidal neurons in the neocortex of mammalian brains. HTM theory mimics neocortex anatomy, neurons interaction in different layers of neocortex and how learning happens in the neocortex (Hawkins and Blakeslee, 2005; Spruston, 2008; Lewis *et al.*, 2018), originally described in Jeff Hawkins book '*On Intelligence*'. As Figure 2-22 shows, HTM Cortical Learning Algorithms (CLA)s are known for modelling spatial temporal features of input data using components including: the encoder, the Spatial Pooler (SP) algorithm, Temporal Memory (TM) algorithm and/or classification algorithm.



Figure 2-22 Components of HTM (adapted from(Cui, Ahmad and Hawkins, 2017))

The HTM is fundamentally a memory-based system and HTM networks are trained on lots of varying data and rely on storing a large set of patterns and sequences. The way data is stored and accessed is logically different from the standard model used by programmers today (Hawkins, Ahmad and Dubinsky, 2011). The classic computer memory has a flat organisation and does not have an inherent notion of time where the HTM memory has a hierarchical organisation and is inherently time based. The information in HTM communicating within regions, levels and between regions and levels, and to/from outside the hierarchy. The HTM receives the input data from the bottom level and the level will extract some basic features which then will be passed to the middle level and top level to learn more complicated features of the data.

For implementation of HTM theory, Numenta.org (Lavin and Ahmad, 2015) has developed the Numenta Platform for Intelligent Computing (NuPIC) platform where various libraries and interfaces based on the theory of HTM can be accessed. NuPIC provides set of APIs to access the classes of SP, TM and classification algorithms. The Numenta also provides the Numenta Benchmark (NAB) as an open source tool which offers a new metric to evaluate performance of the algorithms on streaming data and in an-online fashion.

The HTM will be discussed more in details in upcoming Chapter 3.

## 2.7 Anomaly Detection in AAL – Related Work

This section presents a review of recent literature on anomaly detection techniques applied in AAL in particular health monitoring to manage long-term conditions such as Chronic Obstructive Pulmonary Disease (COPD), heart failure and diabetes.

Hassan *et al.* (2019) proposed a Hybrid Real-time Remote Monitoring (HRRM) framework for monitoring the elderly patients suffering from chronic diseases in real time. The proposed HRRM process and transfer the vital signs data including heart rate, systolic blood pressure, diastolic blood pressure to classify the patient's health condition (e.g. hypertension, hypotension). One of the challenges in this work, is to deal with missing data which may affects the model's work continuity and its performance. Hence a version of Naïve Bayes classifier called NB-WOA is proposed for feature selection. The accuracy of the HRRM is tested using six classifiers and result indicated that Decision tree (J48), Random Forest (RF) and Ripper (JRip) classifiers produced best performance compare to Naïve Bayes (NB), Nearest Neighbour (IBK) and SVM. Moreover, several sampling techniques are tested, and CB and SMOTE techniques produced the best performance. The result of this testing suggested that the HRRM framework has a best performance in classifying patients' health status with high precision in real-time using JRip or J48 along with SMOTH or CB sampling techniques. Hence, the proposed framework has succeeded in increasing the accuracy of classifications and minimizing error rate.

Forkan and Khalil *et al.* has presented PEACE-Home, an HMM-based probabilistic estimator of abnormal clinical episodes using correlation patterns between vital signs. Multiple vital signs including heart rate and blood pressure is measured to monitor health status of the patients and their simultaneous changes indicate a transition of a patient's health status. The PCA method is used to classify the patients with known medical conditions into multiple categories and Hidden Markov Model (HMM) is used for probabilistic classification and prediction of future clinical states. The result of experimentation indicated that the proposed system could forecast clinical episodes and can generate real-time probabilistic estimation of anomalies with very good accuracy.

Ribeiro Filho *et al.* (2016) proposed Mobile Human Activity Recognition System (MHARS), an AAL system to monitor patients with chronic diseases. Two features are selected from the patients including Heart Rate (HR) and accelerometer and it has achieved an accuracy of more than 80%, which is satisfactory. The MHARS was also able to measure the activity intensity level by

measuring the HR, and to provide a rich set of features that includes the detection of user-defined situations.

Skubic, Guevara and Rantz (2015) have proposed a model for detecting health decline by monitoring behaviour and activity patterns of occupants. A one-dimension feature (1-D) alert algorithm was then implemented to generate health alerts to clinicians in a housing facility senior. Clinicians analyse each alert and provide a rating on the clinical relevance. These ratings are then used as ground truth for training and testing four classifiers, including Fuzzy Pattern Tree (FPT), Fuzzy k-Nearest Neighbour (Fk-NN), Neural Network (NN) and Support Vector Machine (SVM). The Experiment was conducted using embedded sensor data and health alert ratings collected on 21 seniors over nine months. Results shown that multi-dimensional classifiers perform significantly better than the 1-D algorithm, with the best 6-D performance at 86% by FPT compared to 39% at 1-D. Additionally, the results shown that a 6-D classifier that used the domain knowledge (unsupervised approach) perform slightly better than the best 6-D classifier are using supervised approach. The result of this study is then used to propose a model for detecting health decline with in-home sensors.

Ghayvat *et al.* (2019) presented a novel wellness indices modelling and detection methodology to improve the anomaly detection over the multiple sensory data in the AAL environment. The presented model classifies the activity in three stages and the whole task of identifying the activities are split into definite room-based sub-activities. The person's activities are distinguished based on locality, period and situation by applying the historical data, real-time and feedback received data. The authors have pointed out that the activity of daily living is unrelated from one person to another which merging the diversity of practices, lifestyle, and proficiencies in AAL environment. The authors proposed the wellness indices analysis model to overcome this obstacle which contrasts the observed actions with the "typical" behavioural pattern of the individual.

Mošmondor and Huljeni (2019) developed an intelligent privacy-aware home care assistance system, SmartHabits. The SmartHabits applying ML approached to learn the user's daily activity patterns, and automatically issuing warnings if

an abnormal situation is detected. Figure 2-23 shows different components of the SmartHabits system in order to interpret data collected from user, learn and process contextual information, detect anomalies and issue warnings. In this work, unsupervised MLs are applied on temporal data to abstract relevant data.

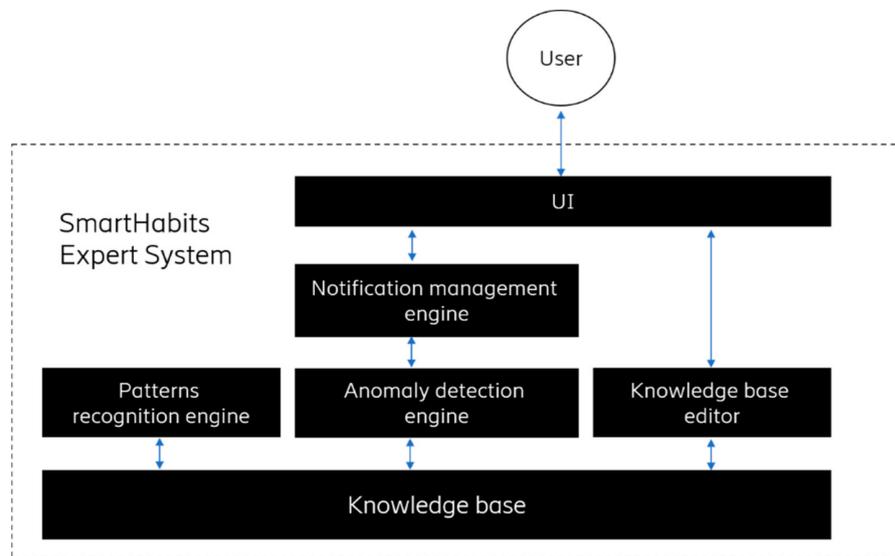


Figure 2-23 The SmartHabits Expert System (Grguri, Mošmondor and Huljeni, 2019)

As a part of the anomaly detection engine, clustering-based and rule-based approaches are used for identifying unusual behaviour of the user in a specific event or a specific situation.

Botia, Villa and Palma (2012) proposed an AAL system to monitor elders who live alone and want to keep living independently. The model scenario for developing this system is based on some assumptions. One assumption is that system deactivated when the elderly person is not at home and second assumption is that the elderly person has a pet, hence simple sensors make the detector not to function properly. Furthermore, a classifier is used for recognising potentially dangerous situations. Several activity sensors are located at different places such as at bed and the armchair to cover cases in which the elderly person is not moving and suffering from a health critical event. In order to ensure an abnormal activity is detected, an approach is used to enable and disable the classifier when needed. Additionally, a rule set is designed to avoid unwanted sources of sensors activation. To overcome an issue with wrong activation of the sensors in cases that the elderly person has a pet, the sensors that are not sensible to pets are used.

Temporal patterns of elderly activity are also abstracted in this work to further model the sensor data and to improve the detection rate of the proposed system.

## 2.8 Research Gaps

Anomaly detection and prediction in context of AAL in particular health monitoring have been studied in the literature, however more research efforts are needed to make these applications more robust and adaptive towards the changes in outside world. Current trends in AAL systems in particular, health monitoring indicate needs to introduce formal reasoning, as well as intelligent data analysis techniques in the extraction of knowledge, regularities and representative cases from patient data stored in medical record.

Yet a novel approach is required to prevent the cardiac arrest by early detection of abnormal patterns of vital signs for elderly patient's health monitoring. In particular a multi-layer framework to employ a common learning technique for clinical decision support to discover patient specific anomalies independently (adaptable to the new normal patterns) and to be capable of detecting contextual anomalies while detect an anomalous point as early as possible preceding a cardiac arrest.

The functioning of the human body relies on to the 24-hour environmental light/dark cycles which are also known as Circadian Activity Rhythm (CAR) (Liu and Chang, 2017). The CAR analysis is a powerful approach for measuring the rhythmic behavioural patterns of individuals and detect changes within these patterns. However, existing works lack in analysing the CAR and modelling the temporal changes in vital parameters. In order to achieve this, a level of intelligence is needed which current ML techniques fall short from modelling temporal patterns of the vital parameters, hence lacking in context-awareness. However, the human brain is more than capable of discovering context, and ML techniques that mimic information processing in human brain could help to push the research towards intelligence decision making.

The introduction of HTM theory and CLA algorithms that is inspired by recent advances in neuroscience is a potential candidate to fulfil the requirements for

detecting anomalies in an AAL health monitoring scenario. However, the HTM CLAs have not been applied in a health monitoring scenario where multiple vital signs are monitored to prevent the critical events. This is because of the fact that HTM is relatively a new theory that requires researchers to gain a better insight of its regions, network and CLAs in order to model the temporal patterns of vital signs data.

Another identified issue is related to the evaluation techniques that are currently used for testing performance of the algorithms. The performance of anomaly detection algorithms in the AAL health monitoring scenarios typically are evaluated by using the traditional evaluation technique (batched process) such as recall and precision. However, these methods do not take into the account the role of time in the evaluation process. In health monitoring scenario is important to prevent critical events by early detection of abnormal patterns. Hence, this work will also apply two forms of evaluation techniques, including NAB and traditional evaluation techniques. These techniques were discussed in Section 2.4.6.

Anomaly detection is in the early stages in context of health monitoring and is more established in other fields, for instance, image processing, theft uncovering, scam finding, medicinal, manufacturing, and more (Ghayvat *et al.*, 2019).

## 2.9 Summary

In this Chapter, the suitability of assisted living systems is reviewed in particular the AAL that are utilised for monitoring health conditions of elder people who are living independent. Moreover, the AmI paradigm is discussed and its influence and relation to the traditional assisted living tools is investigated. Different principles and technologies that contributed to development of AAL in particular health monitoring applications are discussed. Ubiquitous/pervasive computing, wearable technologies, context-awareness and AI are investigated as it was important to understand how each one of these paradigms and disciplines contribute to the intelligence, ubiquitous and adaptivity of the AAL environment. Data mining techniques, including data processing and modelling of user's behaviour in particular vital signs are discussed. Furthermore, the importance of

temporal reasoning in health monitoring is researched which plays an important role in modelling the time related changes in vital signs that are commonly used for diagnosis, prognosis and decision-making tasks. The challenges with modelling temporal patterns and correlation between the vital signs is of interest in this project, therefore, to model these temporal patterns there is a necessity for applying intelligent data analysis techniques.

A review of ML techniques and modelling for prediction and anomaly detection tasks is also conducted. The reviewed techniques covered the classification, regression, clustering and association techniques. A more focused review was dedicated towards the anomaly detection in AAL environment in particular health monitoring scenarios where vital signs of patients are monitored to classify the patient's medical conditions and to prevent the upcoming critical events. There are limited works that specifically consider the adaptability characteristic of AAL systems to the new normal patterns where ML approaches such as statistical, nearest neighbour, and density-based approaches are used. The drawback of these approaches in AAL health monitoring is that they do not consider the time dimension for computing an anomaly score. In AAL health application, time dimension of vital signs is extracted and used to give a different semantical meaning of similar data points that occur in different occasions. Additionally, the current applications of AAL health monitoring do not consider the patterns of patient's daily activity (e.g. sleeping, walking, exercising), hence extraction of this knowledge can improve the context model of the health conditions to potentially improves the performance of existing solutions where false detection of anomalies (misdiagnosis) can be avoided. Moreover, few works are currently focused on elderly patient population and are mainly tackling problems of AAL system for telecare, wellness, digital participation and teleworking services.

Furthermore, the HTM theory, the CLAs and NuPIC were researched as the HTM will be further investigated and it will be applied to the proposed framework.

Finally, the Chapter is concluded with reviewing the related work of anomaly detection in AAL health monitoring application and a discussion on the research gaps in the literature for anomaly detection in AAL health monitoring.

## Chapter 3

# HIERARCHICAL TEMPORAL MEMORY

In this Chapter, there will be a discussion on Hierarchical Temporal Memory (HTM), a biological inspired machine intelligence theory. The HTM theory is developed by Jeff Hawkins in 2005, which is implemented using a number of sophisticated neural network algorithms. The first generation of the algorithms called Zeta 1 which are replaced by the new algorithms called “Fixed-density Distributed Representation”, or “FDR”. However, this terminology is not used anymore, and it is replaced by new terminology: HTM Cortical Learning Algorithms (CLA) or sometimes just the HTM Learning Algorithms. In this Chapter, the HTM theory will be discussed and the algorithms of the HTM will be investigated in order to have a clear understanding of the algorithms that can be used in anomaly detection and prediction tasks by learning patterns of stream data.

The discussion starts with highlighting ability of the neocortex to recognise and predict ordered temporal sequences. Discussion will be followed by a study on the HTM theory and principles that have mimicked the architecture and process of

the neocortex. The main components of the CLA will be presented, including the encoder, the Spatial Pooler (SP) algorithm and Temporal Memory (TM) algorithm. This Chapter will conclude with a review of several anomaly, classification and prediction algorithms. The information that are discussed in this Chapter will be used in forthcoming Chapters where the proposed framework will be implemented and tested.

### **3.1 Cortical Facts**

The neocortex has the ability to recognise and predict ordered temporal sequences that is an important function of the brain by continually processing a stream of sensory data and developing a rich contextual model of the world, including speech recognition and natural vision (Cui, Ahmad and Hawkins, 2016). The neocortex is a thin sheet of brain neurons, about '2.5' millimetre thick, that functionally and anatomically is very uniform among different species. There are billions of neurons in a neocortex tissue and the majority of them are pyramidal neurons (Hawkins and Ahmad, 2016), and in this Chapter the focus will be on this type of neurons. The neurons are vertically interconnected in a columnar structure that maps a complex hierarchical network of neurons. This hierarchy can be different in size and depth among different species, but they will have uniform behaviour and structure.

The neurons in the neocortex are also organised into six horizontal cortical layers (Figure 3-1), the layers are numbered with Roman numerals from superficial to deep, including layer 1 consisting of a few neurons, layer 2 is relatively thin and it is usually combined with layer 3, layer 4 contains irregular shaped neurons, layer 5 consists of large pyramidal neurons and layer 6 consists of small neurons.

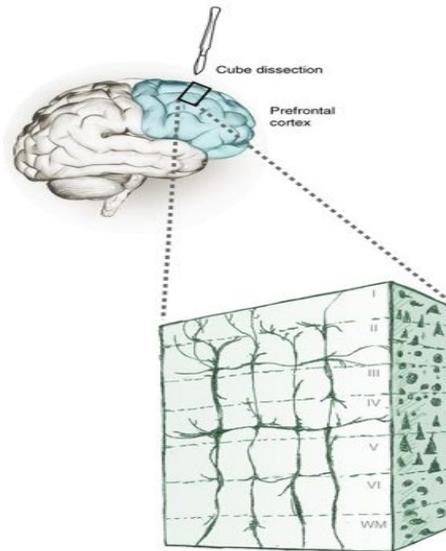


Figure 3-1 six layers of neocortex (adapted from He *et al.*, (2017))

The primary components of the neuron are the cell body (soma), the axon, dendrites and synapses, shown in Figure 3-2. The dendrites are hair-like structures which act like input channels, and thousands of synapses are segregated onto them. The function of the synapses on dendrites is to receive information from other neurons and sources.

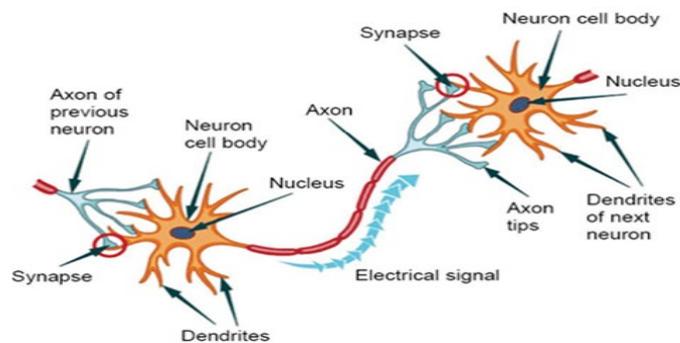


Figure 3-2 primary components of the neuron (Devineni, 2015)

The dendrites are divided into proximal, basal (distal) and apical dendrites. The synapses on proximal dendrites can recognise multiple unique feedforward patterns that can lead to a neuron's firing (Hawkins and Ahmad, 2016; Hawkins, Ahmad and Cui, 2017). The basal dendrites are far away from soma and receive information from neurons within a layer (layer 2/3), the basal dendrites recognise patterns of neuron activity that lead to the neuron's *depolarisation*. The synapses on apical dendrites receive information from the higher region in

hierarchy (feedback input), and they can also depolarise a neuron (Hawkins and Ahmad, 2016).

A pyramidal neuron when it is electrically charged (voltage gated) (Spruston, 2008) and its voltage reaches a certain threshold, generates a *spike* that transmits information down an axon, away from the soma to reach a different neuron, this is known as an *action potential*, it is also named *somatic action potential* (Spruston, 2008; Hawkins and Ahmad, 2016). A neuron also receives inputs from other sources through the synapses within its dendrites, a subset of synapses will be activated when an input pattern is detected, if this happens in close spatial proximity on a dendrite then it generates a N-methyl-D-aspartate (NMDA) dendritic spike. In this Chapter the focus is on two types of NMDA dendritic spikes, NMDA spike generated in basal and  $\text{Ca}^{2+}$  spike generated in apical. These dendritic spikes precede the neuron *depolarisation* (Spruston, 2008; Cui, Ahmad and Hawkins, 2016; Hawkins and Ahmad, 2016), and it is an essential change in a neuron's state for an action potential to be fired.

In Figure 3-3, the neuronal compartment that serves as the spike initiation site is coloured in black. The basal dendrites generate NMDA spikes, but it never contributes to neuron firing, rather it leads to neuron *depolarisation* due to the fact that the basal dendrites are far from soma, and when the NMDA spike reaches the soma, it does not have sufficient amplitude to generate an action potential. The basal dendrites mainly consist of contextual information on input information from the neuron activity (transition in sequences). The depolarisation is a term used to describe a neuron when is in a state that has a higher priority than a neighbouring neuron to be fired (become active) in a short time. In other terms, a neuron in a depolarised state, i.e. predicts that it will be fired soon (Hawkins and Ahmad, 2016) when it receives sufficient feedforward input. The apical dendrites generate a  $\text{Ca}^{2+}$  spike that establish another form of prediction (top-down expectation) which has similar impact on some as NMDA spikes generated from basal dendrites, and in both cases, they lead to neuron depolarisation. In a work by Hawkins *et al.* in (Hawkins and Ahmad, 2016) it is stated that the interaction between apical  $\text{Ca}^{2+}$  spikes, basal NMDA spikes and action potential is not entirely discovered and it is an ongoing research area.

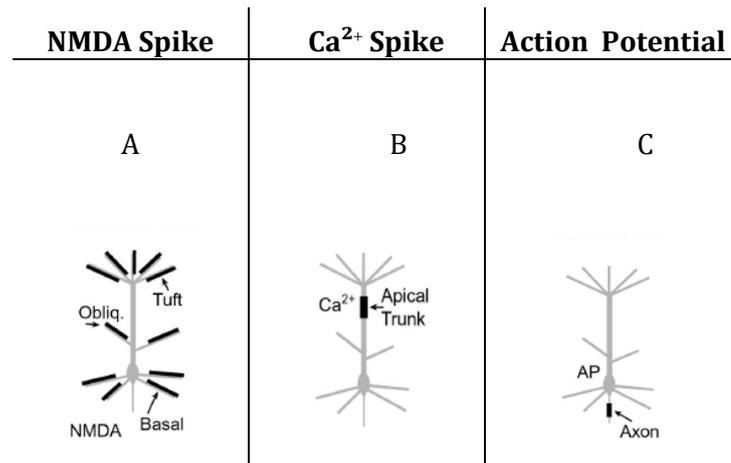


Figure 3-3 Neuronal spike initiation site

These cortical facts make the learning in mammals possible, the complex neocortex anatomy and cortical regions that lead to learn and predict sequences of events in the outside world, are a fundamental concept for a variety of artificial neural networks. Neurons in different layers of the neocortex, have different roles and functions that make it possible to learn about the outside world and make a prediction of future events. This is like when a person is listening to the first notes of a song, and if they have heard this song in the past, then they can sing the song as they can predict the forthcoming notes.

### 3.2 HTM Theory

The HTM theory is inspired by recent advances in neuroscience and the interaction of pyramidal neurons in the neocortex of mammalian brains. HTM theory is an attempt to mimic neocortex anatomy, neurons interaction in different layers of neocortex and how learning happens in the neocortex (Hawkins and Blakeslee, 2005; Spruston, 2008; Lewis *et al.*, 2018), originally described in Jeff Hawkins book '*On Intelligence*' (Hawkins and Blakeslee, 2005).

At the time of writing this Chapter as illustrated in Figure 3-4, a small subset of the HTM theory is completely implemented. Each layer in a region implements variation of a common sequence memory algorithm. Layer 2/3 of HTM framework is implemented and commercially tested, the function of this layer is to learn and predict temporal sequence of feed-forward input by modelling

temporal dynamics of the data. The *sensory motor transitions* layer (labelled as layer 4) is in research and progress state (Hawkins, Ahmad and Cui, 2017; Lewis *et al.*, 2018), the neurons in this layer of neocortex represent an important function of the brain, and this layer is also referred to input layer (Hawkins and Blakeslee, 2005). Neurons in this layer are capable of modelling complex objects by joining the signals from sensory organs and motor cells to create concepts. Layers 5 and 6 are poorly understood at the time of writing this Chapter.

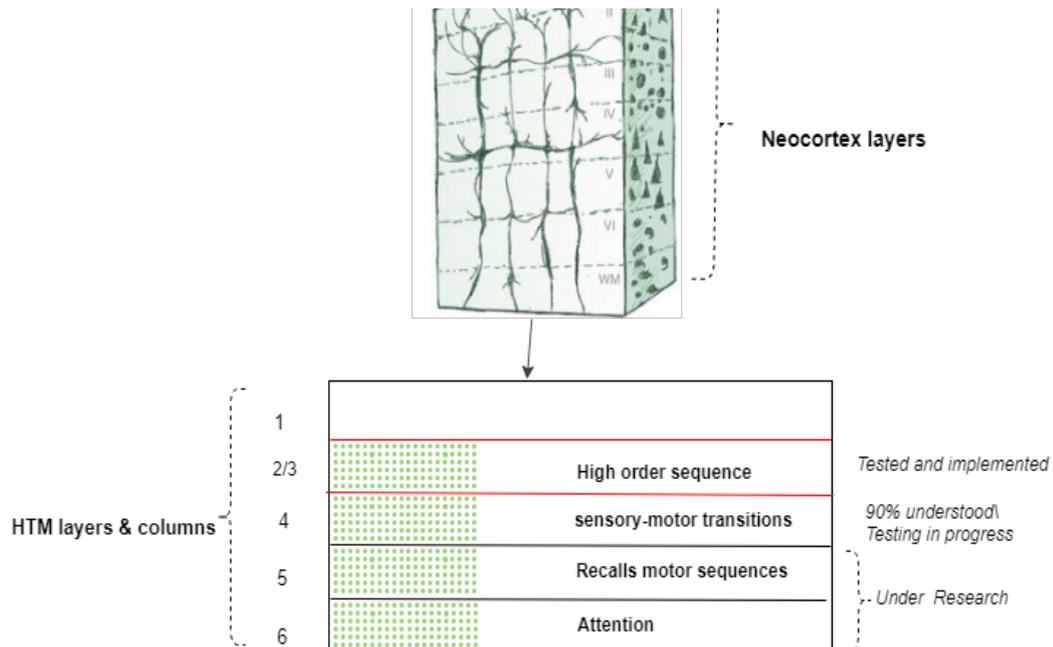


Figure 3-4 HTM Theory – layers of columns in HTM region

HTM is a new form of neural network that can be used in a learning and prediction machine. In HTM, neurons (cells) are arranged in columns, in layers, in regions, and in a hierarchy. The HTM essentially is a system based on memory system similar to the brain, it is trained on time series data (inherently time based) and relies on learning sets of sequences and sequences of patterns. Information in the HTM memory is always stored in a distributed fashion, and the HTM can be modelled as long as the key functions of hierarchy, time and Sparse Distributed Representations (SDR) of information are incorporated.

### 3.2.1 HTM Principles

The HTM theory has the following principles: hierarchy, regions, sparse distributed representation and time.

#### 3.2.1.1 Hierarchy

As mentioned earlier in this Chapter, the HTM is a system based on memory, it consists of different regions that are structured in a hierarchy. The regions represent the main elements of a memory, and typically are located and organised in different levels of the hierarchy (Hawkins, Ahmad and Dubinsky, 2011), Figure 3-5 shows a four-level hierarchy, each layer in the hierarchy consists of a region. The region receives and sends information up and down the hierarchy. As information ascends the hierarchy, the output of regions in lower levels will converge, and the regions on higher levels of the hierarchy make use of the information to produce a bigger picture of an outside world and make the learning and prediction of HTM possible.

Hawkins in his book (Hawkins and Blakeslee, 2005) has highlighted that information can flow upward or downward in the hierarchy. The inputs from lower regions always converging at layer 4 (input layer) of a region in a higher level of the hierarchy. The information from the input layer then will be fed-forward to layer 2/3, where next it will be sent to the input layer of a region in a higher level of hierarchy. Also, information can flow down the hierarchy which does not take a similar direct route to when it goes upward. The journey of information (feedback) when flowing down the hierarchy begins in layer 6 of a higher region and arrives in layer 1 of a lower region. It activates some of the neurons in layers 2/3, 5, and 6 within the region which then will flow downward to layer 1 in lower regions.

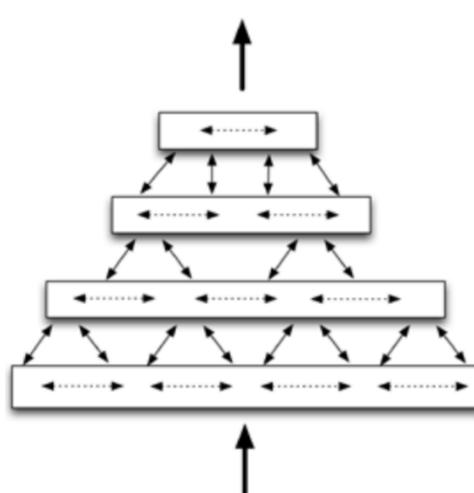


Figure 3-5 Information flows up and down hierarchies (Hawkins and Blakeslee, 2005)

One advantage of a various levelled structure is efficiency as it significantly decreases training time and memory use, this is a result of patterns, found at each level of the hierarchy, are reused when consolidated in novel ways at higher levels. The number of levels in the hierarchy is reliant upon the sum memory that we apportion to a level, as such HTM able to consequently take in the most ideal representations at each level given the measurements of the information and the amount of resources allocated. For instance, in the event that we allocate more memory to a level, the level will shape representations that are bigger and more complex, however that implies it might need less levels. On the other hand, in the event that we assign less memory for a level, at that point representations will be littler and simpler which may mean more various levels might be required.

### 3.2.1.2 Regions

The terminology region comes from the biology. The input to an HTM region may come directly from the outside world (e.g. senses), usually regions at bottom layer of the hierarchy, receives this information, or information may come from other regions (regions-to-regions connectivity). Generally, regions are responsible for a given functionality and they may vary in size and their position in the hierarchy (Hawkins, Ahmad and Dubinsky, 2011). The HTM region consists of six layers of cells that are interconnected and arranged in columns of cells. Figure 3-6 shows a small section of the HTM region that is equivalent to one layer of neurons in a neocortical region, For instance the HTM layer 2/3 in Figure 3-4. The HTM region can implement inference and prediction on complex data by activating and

inhibiting subsample of columns of cells. This will be discussed more in detail in this Chapter.



Figure 3-6 - Implementation of a section of an HTM region comprised of columns of cells

### 3.2.1.3 Sparse Distributed Representations

The Sparse Distributed Representation or in short, the SDR is a one-dimensional binary array representation of information in the HTM system, consists of zero bits (inactive cells) and one bits (active cells) (Ahmad and Hawkins, 2016). Each bit in the SDR has semantic meaning, there is usually 2% of total cells in active state at a given time (sparse) and information are encoded and represented cross set of active cells (distributed) (Purdy, 2016). The SDR is not like ASCII encoding where one-bit changes then total meaning of data changes, this is not a case in the SDR as if one-bit changes in an SDR it should not have any effect on the meaning of the data.

The size and sparsity of input bits in SDR are defined by  $n$  (number of bits) and  $w$  (width) for number of ‘one bits’. For instance, consider an example of an SDR  $y$  with  $n = 48$  and  $w = 15$ :

$y = [0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1].$

In HTM, SDR ‘ $y$ ’ with  $n = 48$  and  $w = 15$  represents an input data as a total of 48 bits of which 15 bits are active. Figure 3-7 illustrates a two-dimensional presentation of an SDR ‘ $y$ ’, including active bits, which are coloured in red.

The SDR is a primary data structure that is used in HTM system, it is used in different regions within the HTM hierarchy to offer different functions for the HTM system. Any input data before is fed into the HTM system needs to be

encoded to an SDR format (Cui, Ahmad and Hawkins, 2016; Wu, Zeng and Yan, 2016). The bottom layer of the HTM hierarchy consists of regions that are responsible for converting the inputted data from the outside world into an SDR (Hawkins and Blakeslee, 2005), and the output of any HTM region will be an SDR that may feed into a different region within the HTM hierarchy. Some of the properties and operations of SDRs are such as overlap sets, uniqueness and exact matches, inexact matching and subsampling that are discussed in (Ahmad and Hawkins, 2015).

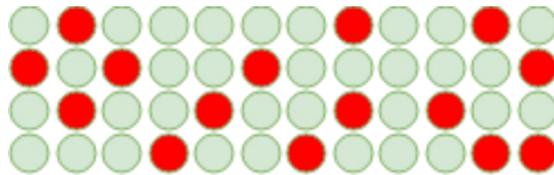


Figure 3-7 A Sparse Distributed Representation of an Information in an HTM Region

#### 3.2.1.4 Time

As highlighted in previous sections, the HTM system is based on memory, and time is playing an important role as the supervisor for training the HTM to learn the temporal sequences. Three main functions of the HTM system are learning, inference (pattern recognition) and prediction of sequences. These functions are reliant on time, learning in HTM must be done by using time series inputs during learning, these inputs are essentially sequences of SDRs that travel from the senses to the different regions in an HTM hierarchy to model the outside world. Learning and recognizing sequences is the basis of forming predictions where it uses high order sequences property (Hawkins and Ahmad, 2016; Wu, Zeng and Yan, 2016). These functions will be discussed more in details in this Chapter.

#### 3.2.2 HTM Neuron Model

The HTM neuron model is inspired by recent neuroscience research that are highlighted in the above sections of this Chapter. The HTM neuron model is more sophisticated than the one used in other types of ANNs such as those in deep learning and spiking neural networks (Wu, Zeng and Yan, 2016).

The artificial neuron model that is used in most of the traditional ANNs has few synapses and no dendrites, it is often called “point neuron”, in which a single weighed sum of its synaptic inputs is computed, Figure 3-8 (A).

The Figure 3-8 (B) shows a biological neuron, the soma is highlighted in green it has thousands of synapses arranged along dendrites. The synapses close to the soma are proximal synapses that receives feedforward inputs, the synapses far from the soma including distal and apical are input zones of contextual and feedback inputs.

The HTM neuron model shown in Figure 3-8 (C), consists of active dendrites as well as proximal, distal and apical dendrites which mimics more properties of the biological neuron model that is shown in Figure 3-8 (B). The soma is represented as a grey triangle, the synapses are presented as dots and active synapses are presented as colour filled dots. The feedforward input is transmitted to the soma from the proximal synapses (highlighted in green) and lead to action potential when enough active dendrites are formed on the feedforward. On the other hand the context input is transmitted to the soma from the distal synapses (highlighted in blue) coming literally from nearby cells within the region and the feedback input received by the soma from its apical synapses from an above region. The inputs are received from distal and apical synapses initiates a dendritic NMDA spike that depolarises the soma, or in other term it changes the state of a cell to predictive state. The cell when in predictive state, predicts that it will become fired shortly.

The fact about the complexity of the HTM neuron model is that Hawkins truly believes active dendrites and synapses have a different impact on the behaviour of the soma and they are key functional aspects of biological neuron model that are missed in traditional artificial neuron models. The proximal synapses have a large effect at the soma and initiates action potential, and they are forming 90% of the total synapses of a neuron. The remaining 10% of synapses are formed by synapses far from the soma, they may construct a small portion of synapses that have little effect on the soma, but from the bigger picture they play an important role in learning and prediction in the sequence memory of the brain (Huaman *et*

*al.*, 2015; Cui, Ahmad and Hawkins, 2016). In cases where several active distal synapses are close in time and space, they initiate dendritic NMDA spikes that depolarise the soma that precedes the neuron firing and inhibition of nearby cells in a region. This results in highly sparse patterns of activity for correctly predicted inputs (Hawkins and Ahmad, 2016).

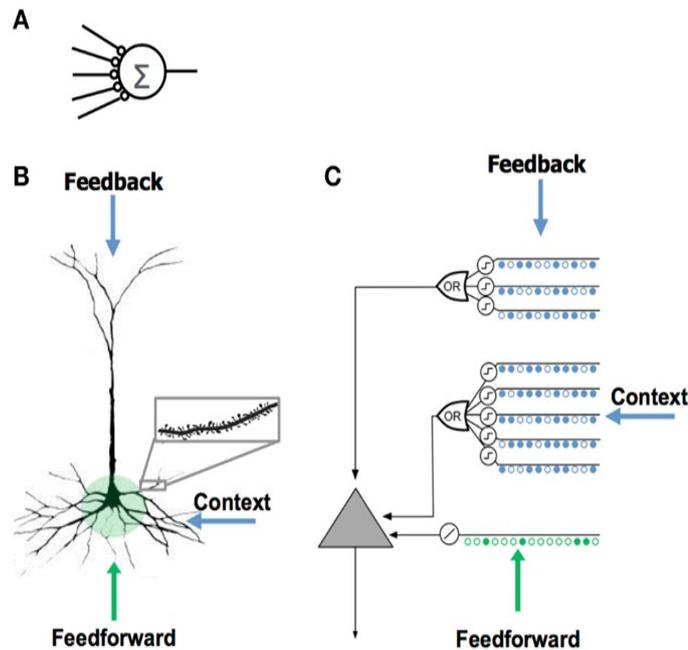


Figure 3-8 A Comparison between a biological neuron and the HTM neuron (Hawkins and Ahmad, 2016)

### 3.2.3 HTM Functions

In HTM network, every HTM region has the following Three basic functions: learning, inference and prediction.

#### 3.2.3.1 Learning

The HTM cortical learning algorithms are on-line learning algorithms, there is no need to separate a learning phase from an inference phase as the regions are capable of on-line learning and they can continually learn from each new inputted data. Learning in the HTM region occurs by discovering patterns in input data. The HTM region does not need to know what inputs represent, it works on the SDR that consists of an array of binary numbers which is a blend of information bits that happen together regularly. The HTM uses SDRs to learn spatial patterns

of the input data, and it is reliant on time to learn temporal patterns for how these spatial patterns show up.

### 3.2.3.2 Inference

As highlighted previously, inference is similar to pattern recognition. HTM when completes the learning phase, it moves to inference phase to carry inference on a new input (it can be possibly novel input). The HTM in inference phase will match the new input to previously learned spatial and temporal patterns.

Successfully matching novel input to previously learned sequence is the core of inference and pattern matching. The HTM in inference phase, continuously looks at a stream of inputs and matches them to previously learned sequences. The approach of an HTM region to cope with a novel input in the inference phase, is through the use of SDRs, as a key property of an SDR is to match a portion of the patterns to be confident that the match is significant.

### 3.2.3.3 Prediction

HTM regions stores sequence of patterns and transitions between them, the regions can form a prediction about what will likely arriving next by matching stored sequence (learned sequence) with the current input. For the HTM regions to carry out the prediction, it requires the majority of memory in HTM to be allocated to sequence memory or storing transitions between spatial patterns. Some key characteristics of HTM prediction are highlighted below:

- *Prediction is continuous*: HTMs do the same thing that we as human do constantly predicting. For example, while we are behind a traffic light, we keep predicting when the light will change. In an HTM region, prediction is not a separate step and it is an important part of an HTM region, the prediction and inference have also similar behaviour and characteristics.
- *Prediction happens in every region at every level of the hierarchy*: regions can make predictions of the patterns by referring to what they have learned.
- *Predictions are context sensitive*: the fact is that the predictions are based on historical data as well as what is occurring now, therefore the collected

data as an input can produce predictions based on previous contexts. One ability of HTM regions is known as “variable order” memory, this means that the HTM regions learns to use as much earlier context as required and can keep the context over both short and long sections of time.

- *Prediction leads to stability*: the output of a region is its prediction, a region not only can predict what will happen next, but it also can predict multiple steps ahead in time.
- *A prediction tells us if a new input is expected or unexpected*: the HTM regions can act as detector, as they can predict what will happen next, therefore it can know when something unexpected happens and know that an anomaly has occurred.
- *Prediction helps making the system more robust to noise*: the prediction can help the system to fill in missing data, as a result of HTM prediction, the prediction can bias the system toward inferring what it predicted.

### 3.3 Cortical Learning Algorithms

The Cortical Learning Algorithms (CLA) are a set of algorithms that implements functions of the HTM theory where two representations of the SDR in a region can be computed when applying the SP and TM algorithms. Currently the CLA implements some parts of the HTM theory and some aspect are still under active research. Figure 3-9, shows components and algorithms provided by NuPIC. There are three main processes that are triggered when data is fed into an HTM region including:

- Form a sparse distributed representation of the input.
- Form a representation of the input in the context of previous inputs.
- Form a prediction based on the current input in the context of previous inputs.

The input to the HTM network, will be encoded to the SDR using an appropriate encoder. Input data to an HTM region can come from a sensory data or from another region lower in the hierarchy. There are numerous types of encoders available that will be discussed in this section. The SDR created by an encoder

will be fed to the HTM region, and it will be inputted into the SP algorithm. The SP algorithm operates on the columns level of the region where the HTM learns the spatial patterns of input space. The SDR output from the SP will be sent to the TM algorithm. The TM algorithm operates on active columns to activate and deactivate the cells within the columns which enables the HTM to learn the transition between spatial patterns. The output of the TM algorithm could be used by different type of anomaly detection algorithms or classification algorithms to detect and predict unexpected patterns respectively.

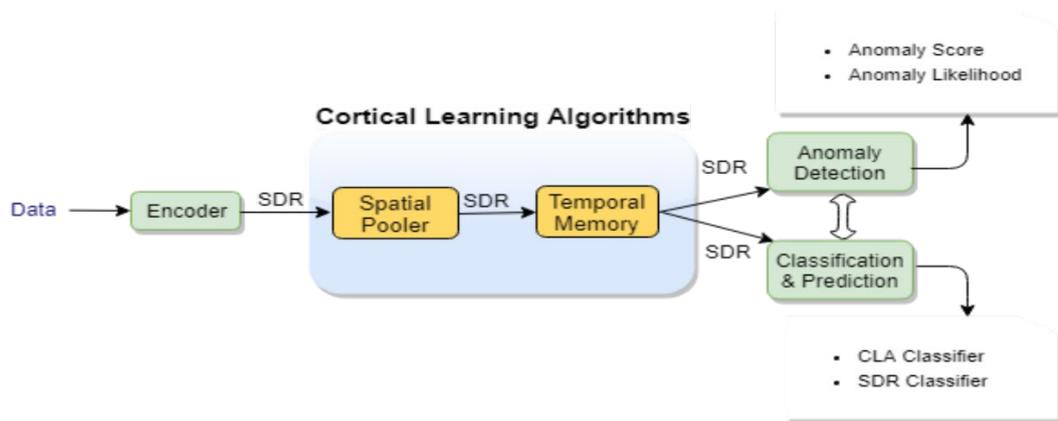


Figure 3-9 Functional steps for using HTM on real-world sequence learning tasks

The HTM network represents high-order sequences (sequences with long-term dependencies) using the composition of two SDRs. At any time, both current feedforward input received from proximal dendrites (activate the columns of cells) and previous sequence context received from distal synapses (depolarise the cell) are at the same time represented using the SDRs (Cui, Ahmad and Hawkins, 2016).

The first SDR is at the column level (Figure 3-10). At any time, the top 2% of columns that receive most active feedforward inputs are activated, this is the task of Spatial Pooler (SP) algorithm which operates on the columns' level of the region where the HTM learns the spatial patterns of the input space.

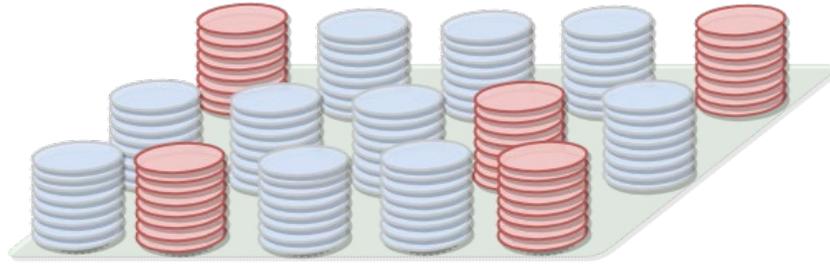


Figure 3-10 - The SDR at column level

The second SDR (Figure 3-11) is at the level of individual cells within the active columns (winner columns). At any given point a subset of cells in the active columns stores temporal contextual information of current patterns which can lead to the prediction of upcoming input data. The cells within a column when are in a predicted state then in the next time step if that column receives sufficient feedforward input these cells first become active which inhibit other cells within that column (Hawkins and Ahmad, 2016; Wu, Zeng and Yan, 2016). If there are no cells in predicted state and column become active then all the cells with that column become active, usually this happens when a new input is fed to the HTM region. The Temporal Memory algorithm (TM) operates on active columns by activating and deactivating the cells within the columns and this is how the HTM learns the transition between spatial patterns.

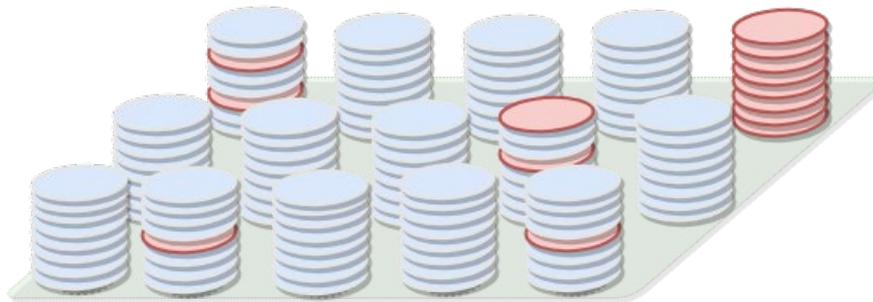


Figure 3-11 - SDR at the level of individual cells

### 3.3.1 Encoder

The HTM regions represent all information using sparse patterns of input data, this makes an HTM system reliant to the SDRs. Any data that can be converted to an SDR can be used in a wide range of applications using HTM systems. The SDR is a primary data structure in an HTM system that consists of a large array of bits of which most are zeros, and few are ones (usually constitute around 2%).

In the HTM region, ‘zero’ bits represent inactive cells and ‘one’ bits represent active cells which are used by an HTM system to learn temporal sequence patterns of input data. The requirement of the encoder is to change over the input data (e.g. temperature, time stamp, image, GPS location, etc.) to the SDR that can figure out which output bits ought to be ones, and which ought to be zeros for that input so that it can catch the semantic significance of the data.

There are several encoders which are accessible through NuPIC<sup>5</sup> for various type of information, these encoders can be ordered to numerical, categorical and Geospatial Coordinate Encoders. According to (Purdy, 2016), the encoders have important features that impact the capacity of the HTM systems to create an optimum result:

- *Capturing semantically similar data:* the encoder should create representations that overlap for inputs that have similarity in one or more of the characteristics of the data that are selected, this means when two SDRs have more than a few overlapping one bits, then those two SDRs have similar meaning (Purdy, 2016).
- *Producing same output for same input:* it should be deterministic, and it should not change the representations of outputs for the same inputs.
- *Producing fixed dimensionality outputs:* the encoder must always produce an output consists of the same number of bits for each of its inputs. SDRs are compared and operated on using a bit-to-bit assumption such as a bit with certain “meaning” is always in the same position. The reason behind this is that if the encoder produces varying bit lengths for the SDRs, then comparison and other operations would not be possible.
- *Producing outputs with fixed sparsity:* for the encoder, keeping the sparsity the same is a rule that should always be followed.

For the rest of this section, the three most common encoders will be discussed.

---

<sup>5</sup> The NuPIC is an open source platform. <http://numenta.org>

### 3.3.1.1 Numbers

Encoders that are used for numbers are Scalar, Adaptive Scalar, Delta, and Log.

- *Basic Scalar Encoder*: To use this encoding scheme, it requires to know the range of the data that it will be encoded, when data falls outside the range then the encoder does not work well, typically it uses smallest bucket for values below the range and the largest bucket for values above the range (Purdy, 2016). The scalar encoder has four parameters: *minimum value*, *maximum value*, *number of buckets*, and *number of active bits ( $w$ )*. Therefore, we always need to decide the entire range of real values by defining a *min* and *max* value up front, these values will be fixed in this approach, the scalar encoder saturates representation of input at the edge and cannot change the range of the encoder at run-time, or learning will be lost. The scalar encoder has four parameters: *minimum value*, *maximum value*, *number of buckets*, and *number of active bits ( $w$ )*.
- *Adaptive scalar encoder* is an implementation of the scalar encoder that adapts the min and max of scalar encoder dynamically. It is useful when the range of input values is not known, therefore this encoder will adapt the *min* and *max* values based on the range of the input data.
- *Numeric Log encoder* is suitable when there is a need to capture similarity between numbers differently based on how large the number is, it is sensitive to small changes for small numbers (e.g. a change from 5 to 21) and it is less sensitive to changes for larger number (e.g. a change from 3500 to 4500).
- *Delta encoder* is suitable for capturing the semantics of the change in a value rather than the value itself and it produces overlapping encoding for values that increased or decreased from the previous value by a similar amount.

### 3.3.1.2 Categories

Some datasets include categorical information, in some cases the data can be discrete, completely unrelated categories or data that consists of categories that

may have some relation, for example characteristics of patient's condition and gender, for instance:

- Male with Hypertension vs female with hypotension
- Male with bradycardia vs female with bradycardia

Other example can be for dates and times that are categorical in nature:

- Weekday vs weekend
- Holiday vs non-holiday
- Days vs night
- Time of day
- Day of the month

### 3.3.1.3 Coordinate and Geospatial Coordinate

The Geospatial Coordinate Encoder (GCE) is a subclass of the Coordinate Encoder (CE). GCE converts and encodes a GPS position to an SDR, and it has the following properties:

1. Position spatially close together having overlapping bits in the encoding.
2. When moving at low speeds, resolution of movement is finer, and when moving at high speeds, resolution of movement is bristlier.
3. It works anywhere in the world and works for an infinitely large space.

GCE has three parameters including *latitude*, *longitude* and *speed*, where CE has two parameters including coordinates and radius.

### 3.3.2 Spatial Pooler

The main task of the Spatial Pooler (SP) algorithm is to convert the region's input into a sparse pattern which is used in the HTM system to learn sequences and make predictions. There are different steps that will be carried out by the SP to achieve its main task, including learning the connections to each column from a subset of the inputs, determining the level of input to each column, and using inhibition to select a sparse set of active columns.

The input to the SP is just a binary vector of zeros and ones, each column is connected to some random subset of the input bits (50% of input), which is called the “potential pool”. Each connection has a synapse which is a connection between the column of the input, and each of these synapses has a value associated with it called the “*permanence*”. The permanence of a synapse is a scalar value ranging between “0.0” to “1.0”. In column level, learning happens by incrementing and decrementing permanence of a synapse. A synapse’s permanence when is above a threshold, it is connected with a weight of “1” and the synapse will be affected by the input bit, on other hand when it is below the threshold, it is unconnected with a weight of “0” and the synapse will not respond to the input bit and it will not be affected. A new record when is inputted into the HTM regions, two processes occur. The first process is to select which columns become active therefore a small subset of the columns become active, and second process is to select which cells within the activated columns should become active as each cells represents different temporal context (input with a different set of history behind it). The concept is that active columns represent the current input, so this helps to have similar representations for similar inputs in a spatial pooler. The pseudocode for the SP algorithm is documented in Appendix A. The SP algorithm includes the following steps:

#### 3.3.2.1 Initialization of parameters

This is the first phase in a SP algorithm, prior to receiving any input, the SP initialization is computed by allocating a list of initial potential synapses for each column. The SP links each column to a random set of binary inputs from the input space (potential pool), in Figure 3-12, the potential connections for two active columns are represented. Each input has synapses with an assigned random permanence value, this permanence value is selected to be in a small range around the permanence threshold to enable potential synapses to become in “*connected*” or “*disconnected*” state after a small number of training iterations.

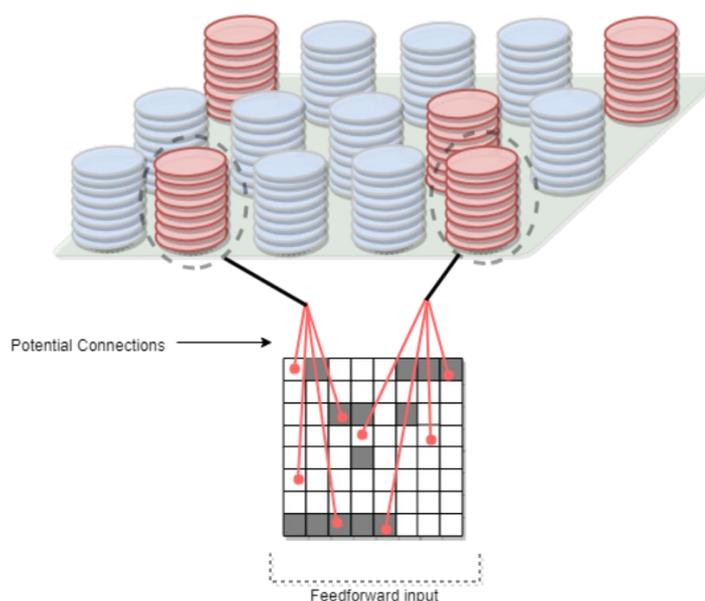


Figure 3-12 Columns in the SP and the potential connections

Table 3-1 shows some of important parameters of the SP algorithm for initialization task.

Table 3-1 Parameters are used in the SP algorithm for initialization

Parameter	Description
<i>columnDimensions</i>	It is a sequence representing the dimensions of the columns in the region. Default "2048"
<i>potentialPct</i>	Or potential percent is defining percent of inputs within a column's potential radius, that a column can be connected to. If it has a value 1 then it means the column will be connected to all input bits within its potential radius. Default value "0.85"
<i>synPermConnected</i>	The default connected threshold. Any synapse whose permanence value is above the connected threshold is a "connected synapse", meaning it can contribute to the cell's firing. Default "0.1".

### 3.3.2.2 Overlap Score

Given an input vector, the SP in this phase calculates the overlap score for each column with that vector. The overlap score for a column is defined as the number of synapses in a "connected" state (connected synapses) that are connected to active inputs ('ON' bits), multiplied by its boost factor.

### 3.3.2.3 Inhibition

The SP algorithm in this phase computes columns inhibition process and then computes the winning columns by calculating which columns remains as winners after the inhibition. The “*globalInhibition*” parameter is used in this phase to define a method of inhibition including local or global inhibition. There are also two parameters, including “*localAreaDensity*”, and “*numActiveColumnsPerInhArea*” that specify the density of active columns, during the inhibition. These parameters are used to calculate the maximum number of columns to remain ON/active within a local inhibition area. Table 3-2 provides more details on the parameters.

The inhibition step ensures that each column has enough connections to input bits to become active. The overall score of a column must be equal or greater than value of “*stimulusThreshold*” parameter in order to be considered during the inhibition step. The columns without such minimal number of connections, even if all the input bits they are connected to turn on, have no chance of obtaining the minimum threshold. For such columns, the permanence values are increased until the minimum number of connections are formed.

Table 3-2 Parameters are used in the SP algorithm for inhibition

Parameter	Description
<i>globalInhibition</i>	If true, then during inhibition phase the winning columns are selected as the most active columns from the region as a whole. Otherwise, the winning columns are selected with respect to their local neighbourhoods. Using global inhibition boosts performance x60.
<i>localAreaDensity</i>	The desired density of active columns within a local inhibition area.
<i>numActiveColumnsPerInhArea</i>	An alternate way to control the density of the active columns. If <i>numActiveColumnsPerInhArea</i> is specified then “ <i>localAreaDensity</i> ” must be less than 0, and vice versa.
<i>potentialRadius</i>	This parameter determines the extent of the input that each column can potentially be connected to. A large enough value will result in 'global coverage', meaning that each column can potentially be connected to every input bit. This

	parameter defines a square (or hyper square) area: a column will have a max square potential pool with sides of length $2 * \text{potentialRadius} + 1$ . Default ``16``.
<i>stimulusThreshold</i>	This is a number specifying the minimum number of synapses that must be "ON" in order for a column to turn ON. The purpose of this is to prevent noise input from activating columns. Specified as a percent of a fully-grown synapse. Default ``0``.

### 3.3.2.4 Learning

In this phase, the permanence values of only the winning column's synapses as well as their boost values and inhibition radius will be updated if it is necessary. The main learning rule is that permanence value of winning columns will be incremented if their synapse is active otherwise it is decremented. In this phase the *boostFunction* will be computed which includes two mechanisms to help a column in learning the connections.

- The boost value of a column will be increased if the column is not in an "active" state often enough compared to its neighbours, this is measured by "*activeDutyCycle*" and "*overlapDutyCycle*" functions,
- The boost value of a column will be set to be less than 1 if a column is active more frequently than its neighbours.

Table 3-3 provides more details on the parameters are used in the SP algorithm for learning.

Table 3-3 Parameters are used in the SP algorithm for learning

Parameter	Description
<i>synPermActiveInc</i>	Amount permanence values of active synapses are incremented during learning phase and it is somewhat data dependent.
<i>SynPermInactiveDec</i>	Amount permanence values of inactive synapses are decremented during learning phase. Its value should be less than the “synPermActiveInc”.
<i>boostStrength</i>	A number greater or equal than 0.0, used to control the strength of boosting. No boosting is applied if it is set to 0. Boosting strength increases as a function of “boostStrength”. Boosting encourages columns to have similar “activeDutyCycles” as their neighbours, which will lead to more efficient use of columns. However, too much boosting may also lead to instability of SP outputs. Default “0.0”.

### 3.3.3 Temporal Memory

The Temporal Memory (TM) algorithm learns transition of patterns, by recognising sequences of spatial patterns over time. As discussed beforehand, output of the SP is an SDR that represents active columns of cells that received the most input, and the resolution of the TM algorithm is to produce representations of temporal sequences at level of individual cells within the active columns.

In the TM algorithm, second form of SDR will be created (Figure 3-11) by activating a subset of cells within every active column (winning column). This enable an HTM region to represent the same input in many different contexts (Ahmad and Lewis, 2017). Learning in the TM algorithm is similar to SP algorithm, in both cases learning involves establishing connections by incrementing and decrementing the permanence values of potential synapses on a dendrite segment. The idea of making synapses more or less permanent is coming from “Hebbian” learning rules, where the permanence value of the synapses will be incremented if the synapses are active that contribute to the cell’s activation, otherwise their permanence value will be decremented which means

they did not contribute to the cell's activation, this process is also referred to reinforcement of the dendritic segment in (Cui, Ahmad and Hawkins, 2016).

In the TM algorithm, when a cell becomes active, it forms a connection to other cells that were active just prior through their distal synapses, hence cells can predict when they become active by looking at their connections. If all the cells do this, collectively they can store and recall sequences, and they can predict what is likely to happen next.

HTM cells at any point at time can be in one of three states. A cell when is active due to feed-forward input (through proximal dendrite), it is in "*active*" state. A cell when is active due to lateral connections to other nearby cells (through distal dendrite), it is in the "*predictive*" state, otherwise it is in "*Inactive*" state.

The pseudocode for the TM algorithm is documented in Appendix A. The TM algorithm computes two primary steps as follows:

1. Identify which cells within active columns will become active on this time-stamp.

A column when becomes active due to a feed-forward input, it looks at all the cells in the column, and one or more cell will become active if they are in the predictive state. In cases where no cells in the column are in the predictive state, then all the cells become active, marking the column as "*bursting*". In active columns, cells in the predictive state marked as a "*winner*", however when a column is activated and cells within the column are bursting, still the winning cell needs to be selected to make it presynaptic candidates for synapse growth in the next time step (Hawkins, Ahmad and Cui, 2017). This is done in following sub steps:

- It searches for near matches, this means that it searches through the cells that any of them potentially could be in a predictive state. This means that they have distal segments that match the previously active cells but their permeance values of synapses are not high enough to construct a connection. If they have been connected, then they would have been in a predictive state and column would not have burst. The cell with near

matches properties will be selected as winner cell which become predicted cell in the next time step.

- If the cells within a bursting column are bursting and they do not have any segments that matches any previous active cells, all the cells with the column will be reviewed and a cell with the fewest number of segments will be selected as a winner cell. This makes sure that cells are utilised properly by not overloading cells with context information which are not necessarily (Ahmad and Lewis, 2017). Next is to create new segments and synapses to the previous active cells. This is done by creating new distal segments and increasing permanence value of distal synapses between the burst cells and previously active cell followed by decreasing the permanence value of distal synapses on inactive cells.

## 2. Identify a set of cells to put into a predictive state

The TM algorithm in this phase, after identifying the current active cells, makes a prediction of what is likely to happen next, by depolarising all the cells that will likely become active due to future feed-forward input. This is done by looking at each cell within the structure and calculating a number of connected distal segments and each synapse on the segment that correspond to current active cells. The number of synapses that are connected to the active cell when exceeds a threshold then that cell's state will be changed to predictive and it will be primed to become active in next time step.

### 3.3.4 Classifiers

There are several classification algorithms that are available through NuPIC for classification and prediction tasks, including: SDR classifier, CLA classifier and KNN classifier. In this section, the process of classification and prediction for each classifier is discussed. The prediction task in HTM can be done on a single field or multiple fields.

### 3.3.4.1 SDR Classifier

The SDR classifier is a variation of the previous CLA classifier, it accepts an SDR output from the TM algorithm also known as activation pattern, and information from the encoders (“classification”) that describing the true input (target), shown in Figure 3-13. The SDR classifier learns associations between a given state of the temporal memory at time  $t$ , and the value of input at time  $t + n$  ( $n$  is number of steps into the future used to predict). In other words, it maps activation patterns SDR (vector of Temporal Memory’s active cells) to probability distributions (for the possible encoder buckets). The SDR classifier accomplish this by implementing a single layer, feedforward, neural network.

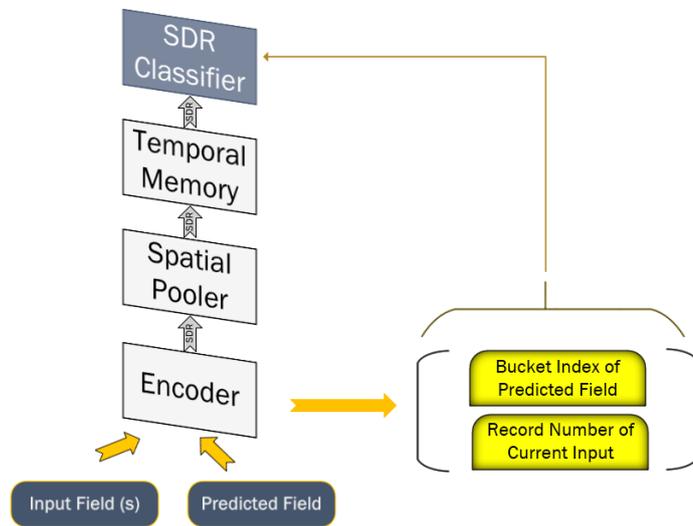


Figure 3-13 Input data to SDR Classifier

The SDR classification model consists of three phases including:

- *Initialisation*: where the weight matrix is initialized with zeros, implying that all classes occur with equal probability before learning.
- *Inference*: compute a probability distribution by applying “Softmax” function to the activation levels, it calculates the predicted class probabilities for each input pattern (probability distribution)
- *Learning*: adjusts the connection weight in proportional to the gradient, computing error scores for each of the output units and adding them to the appropriate weight matrix element. The SDR classifier has a same behaviour as CLA classifier in learning phase, when a predicted field has

a scalar value, keep a rolling average of actual values that correspond to each bucket.

Figure 3-14, shows the feedforward classification network that is used in SDR classifier for the prediction task (Cui, Ahmad and Hawkins, 2017). It classifies a single SDR by mapping a sequence of N-dimensional SDRs ( $x_1, x_2, x_3, x_n$ ) to probabilistic distribution over set of  $K$  class labels ( $y_1, y_2, y_3, y_k$ ) such that the target class labels  $z'$  is well predicted (with high probability). The softmax layer the is output of the classifier.

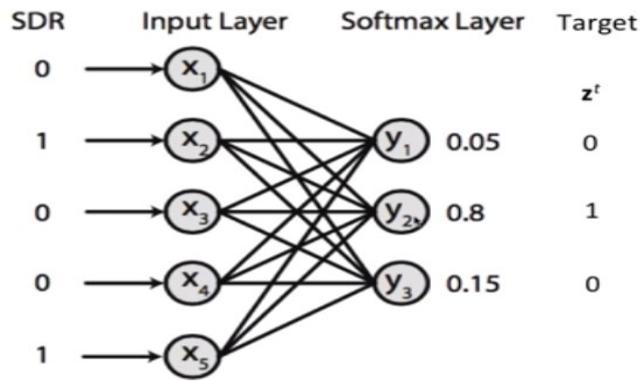


Figure 3-14 Feedforward Classification Network in SDR Classifier (Bobak, 2017)

The SDR classifier has a number of steps and mathematical computations to do the classification and prediction, including:

- *Weighting*, each class labels ( $y_1, y_2, y_3, y_k$ ) receives a weighted sum of the SDR's input bits ( $x_1, x_2, x_3, x_n$ ) every iteration, in the weighting phase, the value of each input bit is scaled (multiplied) by some scalar value and there is only require to know the weight of the active bits. Hence, the weights are organised by row (the output "y" they belong to) and the column (the input "x" they apply to) these weight values can be changed over time as the SDR classifier learns.
- *Summing*, is done to add together all of the weighted "x" inputs for each "y" output to determine their activation level using following equation:

$$a_j = \sum_{i=0}^N W_{ij} x_i$$

Where,

- $a_j$ , is the activation level of the  $j^{th}$  output.
- $N$ , is the number of inputs,
- $W_{ij}$ , is the weight that the  $j^{th}$  output is using for the  $i^{th}$  inputs ( $i$  is the column in the weight matrix, and  $j$  is the row)
- $x_i$  is the state of the  $i^{th}$  inputs (either 1 or 0)

In weighting and summing steps, we compare the activation level of the output by weighting each of the inputs and then adding them all together.

- Softmaxing the activation levels, “Softmax” function is an additional non-linearity function that it does the normalisation to make sure that the prediction probability sums to one by taking the exponential of each input and then divided by all other inputs. The “Softmax” formula is:

$$y_k = \frac{e^{a_k}}{\sum_{i=1}^k e^{a_i}}$$

Where,

- $y_k$ , is a probability of seeing the  $k^{th}$  class a particular number of steps into the future
- $e^{a_k}$ , is base  $e$  raised to the activation level of the  $k^{th}$  output.
- $k$  is the number of possible classes (classification of the input)
- $\sum_{i=1}^k e^{a_i}$ , is the sum of base  $e$  raised to the activation level of each output.

#### 3.3.4.2 CLA Classifier

CLA classifier was introduced as an alternative to Reconstruction algorithm and it has shown a better result. As shown in Figure 3-15, the CLA classifier is similar to SDR classifier in collecting a binary input from the TM algorithm in HTM system, and information from encoders (“classification”).

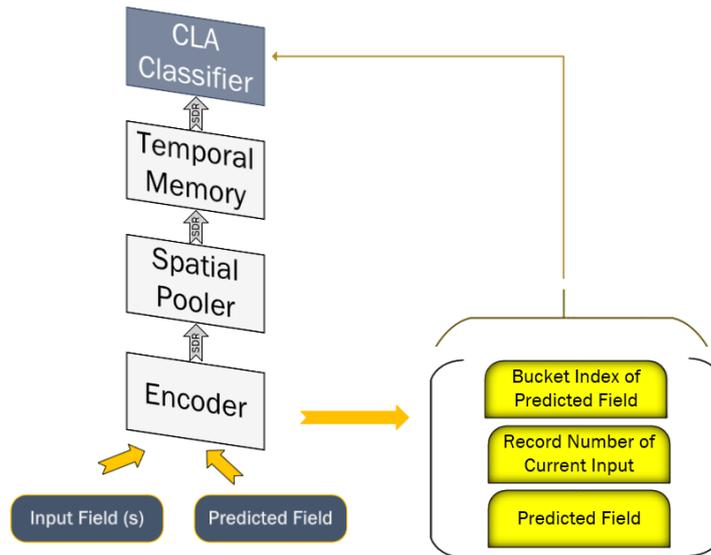


Figure 3-15 Input data to CLA Classifier

As we discussed in Section 3.3.3, the temporal memory makes predictions constantly, but we are using them in the sequence and not to make the prediction. This means that CLA classifier does not make the prediction in the sense like the temporal memory is making prediction of the next step, the CLA is a classifier classifying some future value of the current state with some future value that will be inputted. The intuition behind this is that the state of the temporal memory at any point of time that active cells are, is all the knowledge that we could possibly have about world in the past and the present (a representation of sequence of the patterns will coming in the context of previous learning). Therefore, CLA classifier uses the SDR output from the TM algorithm and it generates predictions by classifying and map ping the SDR to some values of the future data. To do this, the CLA learns a function of an SDR at time  $t$  ( $SDR_t$ ) and then the classifier produces a probability distribution over the predicted filed (PF),  $k$  steps into the future using following formula:  $f(SDR_t) \rightarrow P(PF_{t+k})$ . Afterword, the CLA classifier when computes the probability distribution, for each predicted step ( $k$ ), it then maintains and updates a mapping list of last  $k$  SDRs to current predicted field at time  $t$ :

$$f(SDR_{t-k}) \rightarrow PF_t$$

Where,

T, is refer to the time

PF, is a predicted field,

k, is steps into the future

The CLA classifier during learning phase, for every bit in activation pattern, it records a history of the classification each time that bit is active, the history is weighted so that more recent activity has a bigger impact than older activity. One important parameter that is used during learning is called “alpha” parameter that controls this weighting by adapting the weight matrix during learning. The CLA classifier during inference, it takes an ensemble approach, for every active bit in the “Activation Pattern”, it looks up the most likely classification(s) from the history stored for that bit and then votes across these to get the resulting classification(s).

### 3.3.5 Anomaly Detection

HTM anomaly detection, can be used to create a streaming anomaly detection system, it performs very well across a wide range of data sources and it is an open source and commercially deployable. Some of the applications that HTM anomaly detection can be deployed are such as: monitoring IT infrastructure, real-time health monitoring, tracking vehicles, and monitoring energy consumption. In following subsections, two types of anomaly detection models, including their detection techniques are discussed.

#### 3.3.5.1 Anomaly Score

Anomaly detection in HTM is done by computing the anomaly score. The anomaly score enables the HTM to provide a metric for representing the degree to which each record is predictable, where each record can have an anomaly score between ‘0’ and ‘1’ which ‘0’ represents a completely predicted value whereas a ‘1’ represents a completely anomalous value. The anomaly score feature in HTM is implemented on top of the core SP and TM algorithm, and it doesn’t require any changes to them.

Two models of anomaly detection known as temporal anomaly model and non-temporal model which are describe below:

- Temporal Anomaly model

This model is currently an approved and recommended model to be used for anomaly detection and to report the anomaly score. To compute the anomaly score, it uses the temporal memory to detect novel points in sequences and it calculates the anomaly score based on correctness of the previous prediction. This is done by calculating the percentage of active spatial pooler columns that were incorrectly predicted by temporal memory.

The algorithm for the anomaly score is as below, it describes that raw anomaly score is the function of active columns that were not predicted:

$$anomalyScore = \frac{|A_t - (P_{t-1} \cap A_t)|}{|A_t|}$$

$P_{t-1}$  = Predicted Columns at time t

$A_t$  = Active Columns at time t

- Non-Temporal model

This model is not recommended to be used for anomaly score, however, the idea was to use this model for adding anomaly detection that are “non-temporal” and are described as a combination of fields that does not usually occur, independent of the history of the data.

### 3.3.5.2 Anomaly Likelihood

Anomaly scores are used and post processed to generate anomaly likelihood values (Figure 3-16),  $\mathcal{X}_t$  is a stream data that is encoded into a sparse high dimensional vector  $a(\mathcal{X}_t)$ , and a sparse vector  $\pi(\mathcal{X}_t)$  represents a prediction of an input by HTM in next time stamp ( $\mathcal{X}_{t+1}$ ).

Anomaly score ( $S_t$ ) when is computed, the distributions of anomaly scores are analysed by Anomaly likelihood. The likelihood estimates the historical distribution of anomaly scores and likelihood of the current predictability rate by checking if the recent scores are very different.

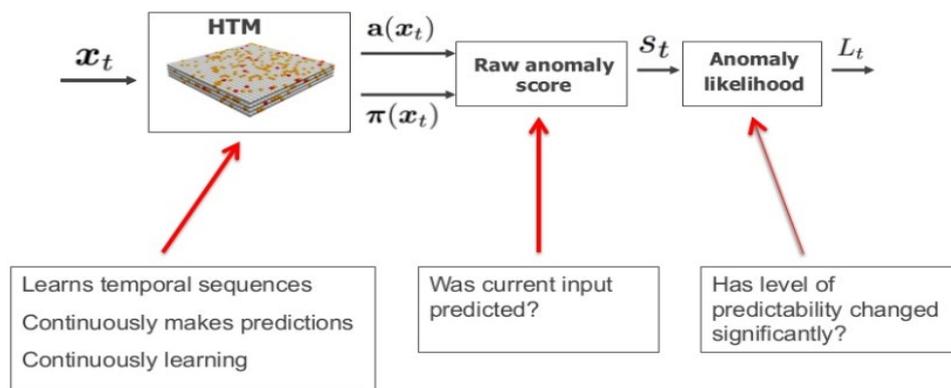


Figure 3-16 The primary functional steps for Anomaly Score and Anomaly Likelihood Process (Bobak, 2017)

### 3.4 HTM Applications

Melis, Chizuwa and Kameyama (2009) compare the implementation of a cellular phone interaction estimation example on HTM and a Bayesian Network (BN) algorithms. The focus of this work was to predict the user interaction during the use of a cell phone based on the sequence of menu choices that a user selects while using the phone. It is suggested by the authors that BN is designed from a much higher level and requires more interaction from the user besides the needs for some pre-processing of the input data from the real world. On the other hand, HTM takes in information from the real world, and the structure of HTM has an impact on its accuracy. Hence, the structure of the application when is more reflected in the HTM system, then even better accuracy can easily be obtained. An interesting fact about this work is that HTM performs well in a scenario where only the sequential information is produced. This also indicates that the SP algorithm in the HTM system can be only activated to deal with the spatial patterns in scenarios where temporal patterns are not captured by a system.

(Mitri *et al.*, 2017) applied HTM and a heuristically driven batch detector for comparing their anomaly detection accuracy. This work is motivated by comparing the accuracy of these two anomaly detection approaches to Continuous Positive Airway Pressure (CPAP) data, hence, to assist physicians in monitoring the breathing patterns and diagnosis of interruption of breathing. The authors of this work highlighted that as HTM attempts to provide a general fixed size models that are independent from domain knowledge cannot perform well in CPAP scenario. Hence, there is need for a custom detector that operates on a frame by frame basis where frame of a predefined size is processed and labelled as normal and abnormal accordingly. Figure 3-17, presents the flowchart of a custom model is used to model a normal and abnormal breathing patterns by assumption that regular and health breathing patterns are marked by periodic patterns. These patterns can change their threshold or offset but their repeatability is maintained throughout.

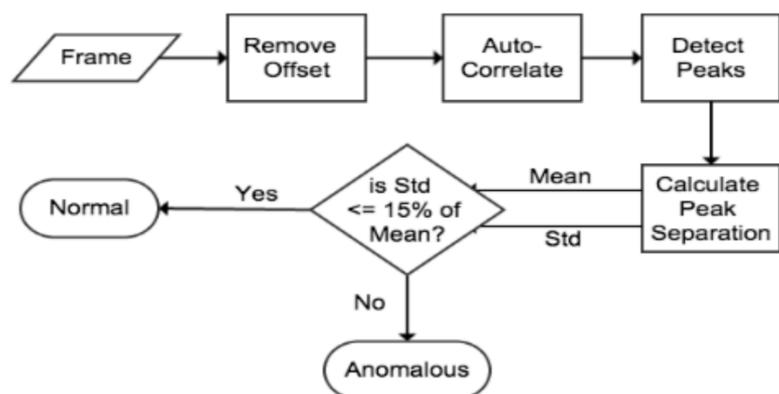


Figure 3-17 Flowchart of custom detector (Mitri *et al.*, 2017)

The result shows that the HTM does not yield the same result for every run, due to random initialization of the HTM. The custom detector on the other hand has bad precision due to its rigidity in detecting periodic patterns.

Rodriguez, Kotagiri and Buyya (2018) proposed a use of HTM networks to detect performance anomalies in the execution of scientific workflows in distributed computing environment such as high-performance computers and cloud environments. The proposed framework detects performance anomalies of application that are caused by the factors such as failures and resource contention which may lead further to lengthy delays in workflow runtime or unnecessary

costs. The proposed framework to detect performance anomalies it analyses the time series data that contain the resource consumption details (i.e. CPU and I/O) of tasks at different stages of their execution. Figure 3-18 presents a high-level of a performance anomaly detection framework is used in Workflow Management System (WMS) to efficiently automate the execution of scientific workflows and reduce their effects on Quality-of-Service (QoS) requirements.

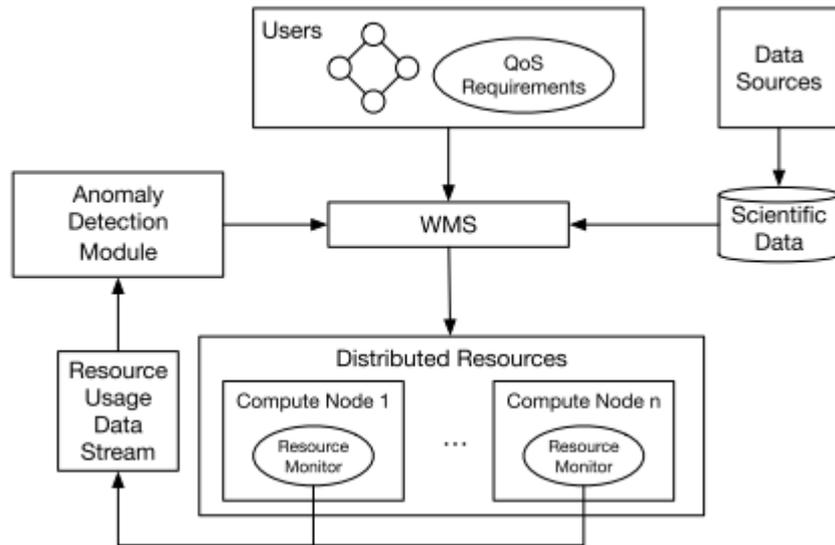


Figure 3-18 High-level overview of a performance anomaly detection framework for scientific workflows (Rodriguez, Kotagiri and Buyya, 2018)

An interesting point is highlighted in this work is that the proposed framework consists of a single HTM model per resource consumption metric including CPU and I/O for each task. The authors believe that combining multiple metrics in a single model does not improve the ability of the system to detect anomalies but rather, has opposite effect. However, we believe that if the output of each single HTM model per resource consumption metric will be combined and feed into a third HTM model could be used to capture correlations between two metrics hence improving performance anomaly detection of the proposed framework.

### 3.5 Summary

In this Chapter, the HTM, a sophisticated form of neural network is presented, and the CLAs including SP and TM are implemented in accordance with the hierarchy defined in the HTM theory. The hierarchy of neurons in the HTM theory are arranged in columns, mini columns, within several layers and regions.

In the HTM theory, the neuron model includes active dendrites as well as proximal, distal and apical dendrites which mimics more properties of the biological neuron model in comparison to the neuron model in traditional artificial neural network.

The HTM essentially is a system based on a memory system similar to the brain, it is trained on time series data (inherently time based) and relies on learning a set of sequences and sequences of patterns. Information in the HTM is always stored in a distributed fashion, and the HTM can be modelled as long as the key functions of hierarchy, time and SDR of information are incorporated.

The HTM theory uses the CLAs to model the spatial patterns which is a blend of information bits that happen together regularly, and after that it looks into time-based patterns or sequences for how these spatial patterns show up. This is done by mimicking the neocortex of the human brain involved in high-order brain function to recognise and predict ordered temporal sequences.

The main components of the CLA are discussed including the encoder, the SP algorithm and TM algorithm. Furthermore, several encoders are represented, and it is identified that different encoders can be selected based on the type of the raw data (e.g. numerical, categorical, date time).

This Chapter has facilitated the researcher to understand the fundamental components of HTM theory and CLAs algorithms. The interconnection among different regions and layers within HTM hierarchy also influences the data processing and modelling to detect and predict anomalies. The collected information has great use in order to propose a novel framework applying the HTM theory. The proposed framework which will be discussed in next Chapter will aim to construct and combine multiple HMT models and structure them in a hierarchy to improve performance of existing HTM algorithms.

## Chapter 4

# PROPOSED FRAMEWORK

In this Chapter, a novel and adaptive framework is implemented for a health monitoring and assistance scenario. The proposed framework is inspired by the HTM theory, and it is built on the CLAs. The novel framework is implemented to monitor the health condition of an older patient by detecting abnormal patterns in several vital signs. The proposed two-layer framework incorporates an online learning algorithm that can learn normal patterns of vital signs and use this knowledge to detect abnormal patterns. The detection of the abnormal patterns in vital signs is essential in the health monitoring and assistance scenario and is driven by improving care and survival of the elderly patients who are living independently.

The structure of this Chapter is organised as follows. First, the key challenges and limitations for anomaly detection used in the AAL health monitoring application are identified and accordingly, a scenario is created in order to develop the proposed framework. Afterwards, the temporal change in vital signs for a 24 hours interval is discussed, followed by a discussion on threshold for vital signs subject to several medical conditions. Next, temporal and spatial dependencies among the monitored vital sign values that lead to a cardiac arrest event are explored. This knowledge is essential to be highlighted as the changes in such

vital signs are highly correlated and they are also patient-specific which can be either repeated or can happen at a different time.

Lastly, the architecture and hierarchy of regions in the proposed framework will be demonstrated by referring to the HTM theory and CLAs that consider the spatio temporal features of input data. The pseudocode and software codes for proposed model are documented in Appendix B.

## 4.1 Challenges

The key challenges and limitations identified during this study are summarised in this section.

Health professionals commonly monitoring the vital signs of patients for diagnosis and prognosis of medical conditions. In recent years, several ML approaches are also attempted in AAL applications for detecting anomalous patterns in single or multiple vital signs including the BP, HR and SpO<sub>2</sub>. However, at the time of developing the proposed framework, there is limited existing works using the online learning approach to model the spatiotemporal patterns of multiple vital signs in elderly people. Besides, there is not any work found that applies the HTM theory to the proposed scenario (Section 4.2).

A core characteristic of the AAL health monitoring application is that raw data (vital sign) from patients are measured in real time, hence, dealing with streams of biosensor data has its own problems and challenges as well as analysing infrequent measurements.

The decision making in the AAL health monitoring application requires a strong modelling and inferring approach with a suitable handling of contextual information. Therefore, it is also important to include contextual information about a patient, such as gender, age and medical history.

Since, the AAL health monitoring application must have robust decision making, the number of considered vital signs has a significant role in improving the results. Hence, temporal and correlation patterns among different vital signs should be carefully considered in the design and implementation of the proposed

framework. Therefore, one of the challenges is to extract more semantical information from each vital sign to support a global reasoning.

Another challenge is that most applications consider the monitoring in clinical contexts therefore it is of importance for the proposed framework to be examined targeting patient groups such as an elderly person in AAL scenario.

Furthermore, the early acknowledgement of inevitable cardiac arrest is a standout amongst the most critical variables for survival. In the AAL health monitoring scenario, the time points of disturbance of individuals must be elucidated with the aim of creating explicit rules for the early recognition of approaching cardiac arrest. Hence, the exact and early expectation of wellbeing related abnormalities is an essential function of the proposed framework.

## 4.2 Model Scenario

This section describes the overall scenario for the model, the terminologies and the concepts used in the design and development of the proposed framework.

The concept of the proposed framework is visualized in Figure 4-1. In the development of the proposed framework, an AAL environment in particular health monitoring scenario is considered where an elderly person lives at his/her home independently and has access to healthcare services. The elderly person wears a biosensor device on his/her wrist that collects information on different vital signs. In this work, four vital signs are collected to train the proposed framework, including Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP), Heart Rate (HR) and Blood Oxygen Saturation (SpO<sub>2</sub>). Pervasive technologies are used to automatically collect and transfer the vital signs data to a server where the health condition of the elderly person can be analysed and processed.

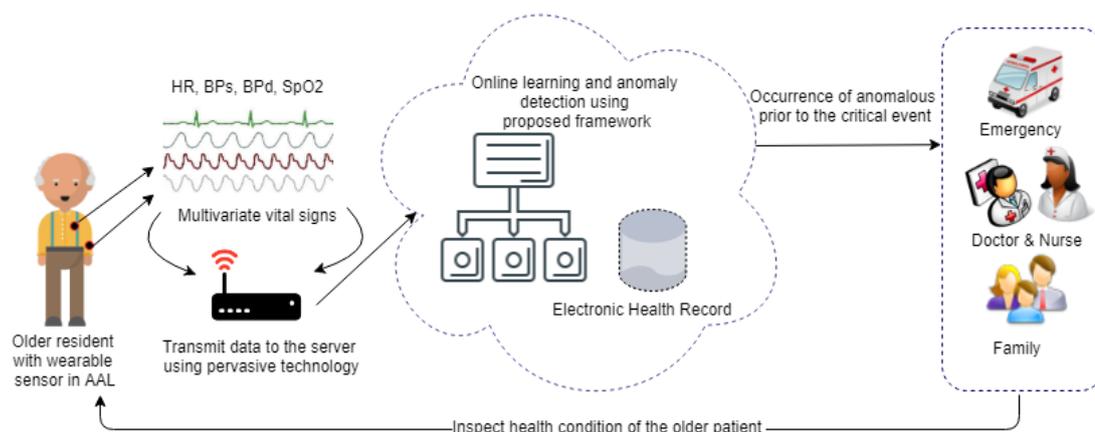


Figure 4-1 Scenario for the proposed framework

The professional caregiver and family members receive information related to the elderly person via a User Interface (UI) on his/her portable device (mobile or wearable) or desktop. The caregiver is required to infer from this information whether the elderly person needs an urgent care. Moreover, the caregiver will automatically receive an alert on his/her device when the proposed framework detects an abnormal pattern in vital signs. The abnormal patterns in vital signs must be detected by the proposed framework at least 10 hours prior to cardiac arrest, therefore the proposed framework can proactively alert a caregiver and family member to abnormal changes before the situation becomes critical. The proposed framework will be able to detect future critical events from observations of temporal sequences of vital sign values.

Under such a scenario, the proposed framework should be proactive by having the following characteristics:

- To have online capability to learn and adapt to new normal patterns.
- Abstract the semantical information from the collected data and learn temporal patterns and correlation between multiple vital signs.
- Apply contextual anomaly detection approach, to detect anomalous data points.
- Detect an anomalous point as early as possible, preferably 10 hours preceding cardiac arrest.

- Employ a common learning technique that can be used for clinical decision support to discover patient specific anomalies independently.

#### 4.2.1 Temporal Change Detection in Vital Signs

As discussed in Section 2.5, one significant problem in the TR paradigm is to decide what sort of information is subject to change in order to abstract meaningful patterns. These patterns are widely used in AAL environments for diagnosis and prognosis tasks. In order to abstract temporal patterns in vital signs, an appropriate approach must be selected first, and in this work, the period-based approach is used to abstract temporal patterns of the vital signs.

According to several clinical studies (Clement DL, De Buyzere M, De Bacquer DA, 2003; Oh, Lee and Seo, 2016; Forkan and Khalil, 2017a), the detectable changes in blood pressure appear 18-20 hours and become dramatic at 5-10 hours before cardiac arrest. There are also noticeable changes in heart rates that begin at 4 hours and become more prominent at 2 hours pre-arrest. These detectable changes in vital signs are used in this work to learn normal and abnormal patterns of the vital signs. The value of vital signs has a threshold which is usually used by professional caregivers to diagnose symptoms or to check if the patients have a medical condition.

In addition to the fact highlighted above, studies have also shown that vital signs such as BP and HR have a daily pattern. For instance, BP is normally lower at night during the sleep period and starts to rise a few hours prior to the wake period. The BP continues to rise during the day, usually peaking in the middle of the afternoon and then in the late afternoon and evening, it begins dropping again.

The 24-hour temporal patterns of vital signs are clearly illustrated in Figure 4-2, where bedtimes occurred between 9:30 PM and 12:00 AM, and wake times occurred between 5:00 AM and 8:00 PM. Figure 4-2, shows 24-hour pattern of three vital signs including SBP, DBP and HR of the same individuals in an ambulatory and in bed rest scenarios, where minimised physical activity is maintained. In both scenarios, the vital sign data have detectable changes, which

will be abstracted by the proposed framework to learn and differentiate between normal and abnormal data.

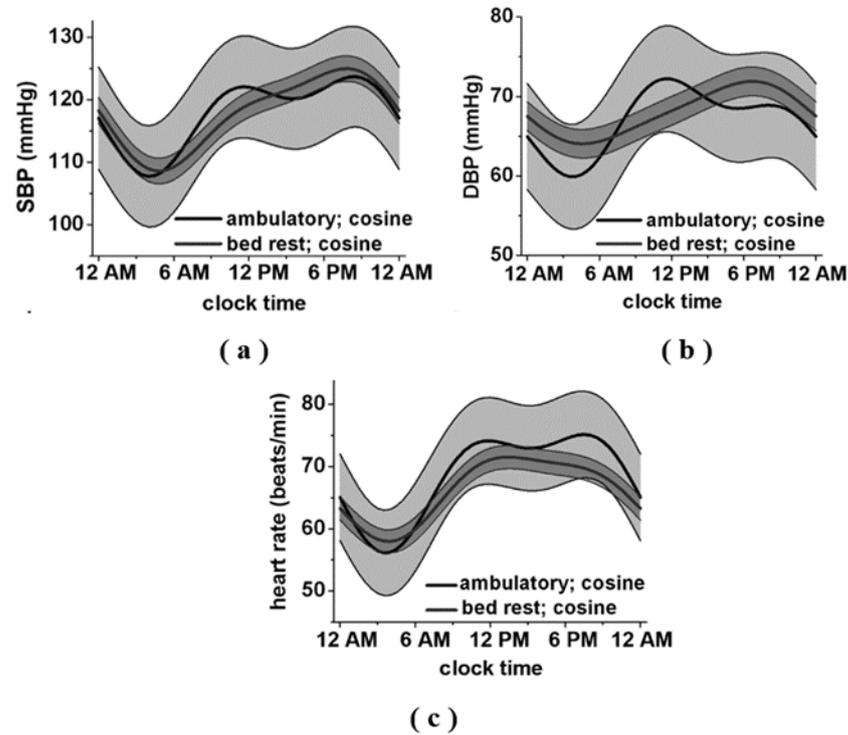


Figure 4-2 24-hour temporal patterns. (a) SBP, (b) DBP, and (c) HR (adapted from (Morris et al., 2013))

#### 4.2.2 Medical Condition

The normal range for the selected vital signs in this work, according to the medical rule are presented in Table 4-1. However, these thresholds can vary subject to risk factors such as age, gender, and medical conditions etc. According to Casiglia, Tikhonoff and Mazza (2005), both SBP and DBP progressively increase with age during childhood, adolescence and adult life. For instance, it is highlighted by the same work that after the age of 60, SBP increases. This shows that it is perhaps more realistic to accept that elderly people have their own normality that is not the same as that of other age groups.

Table 4-1 The vital signs and their generalised normal value

Vital sign	Acronym	Unit	Data type	Normal range
Systolic blood pressure	SBP	mmHg	Continuous, Temporal	90-120 mmHg
Diastolic blood pressure	DBP	mmHg	Continuous, Temporal	60-90 mmHg
Heart rate	HR	Beats per minute (bpm)	Continuous, Temporal	60-100 bpm
Blood O <sub>2</sub> saturation	SpO <sub>2</sub>	Percentage (%)	Continuous, Temporal	95%-100%

In addition to the age factor, the threshold values of vital signs for patients who have a medical condition, exceeding or falling behind the normal baseline thresholds, are described in Table 4-2. As shown in Table 4-2, a low SBP and DBP can be normal for a patient with a hypotensive condition, whereas it is actually abnormal for a healthy patient as described in Table 4-1.

Table 4-2 The vital signs value and medical condition

Medical condition	SBP	DBP	HR	SpO <sub>2</sub>
Hypertension	>140 mmHg	>90 mmHg	-	-
Hypotension	<90 mmHg	<60 mmHg	-	-
Tachycardia	-	-	>100 bpm	-
Bradycardia	-	-	<60 bpm	-
Hypoxemia	-	-	-	<80%

This necessitates the need for a personalized model for clinical decision support that can discover patient specific anomalies (Chester and Rudolph, 2011). In order to achieve this, the numerical thresholds of vital signs will be abstracted in this work to classify patients into different groups, subject to their medical conditions. Furthermore, the proposed framework will be trained and tested for patients with hypotension. Once the proposed framework is trained and tested in the context of patients with hypotension, then it can also be trained and tested for different contexts (e.g. hypertension, Tachycardia).

### 4.2.3 Correlation between Vital Signs

Normally a single vital sign does not contain enough information to be used for detecting anomalies. In order to overcome this issue, the correlation between multiple vital signs can be used. There are always strong positive or negative correlations in multiple vital signs that contain useful information for detecting anomalies. For instance, Figure 4-3 presents 24 hours of data collected from a patient with hypotension conditions, as illustrated, there are strong correlations between four selected vital signs including HR, SBP, DBS and SpO2. These correlations can be repetitive in some patients and vary over time in some patients. The proposed framework will perceive such changes in multiple vital signs to inform the health caregiver early enough to assist them in real-time to make proper clinical decisions. This is achievable by ensuring the proposed framework is fast, well trained and adaptive, hence less likely to make false predictions.

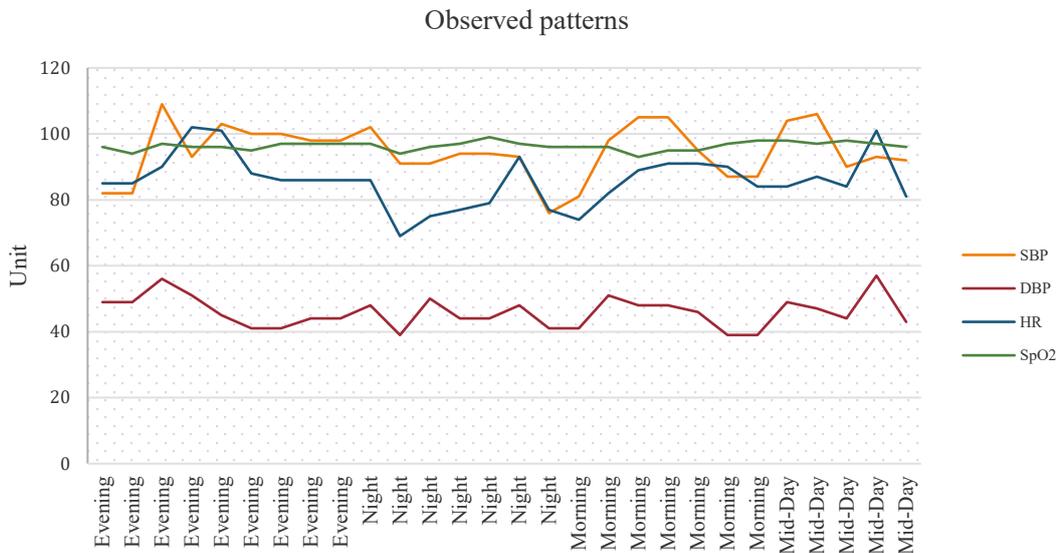


Figure 4-3 Time-series graph of 4 vital signs for an elderly patient with hypotension condition

### 4.3 Proposed HTM Framework

The proposed framework is inspired by the HTM network. As highlighted in Chapter 3, the HTM system is based on memory, and time is playing an important role as the supervisor for training the HTM to model temporal sequences. Information in the HTM memory is always stored in a distributed fashion, and the HTM can be modelled as long as the key functions of hierarchy, time and SDR are incorporated.

The proposed framework illustrated in Figure 4-4 is constructed as a two-level hierarchy, and each level in the hierarchy consists of several regions. According to Hawkins, Ahmad and Dubinsky (2011), an advantage of a multi levelled structure is efficiency as it significantly decreases training time and memory usage. This is because the patterns that are abstracted at each level of the hierarchy are reused when consolidated in novel ways at a higher level. In this work, a two-level hierarchy is used as information ascends the hierarchy, the output of regions in lower levels will converge, and the regions on the higher level of the hierarchy make use of the information to produce a bigger picture of the patient's health condition. This proposed architecture improves the learning and prediction of the proposed framework.

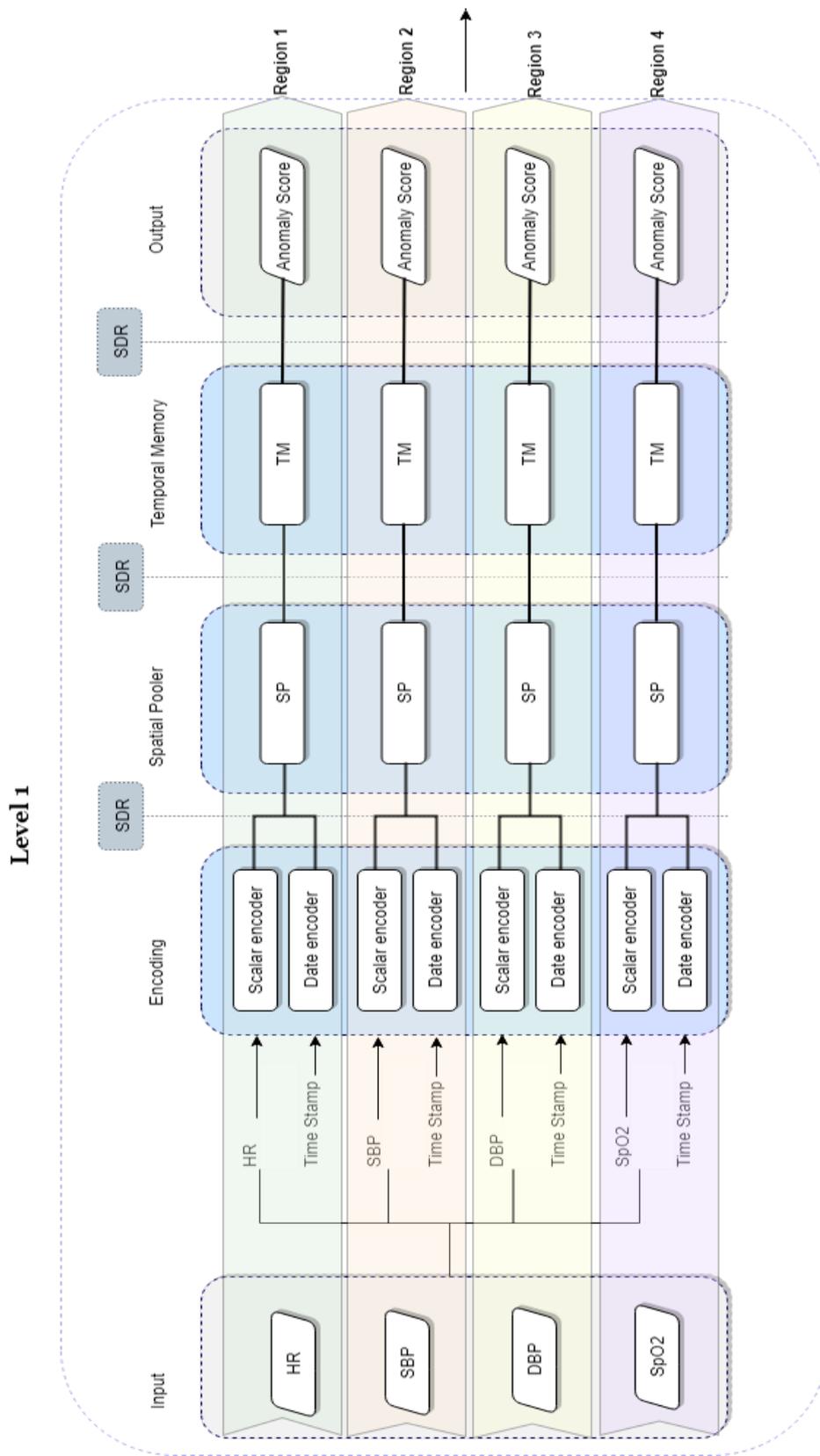
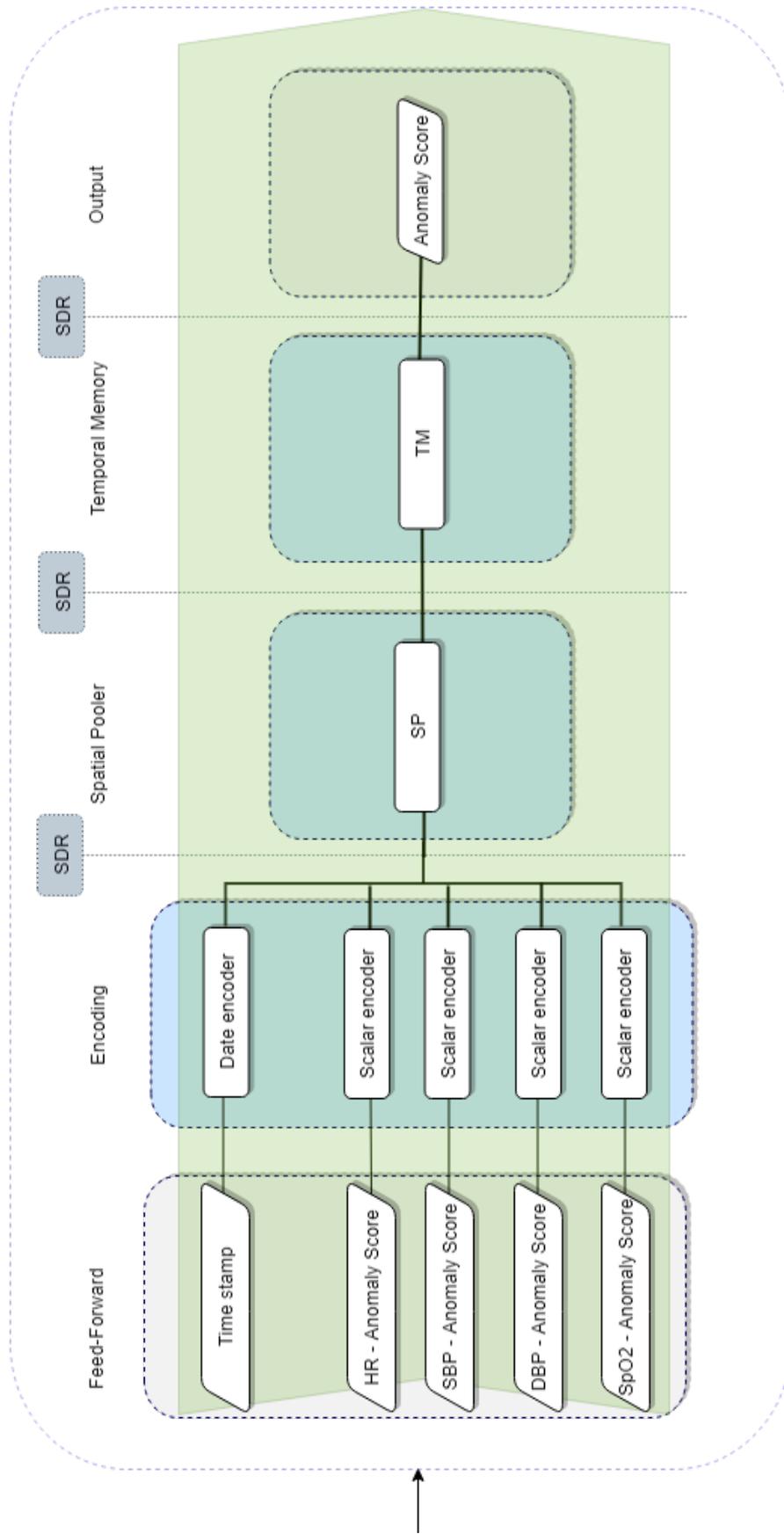


Figure 4-4 Proposed Framework

**Level 2**



### 4.3.1 Hierarchy

In an HTM hierarchy, information is communicated within levels, between levels, and to/from outside the hierarchy. In the proposed framework, this feature of the HTM hierarchy is used, where the vital signs data are inputted into level-one to extract some basic features of vital signs data. Then the semantically processed data from level-one are sent to level-two where more semantic features of input data are modelled. As shown in Figure 4-5, the hierarchy of regions in the proposed framework consists of two levels. In level-one, each region is responsible to learn the spatial and temporal patterns of an individual vital sign, hence in layer one, four regions process four different input streams to produce more meaningful information, which can be used to compute an anomaly score separately for each vital sign data.

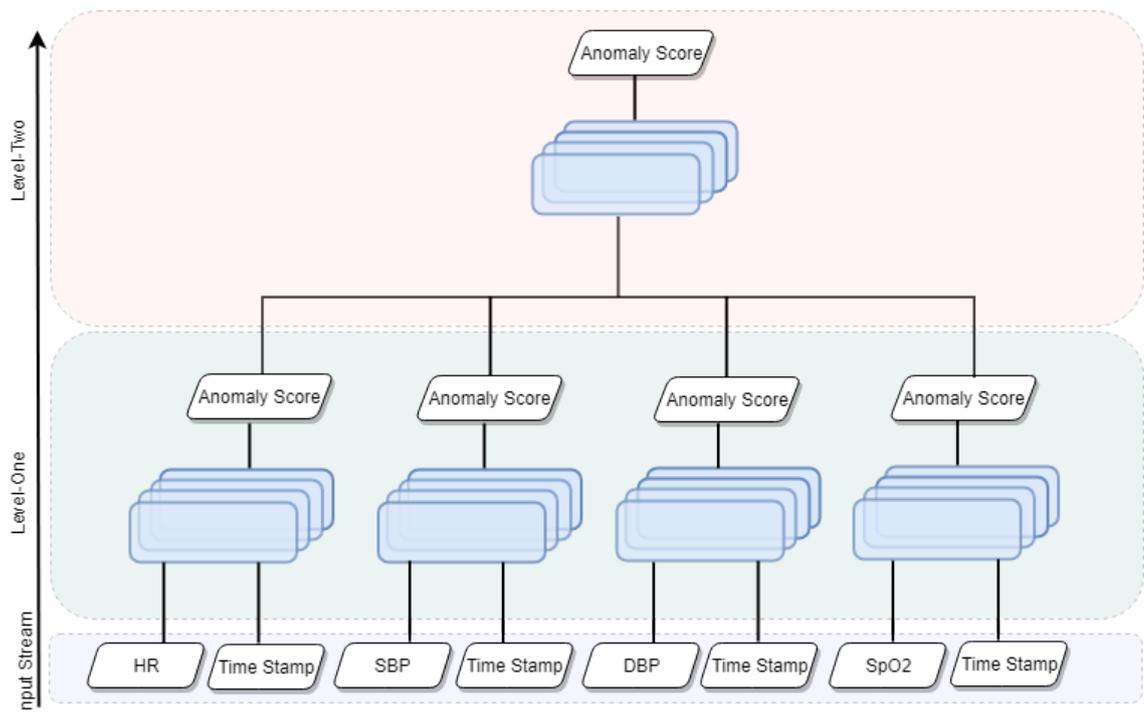


Figure 4-5 Hierarchy of regions in proposed framework

The output from level-one consists of anomaly scores that are computed for each vital sign. The anomaly scores from level-one will be feed-forwarded into level-two for further processing. The rationale for having anomaly scores for multiple

vital signs, as discussed in Section 4.2.3, is that a single vital sign does not contain sufficient information to be used for detecting anomalous points. However, there are always strong positive or negative correlations in multiple vital signs that contain useful information for detecting anomalies. Hence, in the proposed framework, the correlation properties between multiple vital signs is taken into consideration in level-two to improve the accuracy of the standard HTM system in detecting anomalies in AAL health monitoring. Level-two of the proposed framework is responsible for extracting the correlation patterns between vital signs. It consists of one region where its output will be a final anomaly score. The anomaly score computed in level-two will be used as a predictor to detect abnormal behaviour of vital signs prior to cardiac arrest.

Now that two levels of the hierarchy are discussed, the functions of the regions within these two levels will be discussed.

### 4.3.2 Regions

The regions in HTM are responsible for the given functionalities. In the proposed framework, each region consists of four functions including encoder, SP and TM and temporal anomaly function to provide a metric for representing the degree to which each record is predictable. The HTM model does not know what sources of information or what inputs are presented. The data stream, in this case, SBP, DBP, HR and SpO<sub>2</sub>, is encoded into SDRs that enable the proposed framework to search for spatial patterns which is a blend of information bits that happen together regularly (SP), and after that it looks at time-based patterns or sequences for how these spatial patterns show up (TM). The extracted data in the form of SDR then will be fed into the temporal anomaly score model to compute anomaly scores.

The rational explanation for allocating multiple regions in level-one is that in a standard HTM system, one region is allocated to encode multiple vital signs with a timestamp. A drawback of this approach is that the correlation between vital signs is not considered and the learning is carried out on sequences of concatenated values. Figure 4-6 illustrates a standard HTM region consists of the concatenated SDRs of multiple vital signs with a timestamp SDR prior to being

input to the SP and TM. The drawback of this approach is that the correlations between the vital signs will not be learned by SP and TM, instead the HTM will learn the spatial and temporal patterns of the concatenated SDR without knowing about the correlation between each encoded vital sign.



Figure 4-6 Standard HTM system - One region is allocated for decoding multiple vital signs

As illustrated in Figure 4-7, one core feature of level-one in the proposed framework is to overcome this drawback by allocating four regions. One important role of a region is to create an SDR for a single vital sign by encoding and concatenating of the individual vital sign with timestamp before it will be sent to the SP and TM.

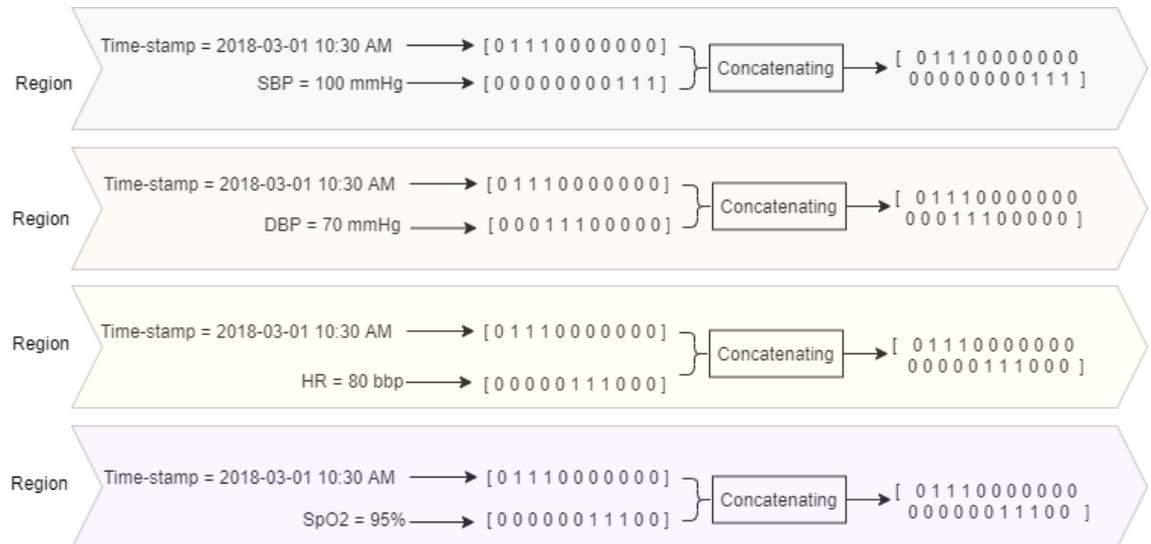


Figure 4-7 Proposed framework - multiple regions allocated for decoding multiple vital signs

The proposed design of regions allows every region to make predictions of the patterns for the individual vital sign by referring to what it has learned from a single vital sign. Every prediction that a region is making will be based on status

of the individual vital signs on the previous context. The fact that a region makes a prediction grounded on historical data as well as what is occurring now, using a single region for an individual vital sign can improve the accuracy of the HTM system. Hence, the regions in the proposed framework can act as a detector, as they can predict what will be the next value of individual vital signs, that it can learn new patterns and know when an anomaly has occurred.

Discussion on performance of the standard HTM system and comparison of its accuracy with the proposed framework are detailed and documented in Chapter 6.

#### 4.3.2.1 Encoding

As discussed in Section 3.2.1.3, the SDR is a primary data structure in the HTM systems, hence the HTM systems reliance on the SDRs. Therefore, an encoder is to convert the stream data into SDR that represents dynamic state of information in a region. The process is similar to a human retina when it receives external signals and human brain converts the signal to binary neural representation. The requirement of the encoder is to change over the vital signs data to the SDR that can figuring out which output bits must be ones, and which must be zeros for that input so that catch semantic significance of the individual vital sign data.

On the other hand, according to (Hawkins and Ahmad, 2016; Wu, Zeng and Yan, 2016), the time dimension is playing an important role as a supervisor for training the HTM to learn the temporal sequences. The vital signs in an AAL environment are measured at different time instances (Batal *et al.*, 2011), and they are recorded by adding a timestamp with the vital signs to be used for diagnosis and prognosis purposes. The timestamp can come in different formats. For instance, the timestamp may indicate date and time in which a data point (vital sign) is recorded, however based on the requirements of the proposed framework, more semantic information is required to be abstracted from time-stamps. For instance, knowing which time of the day the vital signs were recorded. The rationale behind this, as discussed in Section 4.2.1., is that human body functioning relies on the 24-hour environmental light/dark cycles or Circadian Activity Rhythm (CAR) (Liu and Chang, 2017), that results in vital signs having

different patterns during a day. Hence, when detecting an anomaly in AAL health monitoring, a data point on its own is not considered as anomalous but, when it is analysed in a specific context then it is regarded as anomalous. For instance, if two measurements of BPS at morning and night are the same (e.g. 100 mmHg), the value of SBP at morning or night may still be considered as an anomaly subject to the time of day that the SBP is measured (different context). Therefore, the time dimension for vital signs is extracted to give a different semantical meaning of similar data points that occur at a different time of day. To achieve this, each region in the proposed framework consists of two types of the encoder. An adaptive scalar encoder is used for encoding the vital sign data, and the Date encoder is used to convert the timestamp to the SDR. In order to encode time semantics, the continuous time encoding (timeOfDay) is used which allows encoding more semantics by converting time-stamp input stream into four categorical representations of time of day including: Morning, Mid-Day, Evening and Night. Figure 4-8 illustrates four categorical representations that are encoded into the same space of bits.

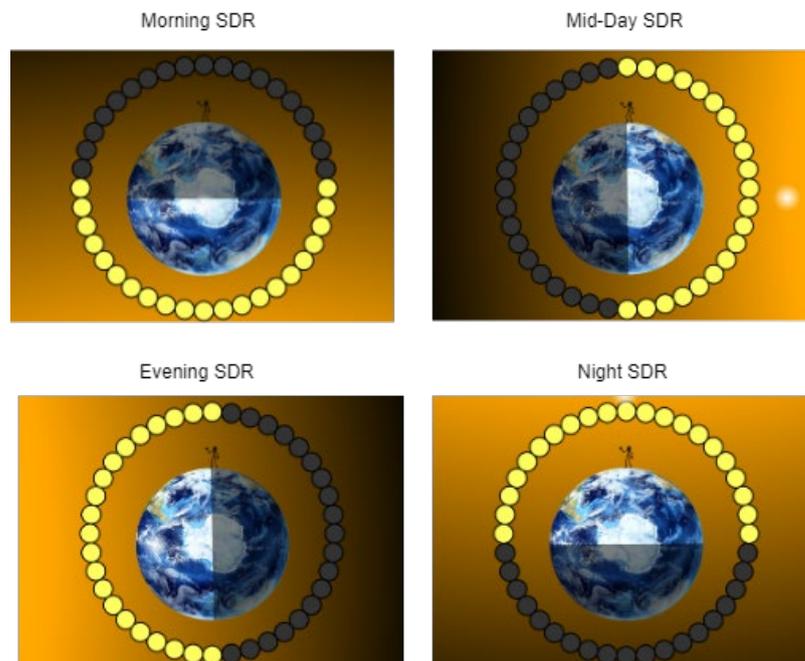


Figure 4-8 Sparse Distributed Representation of timestamp in proposed framework

The stream data including vital signs and timestamp when are encoded to SDR using appropriate encoders, then they will be forwarded to the SP where a fixed sparsity representation of the encoded data will be computed.

#### 4.3.2.2 Spatial Pooler

The SP algorithm is a core component of the HTM system, and its main goal is to create flexible and robust representations of input space. The SP enables HTM system to learn spatial patterns of input data by transforming the input SDR into an output SDR of different size with a fixed sparse number of activated columns (bits). It searches for input bits that often occur together in a continuous online fashion. Each active bit in the output SDR has semantic meaning. From the biological perspective, the neocortex represents information using sparsely distributed patterns of an activity. According to (Ahmad and Hawkins, 2016), the rationale behind sparsity is that at any point in time, only a small percentage of neurons are active while rest are inactive. The active neurons are known to be distributed in a region, and it is the pattern of the distributed active neurons that determines what is being represented.

The fixed sparsity property of the SP as discussed in Section 3.3.2, is achieved by applying the boosting and inhibition functions to select a sparse set of active columns. The sparsity of the SDR ensures all input patterns can be equally detected. For instance, when two different representations are fed into the SP over time which has a high overlap score, meaning they are semantically similar and the output data that the SP creates to represent those pieces of data also will have a high overlap score. Therefore, the SP creates two similar outputs when it gets two similar inputs, on the other hand it produces two very dissimilar outputs when it receives two dissimilar inputs. This might raise a question that what is the difference between input SDR and output SDR when both SDR using overlap function to represent similar and dissimilar inputs. The initiative in here as highlighted by (Cui, Ahmad and Hawkins, 2016), is that the SP by allocating a limited number of columns, it will ensure every column responds to a fraction of the input such that all columns participate in representing the input space while it captures the semantics. Figure 4-9, shows the SP forms SDR of input data with fixed sparsity close to 2% (bottom), despite the large variations of input sparsity (top) which it varies between 2% to 20%.

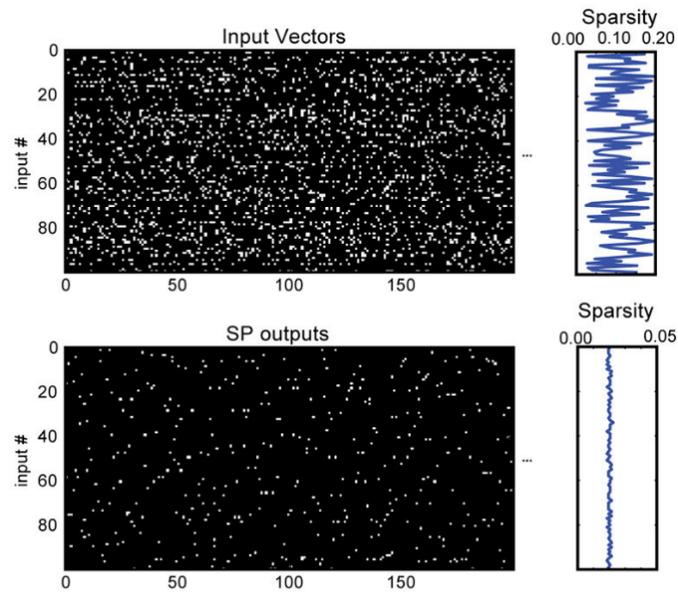


Figure 4-9 Example of sparsity ratio in input SDR and output SDR (adapted from (Cui, Ahmad and Hawkins, 2017))

One significant factor in here is to have a balanced level of activation densities. As highlighted in a work by (Cui, Ahmad and Hawkins, 2017), high activation density would more likely contributes to high false positives error whereas low activation density on the other hand can contributes to false negative error as it will be much harder to detect patterns with low activation densities.

In the proposed framework, the rational that each region consists of the SP is to contribute to one of the desirable properties of the SP, to be fault tolerant. The SP should continue to function in event of system faults such as loss of input or output neurons in the network. Therefore, allocation of the SP in each region to transform an individual vital sign input into an output SDR, it reduces amount of computation work that should be carried out by the SP in order to recover following a system fault. Hence, the SP in case of malfunctioning of a biosensor in AAL health monitoring when no values for a vital sign are measured, it only works on recovery following malfunctioning for that individual vital sign without disturbing performance of other SPs. The proposed approach also enables the proposed framework to function in event of data loss when at least data from one vital sign is available. This might affect accuracy of the anomaly detection in a short interval, however, it will be essential for the proposed framework to

function until effected regions recovery is accomplished. Furthermore, it also improves the performance of the SP to be robust to noise in the input.

In addition to above points, the flexibility and adaptability of the proposed framework to changing input statics are also considered which is particularly important for AAL health monitoring scenario with a continuous data stream that has changing statics. Regions in two-level hierarchy enables the SPs in level-one within the proposed framework to adapt to changes in individual vital sign by adjusting its synaptic connections. In level-two, the SP will create SDR of feed-forward from layer-one consists of anomaly scores of each vital sign to enable the TM in layer-two models the correlation between the vital signs.

In this section, spatial patterns are discussed and how the SP translates them from input space into a normalised representation with a fixed sparsity that still contains all the semantic information of original encoded data. In order to capture temporal patterns of input data and to model transition between the spatial patterns, the SDR created in SP will be sent to the TM.

#### 4.3.2.3 Temporal Memory

The core function of the TM is to learn sequences of spatial patterns over time. As dicussed in Section 3.2.1.4, the HTM system is based on memory, and time is playing an important role as the supervisor for training the HTM to learn the temporal sequences. The TM uses timestamp to learn and predict based on the knowledge learned previously.

The TM models the temporal patterns on input data by learning sequences of active columns from spatial pooler and predicting what spatial patterns is coming next based on the temporal context of each input. As discussed in Section 3.2.2, the TM works on cells within active columns and checks distal connection between cells to detect contextual patterns in order to predict the next feedforward input. For instance, a network where 2% of the columns are active, each column has 24 cells, and each cell recognises 100 patterns on its distal dendrites, can store approximately 120,000 transitions.

According to (Gustafsson–sebgu209, 2011; Hawkins and Ahmad, 2016; Wu, Zeng and Yan, 2016), learning and recognizing sequences is the basis of forming predictions where it uses high order sequences property. The high-order sequence memory paradigm enables the TM learns continuously, and to make multiple simultaneous predictions Hence, the TM can learn very long-range temporal dependencies and statistic for a particular input that can appear in different temporal contexts.

These features of the TM are used in the proposed framework where temporal patterns of each vital sign can be learned in level-one and are fed-forward to layer-two where temporal patterns of feed-forward will be modelled to learn correlation between vital signs. This enables the proposed framework to predict the value of vital signs in next timestamp. It is also more realistic to use a TM in each region within layer-one to capture the temporal patterns for each vital sign rather than using a series of concatenated multiple vital signs.

The prediction of the TM in level-two will be used to compute the anomaly score. This step provides a metric for representing the degree to which each record is predictable. Each record can have an anomaly score between ‘0’ and ‘1’ which ‘0’ represents a completely predicted value whereas a ‘1’ represents a completely anomalous value.

## **4.4 Summary**

This Chapter completed the development of a novel and adaptive framework by applying the HTM theory and ideas underlying combination of hierarchy networks in HTM. This Chapter covered the challenges in monitoring health condition of the elderly patient in AAL scenario to prevent critical events. Furthermore, the research challenges of modelling temporal and correlation patterns between vital signs are discussed. Finally, the hierarchy of the proposed framework including regions and core components of the proposed framework are detailed.

The main highlighting of this Chapter includes challenges that are tackled in this research. One of the challenges was to ensure online capability of the proposed

framework to learn and adapt to the new normal patterns, while abstracting the semantics of vital signs and correlation among them using TM and SP algorithms. The two layered hierarchy is used to detect abnormal patterns in vital signs by modelling temporal and correlation patterns between the vital signs. Multiple regions are constructed in level-one, to improve learning of temporal pattern for each vital sign rather than applying a series of concatenated multiple vital signs.

One of the core features of the proposed framework is to detect an anomalous point as early as possible (10 hours preceding cardiac arrest). This is achieved by applying the high order sequence memory which enables learning of very long-term temporal dependencies and statistic for a particular vital sign that can be appeared in different temporal contexts (time of day).

As a result, two-layered hierarchy framework consists of encoders, SP and TM is proposed to suite the requirements of the AAL in particular proposed scenario for health monitoring.

## Chapter 5

# MIMIC-III DATABASE

In order to train and test the performance of the proposed framework, the MIMIC-III, a clinical database is considered. In the ML literature, three data selection approaches typically are applied, including experimental wearable sensor data, medical databases, and simulated sensor data (Banaee, Ahmed and Loutfi, 2013). In the next section, several factors for the selection of the medical database are discussed.

The use of any data selection approaches, in particular, the medical database, requires processing of patient data before it is used in training and/or testing of the ML algorithms. This is due to a number of factors such as: data volume, incompleteness (missing data due to system failure or malfunctioning), incorrectness (noisy data due to vibration, humidity), inexactness (irregularity in data types, or/and unnecessary features) (McNaull *et al.*, 2012). The DM paradigm consists of several steps including, data pre-processing and processing. Nevertheless, DM steps and techniques that have been applied to the medical database are varied and subject to the nature and quality of the data. In this work, several DM techniques are used to cope with these obstacles (e.g. to cope with the missing values and preparation of input parameters/features) and querying the

medical database to look for the emerging patterns that shape the decision-making process of the proposed framework. In the next subsection, MIMIC-III database is compared with 3 other medical databases that are commonly used in ML literature.

## 5.1 Database selection

The Multiparameter Intelligent Monitoring in Intensive Care-III (MIMIC-III), a medical database, is considered in this work. Medical databases are most commonly used approach for data acquisition in ML algorithms in particularly for AAL health monitoring scenarios.

In recent years, due to emerging technologies, the hospitals are equipped with monitoring and several data collection devices which enable the hospitals to collect and store patients' data. Patients' records commonly are accessible in medical databases for diagnosis and prognosis, such databases typically include values of age, date of birth, gender, clinical and laboratory parameters as well as records of the medical condition(s). Moreover, databases used in a monitoring setting have additional characteristics, as they include the measurements of a set of parameters (e.g. vital signs) at different times, requesting the temporal component to be considered in DM and ML (Lavrač, 1999).

Several factors are defined based on the proposed model scenario and, in order to select a suitable medical database for the training and testing of the proposed framework.

These factors are listed below:

- Availability of clinical information, including age, gender, and health condition. It must also contain records of vital signs including SBP, DBP, HR and SpO<sub>2</sub>.
- The selected database must include records of vital signs at least for every hour (~ 1 data point per hour).
- Records of vital signs must include a time-stamp.
- Records for the elderly population.

- Adequate amount of records (data volume) that can be used for training and testing the proposed framework (measurements of the vital signs for at least 24 hours).

Several medical databases in ML literatures in particular for AAL health monitoring are investigated, and further, four of them that fulfilled the appointed requirements are selected. Next, the selected databases are compared to select the most suitable database for querying patient's data. As presented in Table 5-1, MIMIC-III, UCI-ICU, UCL-THIN and CPRD-HES are compared based on the level of access, size of the database, selection of parameters and interval for measurement of vital signs (temporal).

Table 5-1 Clinical Database Comparison

Data source	Access	Records	Vital Signs			Acute monitoring <sup>6</sup>	References
			BP	HR	SpO2		
PhysioNet–MIMIC III	Open Source	53,423	YES	YES	YES	YES	(Moody and Lehman, 2009), (Iram <i>et al.</i> , 2012), (Banaee, Ahmed and Loutfi, 2015), (A. E. W. Johnson <i>et al.</i> , 2016), (Forkan and Khalil, 2016), (Lehman <i>et al.</i> , 2013), (Chen <i>et al.</i> , 2018)
UCI – ICU	Open Source	303	YES	YES	YES	YES	(Bouckaert and Frank, 2004), (Wang, Yang and Dong, 2017), (Altun, A and Tunc, 2010), (Kaur and Wasan, 2006), (Polat and Güneş, 2006), (Goldstein and Uchida, 2016), (Swersky <i>et al.</i> , 2016)
UCL - THIN	Fee	11.5 m	YES	YES	YES	NO	(Blackburn <i>et al.</i> , 2018), (Pham <i>et al.</i> , 2018), (Epidemiology, 2016), (Welch, Bartlett and Petersen, 2014), (Pedersen, Henriksen and Hammer, no date)
CPRD – HES	Fee	599,586	YES	YES	YES	YES	(Analysis <i>et al.</i> , 2017), (Padmanabhan <i>et al.</i> , 2018), (Zghebi, 2018), (Tran <i>et al.</i> , 2018), (M. E. Johnson <i>et al.</i> , 2016), (Varas <i>et al.</i> , 2018)

More details on architecture and patient's data in MIMIC-III database will be discussed in Section 5.2.

<sup>6</sup> Temporal, the requirement for datasets including a record of vital signs for at least every hour (~ 1 data point per hour)

One of the medical databases that is reviewed is the ICU database which contains a large set of data including required vital signs parameters covering a period of approximately 12 hours during the patient's stay at an ICU unit. The ICU is accessible through the UCI machine learning Repository and the Waikato Environment for Knowledge Analysis (WEKA)<sup>7</sup>. The UCI machine learning Repository is a collection of 462 databases that is currently used by ML communities and the ICU database is one of the available databases that can be accessed. The WEKA is a software workbench delivered by the university of Waikato<sup>8</sup>. The WEKA is widely used in academic and commercial projects as it provides a number of ML algorithms and techniques in addition to several databases that are currently used for classification, anomaly detection and prediction tasks in smart home systems (Srinivas, Rani and Govrdhan, 2010).

The ICU database is not selected because, the volume of patient records is too small as it contains records of only 303 patients. The volume of patient records can eventually become smaller after the data processing due to missing values and irrelevant information. Moreover, the UCI covers measurements of vital signs for a short period (approximately for 12 hours), which will be insufficient to train the proposed framework. One of the core factors to train the proposed framework is that, the database should cover measurements of the vital signs for at least 24 hours.

The Health Improvement Network (THIN) is another clinical database which is investigated. THIN database is a primary care database provided by UCL<sup>9</sup> research department Primary Care and Population Health (PCPH) and Infection and Public Health (IPH). It contains records of routinely collected clinical

---

<sup>7</sup> <https://www.cs.waikato.ac.nz/ml/weka/datasets.html>

<sup>8</sup> <https://www.cs.waikato.ac.nz/>

<sup>9</sup> <https://www.ucl.ac.uk/>

data from patients registered at 587 GP surgeries across the United Kingdom (UK).

On the other hand, the Clinical Practice Research Datalink (CPRD) is linked to NHS Digital<sup>10</sup> that gives an access to databases such as hospital data, national cancer registration data and national mortality records. One of these data sets is the Hospital Episode Statistics <sup>11</sup> (HES) database containing details of all admissions, A&E attendances and patient admitted to NHS hospitals in England.

The THIN and HES databases are not selected because of two factors. Firstly, both of the databases are not open source and they require an access fee. Secondly, in the THIN database, the vital signs are not recorded periodically as it is established in this work. The second reason is that the MIMIC-III database compared to THIN and HES includes relatively the same volume of records while MIMIC-III is an open source and covering measurements of the vital signs for at least 24 hours.

## 5.2 MIMIC III

In order to evaluate the performance of the proposed framework, the experiments are conducted using the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III) of MIT PhysioBank<sup>12</sup> archive. The MIMIC-III is an update of the MIMIC-II database which is widely used in fields such as academic and industrial research (A. E. W. Johnson *et al.*, 2016). The MIMIC-III database is a relational database consisting of 26 tables, which are provided as a set of comma separated

---

<sup>10</sup> The national information and technology partner to the health and social care system

<sup>11</sup> <https://digital.nhs.uk/data-and-information/data-tools-and-services/data-services/hospital-episode-statistics>

<sup>12</sup> PhysioBank is a large and growing archive of well-characterized digital recordings of physiologic signals and related data for use by the biomedical research community. <https://physionet.org/physiobank/>

value (CSV) files, along with scripts for importing the data into several types of Database Management System (DBMS) including: PostgreSQL an object-relational DBMS, MySQL a relational DBMS, and MonteDB a column-oriented DBMS (A. E. W. Johnson *et al.*, 2016). The MIMIC-III database is accessible to researchers under a data use agreement where an access must be granted via a process documented on the MIMIC website<sup>13</sup>.

As illustrated in Figure 5-1, the MIMIC-III database includes anonymised clinical records (laboratory values, etc.) and vital signs data (HR, SBP, DBP, SpO<sub>2</sub>, etc.) of patients who are admitted to the Beth Israel Deaconess Medical Centre in Boston, Massachusetts. The patient's data are archived before being shifted to the MIMIC-III database, also data format conversion has been computed, and patient's data was protected by computing de-identification functions. The fact that it is an open source database, the quality of data in MIMIC-III is repetitively improved by users' feedback and corrections in ways that would not otherwise be possible.

---

<sup>13</sup> <https://mimic.physionet.org/gettingstarted/access/>

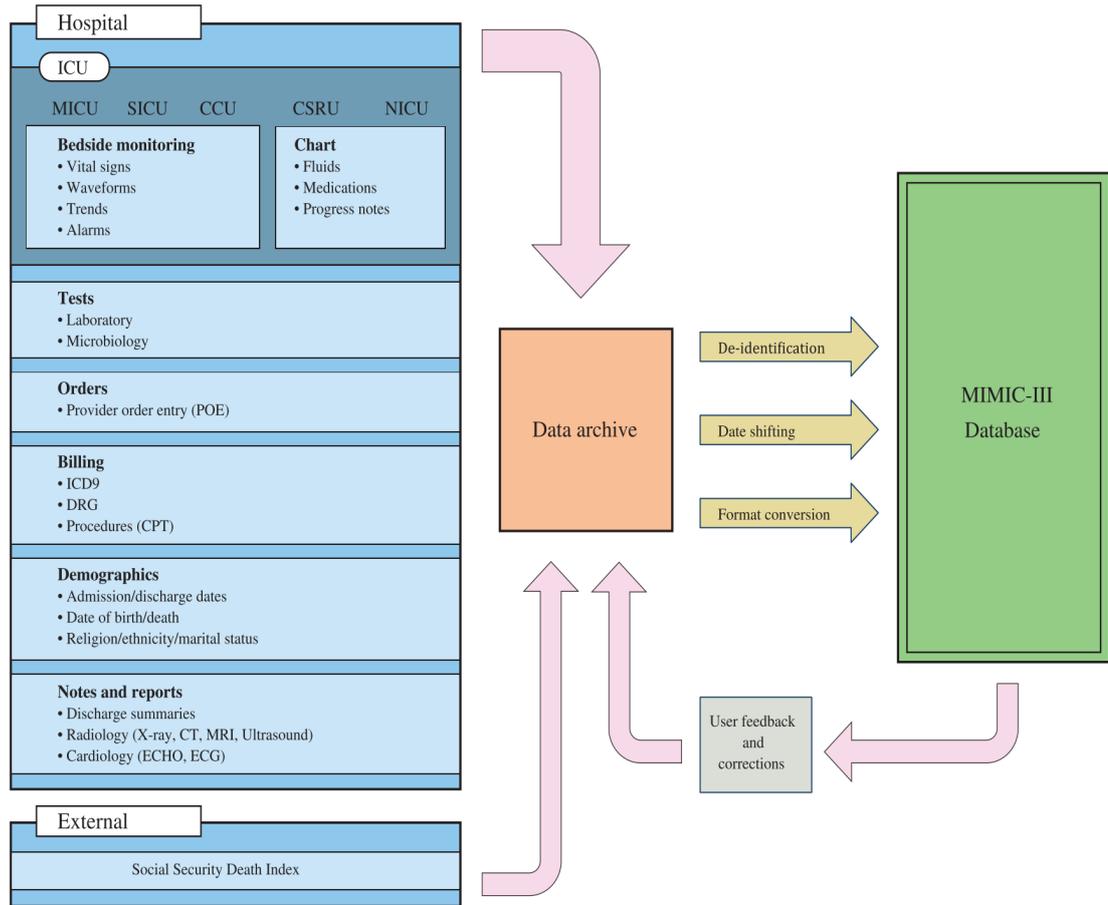


Figure 5-1 Overview of the MIMIC-III database (A. E. W. Johnson *et al.*, 2016)

The MIMIC-III database is selected for training and testing of the proposed framework because it fulfilled the criteria discussed in Section 5.1. Moreover, to the best of the author’s knowledge, there is no public database available that contains vital signs data from a larger number of patients. Table 5-2 illustrates some of the core characteristics of the MIMIC-III database that are of interest to this project.

Table 5-2 Characteristics of the MIMIC-III database

Median age (years)	65.8
Median length of an ICU stays (days)	2.1
Male sex (n, %)	55.9
Mortality	11.5

As highlighted in Section 4.2, the proposed framework is aimed to assist elderly patients who are living independently, therefore a database with sufficient

records of elderly patients is essential. According to NHS England<sup>14</sup>, generally someone over the age of 65 is considered as an elderly person, and median age of patients in MIMIC-III is 65 years. In addition to this requirement, there is a need for the database to contain at least 1-day numerical trend data of four vital signs that are discussed before. The MIMIC-III fulfilled this requirement as the median length of an ICU stay, where vital signs are recorded, is 2.1 days. The MIMIC-III also includes a good ratio of records for male and female genders. As previously discussed, a balanced ratio of records for male and female genders is an important factor in this work since the threshold of vital signs can be different based on the patient's gender. On the other hand, the mortality rate is another important factor that is considered in selecting the MIMIC-III as it has records of vital signs for ICU mortality too. This is a significant factor as patterns of the vital signs are varied (discussed in Section 4.2.1) for a patient who survives at the end of his/her ICU stay with a patient who does not. This information is considered in training and testing of the proposed framework for the learning and inference of the normal and abnormal patterns of the vital signs to predict a cardiac arrest (prognosis).

### 5.2.1 Data Acquisition and Processing

The MIMIC-III is a relational database and it consists of 26 tables. The relational database uses feature values in order to associate data from different tables. Tables in the MIMIC-III are linked to one another by unique identifiers which normally have the 'ID'. This characteristic of the MIMIC-III database is used to capture those features that are of interest for training and testing of the proposed framework. This is due to the fact that required features are stored in different tables. The CSV format of the MIMIC-III is used for this work. The tables are mainly grouped into three categories based on the type of information they contain. Five tables are to track patients' stays including: 'ADMISSION', 'PATIENTS', 'ICUSTAYS', 'SERVICES' and 'TRANSFERS'. Another five tables are known as dictionary tables which are used for cross-referencing codes against

---

<sup>14</sup> NHS England <https://www.england.nhs.uk/>

their respective definitions including: ‘D\_CPT’, ‘D\_ICD\_DIAGNOSES’, ‘D\_ICD\_PROCEDURES’, ‘D\_ITEMS’, and ‘D\_LABITEMS’. Third group of tables are associated with patients’ care including caregiver observations, billing information etc.

In order to evaluate the performance of the proposed framework, required features are queried from 7 tables by using the unique identifiers (primary key and secondary key) illustrated in Figure 5-2.

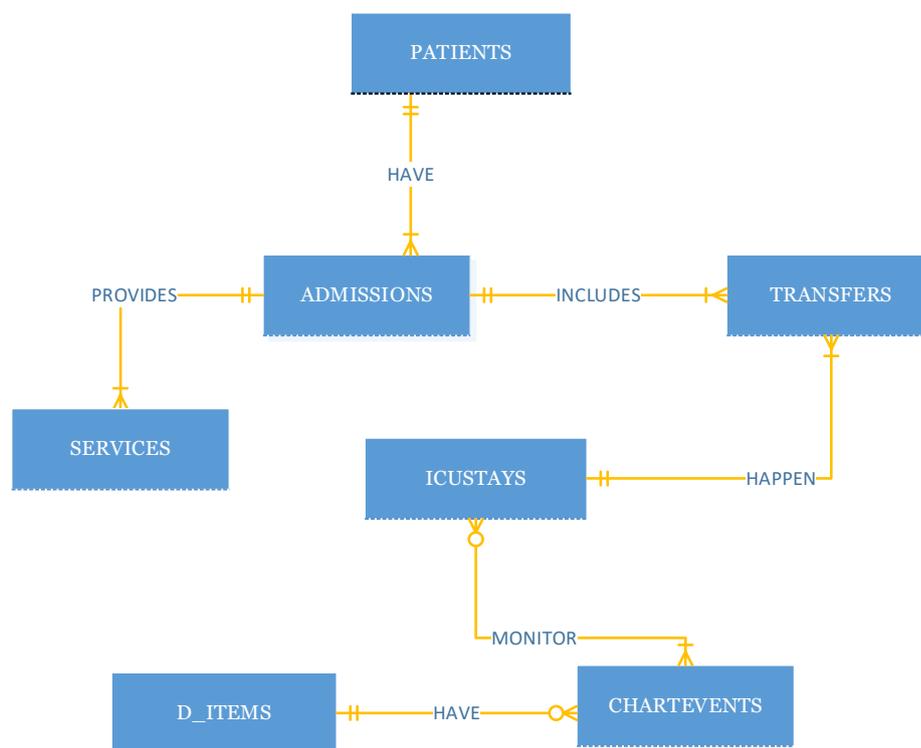


Figure 5-2 Entity Relationship Diagram of selected Tables

### 5.2.1.1 PATIENTS

The PATIENTS table covers records of patients who are admitted to the hospital. The PATIENTS table is linked to DMISSIONS and ICUSTAYS tables by ‘subject\_id’ field.

This table is used to collect the value of gender field and to calculate the age of the patient. The gender feature is used in this work to classify the patients into male and female groups. On the other hand, one of the core requirements for the

training and testing of the proposed framework is to monitor the elderly patients. Therefore, the patient's data is filtered by selecting data of patients who are older than 65 years old. To calculate the age, the value of 'dob' field from the PATIENTS table and value of 'intime' field from ICUSTAYS table are selected, and next, the value of 'intime' is subtracted from value of 'dob'. The 'intime' field store the date and time when a patient is transferred into the ICU.

Table 5-3 Detailed description of PATIENTS table

<b>Feature</b>	<b>Description</b>
SUBJECT_ID	A unique identifier which specifies an individual patient
GENDER	A genotypical sex of the patients
DOB	The date of birth of the given patient.
DOD	The date of death for the given patient.
DOD_HOSP	The date of death as recorded in the hospital database
DOD_SSN	The date of death from the social security database
EXPIRE-FLAG	A binary flag which indicates whether the patient died

One of the challenges in this stage was to transform the date of birth of elderly patients, because all dates stored in the MIMIC-III database are shifted to protect patient confidentiality. According to (A. E. W. Johnson *et al.*, 2016) dates are internally consistent for the same patient, but randomly distributed in the future. Dates of birth which occur in the present time are not true dates of birth. Furthermore, dates of birth which occur before the year 1900 occur if the patient is older than 89. In these cases, the patient's age at their first admission has been fixed to 300.

#### 5.2.1.2 ADMISSIONS

The ADMISSIONS table contains information regarding a patient's admission to the hospital. Each row of this table includes a unique identifier 'HADM\_ID' that specifies a unique visit from a patient.

The ADMISSIONS table is linked to the PATIENTS table using 'subject\_id'. This link is used for representing details of multiple admissions for a single patient. The 'diagnosis' field is used in this table which stores the medical conditions of a

patient such as hypertension, hypotension, bradycardia and tachycardia. The value of this field is selected to group the patients into a different class of medical condition. The Table 5-4 shows other features of the ADMISSIONS table for storing different information on admission.

Table 5-4 Detailed description of ADMISSIONS table

<b>Feature</b>	<b>Description</b>
SUBJECT_ID	A unique identifier which specifies an individual patient
HADM_ID	A unique identifier which specifies an individual patient's admission to the hospital
ADMITTIME	The date and time the patient was admitted to the hospital
DISCHTIME	The date and time the patient was discharged from the hospital
DEATHTIME	The time of in-hospital death for the patient
ADMISSION_TYPE	The type of the admission: 'ELECTIVE', 'URGENT', 'NEWBORN' or 'EMERGENCY'
ADMISSION_LOCATION	Information about the previous location of the patient prior to arriving at the hospital
DICHARGE_LOCATION	Information about the location of the patient where discharge from the hospital
INSURANCE	
LANGUAGE	
RELIGION	These attributes describe patient demographics
MARITAL_STATUS	
ETHNICITY	
EDREGTIME	
EDOUTTIME	The time that the patient is discharged from the emergency department
DIAGNOSIS	A preliminary, free text diagnosis for the patient on hospital admission
HOSPITAL_EXPIRE_FLAG	This indicates whether the patient died within the given hospitalization. 1 indicates death in the hospital, and 0 indicates survival to hospital discharge
HAS_CHARTEVENT_DATA	Information about charted data available for a patient

## 5.2.1.3 ICUSTAYS

This table is derived from the TRANSFERS table. Specifically, it groups the TRANSFERS table based on 'icustay\_id'. The ICUSTAYS table consists of information on first and last ICU type that a patient may have receive during the ICU stays. Furthermore, the link between ICUSTAYS and TRANSFERES is used based on 'icustay\_id' and 'subject\_id' in order to select length of ICU stays (LOS) for each patient. The Table 5-5, shows other features of the ICUSTAYS table for storing different information during entire ICU stays.

Table 5-5 Detailed description of ICUSTAYS table

<b>Feature</b>	<b>Description</b>
ICUSTAY_ID	A unique identifier which specifies an individual patient ICU stay
SUBJECT_ID	A unique identifier which specifies an individual patient
HADM_ID	A unique identifier which specifies an individual patient's admission to the hospital
DBSOURCE	Information on source of ICU database. Two sources are included: 'carevue' and 'metavision'
FIRST_CAREUNIT	Information on first ICU type in which the patient is cared for
LAST_CAREUNIT	Information on last ICU type in which the patient was cared for
FIRST_WARDID	Information on first ICU unit in which the patient stayed
LAST_WARDID	Information on last ICU unit in which the patient stayed
INTIME	The time and date the patient was transferred into the ICU
OUTTIME	The time and date the patient was transferred out of the ICU
LOS	The length of stay for the patient for the given ICU stay, which may include one or more ICU units and it is calculated in fractional days.

## 5.2.1.4 TRANSFERS

The TRANSFERS table is mainly used to filter the patients where their lengths of ICU stay are at least 24 hours. Value of LOS field is used in this table based on 'subject\_id' and 'icustay\_id' from the ICUSTAYS table. The Table 5-6, shows other features of the TRANSFERS table to store information during a single ICU stay including information on time and date when a patient was transferred into and when he/she has transferred out of the current care unit. This duration is

calculated by subtracting value of 'outtime' field from value of 'intime' field. The calculated value is used and compared with 'charttime' field in CHARTEVENTS table to ensure measurement of vital signs for a single ICU stay are selected.

Table 5-6 Detailed description of TRANSFERS table

Name	Description
SUBJECT_ID	A unique identifier which specifies an individual patient
HADM_ID	A unique identifier which specifies an individual patient's admission to the hospital
ICUSTAY_ID	A unique identifier which specifies an individual patient ICU stay
DBSOURCE	Information on source of ICU database. Two sources are included: 'carevue' and 'metavision'
EVENTTYPE	Information on what transfer event occurred for an admission including 'admit', 'transfer' and 'discharge'
PREV_CAREUNIT	Information on care unit in which the previously resided
CURR_CAREUNIT	Information on care unit in which the currently resides
PREV_WARDID	Information on ward where the patient was previously stayed
CURR_WARDID	Information on ward where the patient was currently stayed
INTIME	The time and date the patient was transferred into the current care unit from previous care unit
OUTTIME	The time and date the patient was transferred out of the current care unit
LOS	Information on length of stay for the patient for the given ward stay

#### 5.2.1.5 SERVICES

The SERVICES is used for filtering the patients who have received Medical-general service for internal medicine (MED). According to Health Education England<sup>15</sup> (HEE), patients who receive the MED service, normally present a wide range of acute and long-term medical conditions and symptoms (hypertension, hypotension for example). The SERVICES table also consists of other types of service that a patient is admitted under. The list of service types including a description of each one of them is documented in Appendices. The reason for

<sup>15</sup> <https://www.hee.nhs.uk/>

selecting patients who only received MED service is to improve contextual information by having patients who have received similar treatments. The Table 5-7, presents other features of the SERVICES table.

Table 5-7 Detailed description of SERVICES table

<b>Name</b>	<b>Description</b>
SUBJECT_ID	A unique identifier which specifies an individual patient
HADM_ID	A unique identifier which specifies an individual patient's admission to the hospital
TRANSFERTIME	The time at which the patient moved from the PREV_SERVICE to CURR_SERVICE
PREV_SERVICE	Information on previous service that the patient resides under
CURR_SERVICE	Information on current service that the patient resides under

#### 5.2.1.6 CHARTEVENTS

The CHARTEVENTS consists of the measurement of vital signs during the patient ICU stays. The patient's routine vital signs and any additional information relevant to their care are measured and stored regularly. The CHARTEVENTS is linked to PATIENTS and ICUSTAYS tables. The relationship between these 3 tables are used to firstly, select the timestamp at which a measurement was made, and secondly to select the value of four required vital signs including SBP, DBP, HR and SpO2. Value of the 'charttime' field is selected as it represents the timestamp at which a measurement for vital sign is made. Moreover, the 'itemid' is selected to link to the D\_ITEM table in order to identify the required vital signs. For instance, 'itemid' for SBP, DBP, HR and SpO2 are 220180, 220181, 220045, 220277 respectively (metavision source) which can be found in D\_ITEM table. The Table 5-8, shows other features of the CHARTEVENTS table.

Table 5-8 Detailed description of CHARTEVENTS table

<b>Name</b>	<b>Description</b>
SUBJECT_ID	A unique identifier which specifies an individual patient
HADM_ID	A unique identifier which specifies an individual patient's admission to the hospital
ICUSTAY_ID	A unique identifier which specifies an individual patient ICU stay
ITEM_ID	A unique identifier for a single measurement type in the database. Each row associated with one ITEMID (e.g. 220277) corresponds to an instantiation of the same measurement (e.g. SpO2).
CHARTTIME	The time at which a measurement was made
STORETIME	The time at which a measurement was manually inputted by a member of the clinical staff
CGID	The identifier for the caregiver who validated the given measurement
VALUE	Value measured for the parameter identified by the ITEMID
VALUENUM	Information on the unit of measurement for the VALUE
WARNING	'Metavision' specific column which specify if a warning for the value was raised
ERROR	'Metavision' specific column which specify if an error occurred during the measurement
REULTSTATUS	'carevau' specific column which specify the type of measurement
STOPPED	'carevau' specific column whether the measurement was stopped

#### 5.2.1.7 D\_ITEMS

The D\_ITEMS table belongs to the dictionary table group which are used for cross-referencing codes against their respective definitions. It contains label and abbreviation columns in order to describe the concept which is represented by the 'itemid'. The 'itemid' from CHARTEVENTS table is selected and linked to D\_ITEMS table to find the label of a correct vital sign. The Table 5-9, shows other features of the D\_ITEM table.

Table 5-9 Detailed description of D\_ITEMS table

Name	Description
ITEMID	A unique identifier which represents measurements of a parameter
LABEL	Describes the concept which is represented by the ITEMID
ABBREVIATION	'Metavision' specific column which lists a common abbreviation for the label
DBSOURCE	Information on source of ICU database. Two sources are included: 'carevue' and 'metavision'
LINKSTO	Provides the table name which the data links to
CATEGORY	Information of the type of data the ITEMID corresponds to
UNITNAME	Specifies the unit of measurement used for the ITEMID
PARAM_TYPE	Describes the type of data which is recorded
CONCEPTID	A unique identifier which represents concept represented by the ITEMID

### 5.3 Extracted Dataset

A dataset is extracted based on several features from MIMIC-III database. The PATIENTS table is filtered to use the dataset collected from a group of elderly patients. In this work, it has been decided to remove the data for the patients who have not received the MED service as their length of an ICU stays were too low to be used for anomaly detection.

Moreover, data of patients who have had a hypotension condition are selected to specify a normal baseline threshold for training and testing the proposed framework. Furthermore, it is noticed that some of the vital signs have a missing value. The *mean* function is used to calculate mean values of available measurements of the vital signs, and missing values are substituted with the computed mean value. The characteristics of the generated dataset is shown in Table 5-10.

Table 5-10 Characteristics of extracted dataset

<b>Appearances</b>	<b>Total record</b>
No. of 'MED' service	247
Male sex (n, %)	50
Survival ratio (n, %)	19
Median length of an ICU stays (days)	1.2
Median age (years)	67

The generated dataset which will be used for training and testing of the proposed framework consists of five features including: timestamp, HR, SBP, DBP and SpO<sub>2</sub> (Table 5-11).

Table 5-11 Example of features in generated dataset

timestamp	hr	bps	bpd	spo2
04/08/2154 19:00	97	82	49	94
04/08/2154 19:46	97	79	49	94
04/08/2154 19:49	80	79	49	94
04/08/2154 20:00	76	80	39	94
04/08/2154 20:02	76	80	39	92
04/08/2154 20:17	78	80	39	93
04/08/2154 20:52	83	80	39	93
04/08/2154 21:00	85	78	47	99
04/08/2154 22:00	81	80	46	99
04/08/2154 23:00	86	88	50	96
05/08/2154 00:00	96	98	61	95
05/08/2154 01:00	93	100	55	95
05/08/2154 02:00	97	113	64	97

## 5.4 Summary

In this Chapter, a secondary contribution of this thesis is achieved by generating a dataset that consists of the required features. The MIMIC-III medical database among several medical databases are investigated, and their characteristics are compared against a number of factors. As a result of the comparison, it is decided to use the MIMIC-III database. As highlighted previously, 7 tables out of 26 tables are selected for further data processing. Features of each table are discussed and the relationships between tables are also identified and used for selecting the required features. The relationship between tables is used to filter and select 5 features including: timestamp, HR, SBP, DBP and SpO<sub>2</sub>. Total records of 247

older patients are selected who suffer from hypotension. It is ensured that records of distinctive patients consist of measurements of vital signs for at least 24 hours. Furthermore, it is ensured to have a balance record of patients for different genders (50% male).

The generated dataset will be used in the next Chapter for training and testing the proposed framework. Furthermore, the generated dataset will be applied for evaluating the performance of several state-of-the-art algorithms and for comparing their performance with the performance of the proposed framework.

## Chapter 6

# TEST AND EVALUATION

In this Chapter, the performance of the proposed model is evaluated to gain additional insight into its real performance in detecting abnormal patterns in multiple vital signs. Furthermore, the performance of the proposed model is compared with several ML techniques and algorithms to detail the best-performed algorithm toward the predefined model scenario (Section 4.2).

This Chapter starts with a detailed explanation of how the experiments are designed by discussing the contextual information, different experiment parameters and evaluation metrics. This is used to form the basis for the next phase of selecting the parameter setting for the proposed model with the best performance. The parameter settings with the best performance are used in order to compare the performance of the proposed model with several anomaly detection algorithms and techniques. The performance evaluation is conducted using both an on-line learning and a traditional learning fashion. The Numanta Anomaly Benchmark (NAB) is used for on-line learning and other performance metrics including recall, precision and F-measure are used as a traditional evaluation method for the algorithms that use batch processing.

## 6.1 Experimental Setup

Experimental studies are conducted on several anomaly detection techniques and algorithms to test and compare their performance. The overall design of the experiment is shown in Figure 6-1.

After the preparation of the 247 records extracted from the MIMIC-III database (see Section 6.1.2), the records of each dataset are inputted into a number of ML models. The ML models produce an anomaly score for each record it reads. The output of the models is a patient dataset with the row's number and the associated anomaly score. To evaluate the performance of the models, two evaluation metrics are used including NAB score and F-measure (see Section 6.1.4).

As part of the testing methodology, an empirical experimentation is conducted to optimise the parameter settings of the SP and TM algorithms of the proposed model. Hence several performance passes are conducted to test the behaviour of the proposed model using the NAB score. From the empirical experimentation, the best model parameter settings producing the best NAB score are selected and used to additionally compute the F-measure of the proposed model. Furthermore, NAB score and F-measure of an HTM model with a default parameter setting recommended by the NuPIC are calculated. This is followed by testing the performance of several other algorithms including: k-Nearest Neighbour (k-NN), k-NN-PCA, INFLO, INFLO-SVD, SVM one class and HBOS.

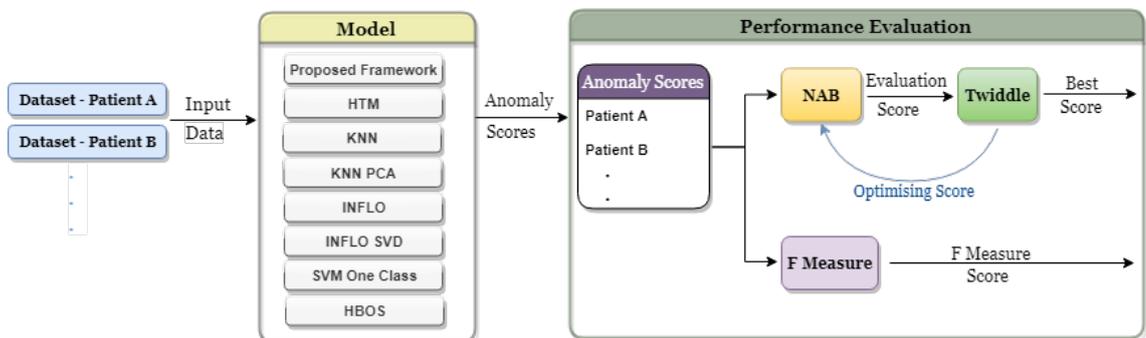


Figure 6-1 The Experimental setup

Two data science software platforms are used for implementation and evaluation of the models. The NuPIC version 1.0.5 platform with NuPIC Bindings version 1.0.0 and Python version 2.7.3 are used to build the HTM models. Furthermore, the RapidMiner Studio<sup>16</sup> platform version 9.0 is used to implement the k-NN, k-NN with PCA, INFLO, INFLO with SVD, HBOS, and SVM one class.

The work in this Chapter presents a secondary contribution to knowledge, as the performance of the proposed model is tested and compared with several state-of-the-art algorithms using two different types of evaluation metrics including the NAB score and F-measure.

### 6.1.1 Contexts

As discussed in Section 2.2.3, the context-awareness computing is about using the AAL systems in particular health monitoring to automatically perceive contextual information from a person and take an action according to the person's current context and needs (Mshali *et al.*, 2018). More importantly, according to Jih, Hsu and Tsai (2006) context models play an essential role in temporal reasoning tasks.

Hence, a context model is defined in this work to represent the age, gender, medical condition, and time contexts. The age context is defined for patients over the age of 65 (older person). The gender contexts are 'male' and 'female' contexts. The medical condition context is defined for the elderly patients who have hypotension symptoms, and the time contexts are defined as 'morning', 'mid-day', 'evening', and 'night' contexts.

A rationale for having the context model described above is that by specifying the age group, gender and medical condition, a threshold for normal and abnormal values of vital signs could be defined that will be fed to the ML algorithms.

---

<sup>16</sup> <https://rapidminer.com/>

As discussed in Section 4.2.1, time contexts in the TR paradigm are used to abstract meaningful patterns. These patterns are widely used in AAL environments for diagnosis and prognosis tasks. Hence, in this work, the period-based approach presented in Figure 6-2 is used to abstract temporal patterns of the vital signs in order to detect contextual anomalies at least 10 hours prior to a cardiac arrest. The period-based approach is discussed in Section 2.5.1.

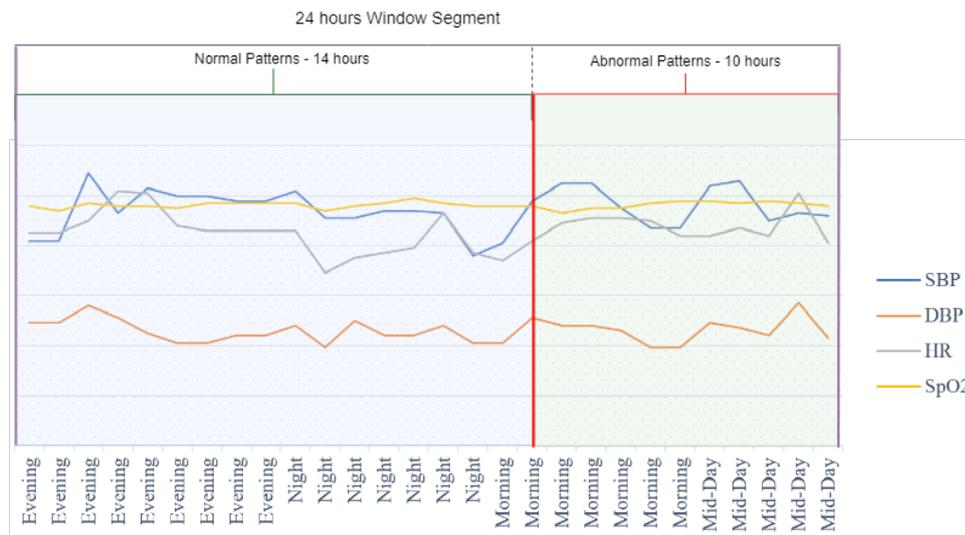


Figure 6-2 An applied period-based approach for temporal abstraction

According to several clinical studies (Clement DL, De Buyzere M, De Bacquer DA, 2003; Oh, Lee and Seo, 2016; Forkan and Khalil, 2017a), the detectable changes in the vital signs appear 18-20 hours before the cardiac arrest and become dramatic at 5-10 hours.

### 6.1.2 The Patients and Datasets

In this experimentation, the extracted datasets from the MIMIC-III database are employed. In detail, the record of each patient is stored in a single dataset. Hence, 247 individual datasets including 219 survival cases and 28 mortality cases from patients who are admitted to the ICU are extracted from the MIMIC-III database (see Table 6-1).

Table 6-1 The 247 datasets and the corresponding number of records

	Learning Set	Inference Set	
Survival	180	39	<u>18 Females</u> 21 Males
Mortality	0	28	<u>17 Females</u> 11 Males
Male	91	32	
Female	89	35	
Total	180	67	

The datasets are split into learning and inference sets using the “HOSPITAL\_EXPIRE\_FLAG” filed in the admission table. This field indicates whether the patient died within the given hospitalization. The learning sets consist of the records of patients who have survived during their ICU stays, whereas the inference sets consist of a combination of records from patients who have survived during their ICU stays (39 datasets) and patients who have died during their ICU stays (28 datasets).

### 6.1.3 Experiment parameters

As discussed previously in Section 3.2.3, two core functions in the HTM system are identified as learning and inference. In the proposed framework, each region is configured to firstly learn the complex patterns from the learning set. The regions begin the inference process once they complete the learning process. The rationale for the learning and inference phases is for the regions to first learn the normal patterns of vital signs for each patient who has survived the ICU stay (180 patients). Afterwards, the learning function is switched off and the inference is activated. The pseudocode and algorithms code of the proposed framework are documented in Appendix B.

In the inference phase, regions within two levels recognising spatial and temporal patterns of the vital signs as similar to previously learned patterns without updating the synapses, and if abnormal patterns (unseen patterns) are detected then the proposed framework will compute scores that indicates likelihood of an

anomaly. Hence, for the inferencing, the inference set (67 patients) consists of survival and mortality records are used to evaluate the performance of the proposed framework. The proposed framework must detect abnormal patterns of vital signs for a patient who does not have survived during his/her ICU stay.

#### 6.1.4 Evaluation metrics

In this work, the experiments focus on two types of evaluation metrics including F-measure and NAB score. These two metrics are used to test the performance of the ML models under the experimentation including the proposed framework.

To evaluate the performance of a ML model, the anomaly scalar score (range from 0.0 to 1.0) of the ML algorithm under consideration is normally labelled as positive ( $p$ ) or negative ( $n$ ) which then can be compared to the input value. As discussed in Section 2.4.6, different evaluation techniques are used to evaluate and analyse the performance of the ML algorithms, to name a few: precision, recall, F-measure (F-score), Hamming Loss and Hamming score (Forkan and Khalil, 2017a). According to (Nehmer *et al.*, 2006) and (Haque, Rahman and Aziz, 2015), anomaly detection models that are applied in AAL health monitoring systems should have a high recall rate in detecting every real emergency immediately and have a high precision rate, in order to deny false emergency detections and alerts.

However, the techniques emphasised above are almost sufficient to evaluate the performance of ML algorithms when used in a traditional fashion i.e. batched processing, and when the time aspect is not considered in the evaluation process. In a case of AAL in particular health monitoring scenario where vital signs are measured temporally and in an online/real time mode, it is essential to detect anomalies in vital signs as early as possible in order to prevent any critical events.

For this reason, in this work as a secondary contribution, the NAB an open source tool is used to evaluate performance of the algorithms on temporal data and in an-online scenario. The NAB is used in this work as according to (Lavin and Ahmad, 2015), the NAB evaluates the performance of ML models by giving more

credit to algorithms that detect anomalies as soon as possible while the false alarm can be ignored. More details on the NAB are discussed in Section 2.4.6.

In order to compute the F-measure and the NAB score, the best F-measure score for each algorithm is firstly selected. In this phase, different parameter settings for each model are attempted. Afterwards, the anomaly scores of the parameter setting that has produced that best F-measure score are stored and used to compute the NAB score. More details on the outcome of testing are documented in Section 6.2.

#### 6.1.4.1 NAB Score

As previously discussed in Section 2.4.6, for the NAB to evaluate the performance of a ML model, it relies on three components, including: *anomaly windows*, *scoring function* and *application profiles*. The anomaly windows are represented around every anomaly point which are the actual values. The location of the anomaly window must be selected based on the contextual information. In this work, the size of the anomaly window is selected and labelled based on the interval of abnormal patterns which are shown in Figure 6-2. The labels are applied for the NAB to evaluate the performance of the models. Furthermore, the NAB uses application profiles for customising the scoring methodology, these profiles are highlighted in Section 2.4.6. In this work, the selected application profile is based on the proposed model scenario (Section 4.2) which rewards the model that does not miss any TP anomalies, it would rather trade off a few FPs than miss any true anomalies. This is because of a necessity to not miss any indications of critical events in AAL health monitoring scenarios.

#### 6.1.4.2 Threshold Optimisation

In order to evaluate the performance of a ML model, a threshold value must be defined for which a score can be interpreted as a normal or an anomaly score. In AAL health monitoring scenarios, patients have different characteristics due to various risk factors, in the health domain are referred to as the modifiable factors.

The modifiable factors are defined by British Heart Foundation<sup>17</sup> as risk factors that can be controlled by patients, to name a few, weekly alcohol consumption, Body Mass Index (BMI), smoking, regular exercise etc. Hence, what is considered as an anomaly for a patient, might not be an anomaly for another patient, consequently a fixed threshold for all the dataset will not be an ideal solution. In this work, to overcome this obstacle, the Twiddle algorithm provided in the NAB is used (shown in Figure 6-1). The Twiddle algorithm is known as a local hill-climbing algorithm that tries several thresholds and selects the ones that can be used for all datasets to produce the best performance.

#### 6.1.4.3 F-measure

In this experimentation, the performance of the models is also evaluated using F-measure score. The F-measure score is the harmonic mean of the precision and recall scores. In this case, F-measure is used to compare the precision and recall of the models. Using the F-measure for testing will define how precise the model is (how many instances it classifies correctly) as well as how robust it is (it does not miss a significant number of instances). The equations for recall, precision and F-measure are shown in Equation (3).

## 6.2 Results

This section describes the experimental results obtained from the proposed framework followed by a comparison of its performance with several ML algorithms. The performance of several anomaly detection techniques and algorithms are tested using the NAB score and F-measure metrics. The ML algorithms are tested in this section are split into HTM based including HTM model and proposed model, Nearest Neighbour based, Cluster and Density based techniques. Hence several supervised and unsupervised anomaly detection algorithms are tested where appropriate steps are followed in particular to train and test the supervised algorithms. The main reason for evaluating the performance of supervised and unsupervised algorithms is to compare their

---

<sup>17</sup> <https://www.bhf.org.uk>

performance with each other and to gain a better insight of these algorithms in AAL health monitoring scenario.

### 6.2.1 HTM

Two HTM models were built, the first model is implemented by using parameter settings from NuPIC, and the second model is implemented based on the proposed framework. The algorithm API is particularly used to implement the proposed framework wiring the SP and TM algorithms for different regions according to HTM theory. This allows direct access to different components of the proposed framework to instantiate these components within the regions manually and wire regions together within two levels of the proposed framework hierarchy.

Moreover, to evaluate and compare the performance of these two models, the raw anomaly score metrics are used. The raw anomaly score is the ratio of the active columns that are wrongly predicted by the model to the total number of columns.

#### 6.2.1.1 HTM model

As highlighted previously, the HTM model is built using parameter settings recommended by NuPIC. As the project's contribution, the performance of the proposed framework will be compared with this model to check if the proposed framework improves the performance of existing HTM model in AAL health monitoring scenario. The Listing 6-1 shows the parameter settings for the SP and TM algorithms selected from NuPIC stable version<sup>18</sup>.

---

<sup>18</sup> <http://nupic.docs.numenta.org/stable/>

```

SpatialPooler (
    inputDimensions = (884,),
    columnDimensions = (2048,),
    synPermConnected = 0.1,
    numActiveColumnsPerInhArea = 40.0,
    boostStrength = 3.0,
    synPermActiveInc = 0.04,
    synPermInactiveDec = 0.005,
    globalInhibition = True,
    potentialPct = 0.85,
    spVerbosity = 0,
    minPctOverlapDutyCycle = 0.001,
    dutyCyclePeriod = 1000,
    localAreaDensity = -1.0,
    potentialRadius = 16,
    stimulusThreshold = 0),

TemporalMemory (
    columnDimensions = (2048,),
    cellsPerColumn= 32,
    initialPermanence = 0.21,
    connectedPermanence = 0.5,
    maxNewSynapseCount = 20,
    permananceIncrement = 0.1,
    permananceDecrement = 0.1,
    minThreshold = 12,
    activationThreshold = 16,
    maxSegmentsPerCell =128,
    maxSynapsesPerSegment =32,
    predictedSegmentDecrement = 0.0)

```

---

Listing 6-1 Parameter settings for typical use cases recommended by NuPIC

The NAB score of the model is computed and then its performance is evaluated using a traditional metric, f-measure.

Table 6-2 HTM model results

<b>NAB score (%)</b>	<b>F-measure (%)</b>	<b>Recall (%)</b>	<b>Precision (%)</b>
62.5	27.16	27.26%	77.76%

The HTM model is tested using the NAB and f-measure metrics. As shown in Table 6-2, the average NAB score achieved by the HTM model is 62.5% and F-measure score is 27.16%.

Next, the proposed framework is tested, and its performance will be compared with the HTM model and several algorithms.

### 6.2.1.2 Proposed Model

In this section, the result of applying the proposed model on 247 patient datasets is presented. An empirical experimentation is conducted due to the lack of a good

optimisation algorithm for selecting optimum values for the parameters for SP and TM algorithms.

In order to evaluate the performance of the proposed model, the NAB score is firstly used in empirical experimentation to select optimum parameter settings for SP and TM algorithms. Afterward, the optimum parameter settings are used to evaluate its performance using traditional evaluation metrics including F-measure, recall and precision scores.

As discussed in Section 4.3, each region within the 2 levels of the proposed model has allocated SP and TM algorithms which are used to learn spatial and temporal patterns of the input space. Hence, parameter settings for the SP and TM in level one, are firstly optimised before the optimisation for parameters of SP and TM in level 2.

For each level in the proposed model's hierarchy, optimum parameter settings of the SP algorithm are first selected followed by selection of optimum parameter settings for the TM algorithm. This is due to the fact the encoded data is fed into the SP for columns activation (spatial patterns) before it is fed into the TM algorithm for cells activation (temporal patterns).

#### *6.2.1.2.1 Level One*

The column dimensions parameter (`columnDimensions`) in the SP and TM algorithms defines the dimensions for each algorithm within regions of the proposed framework. As illustrated in Figure 6-3, several values for this parameter is tested and a best performance is achieved when column dimensions are set to 600. Comparing this value with the default value (etc. 1024) recommended by NuPIC has shown an improvement by 6%.

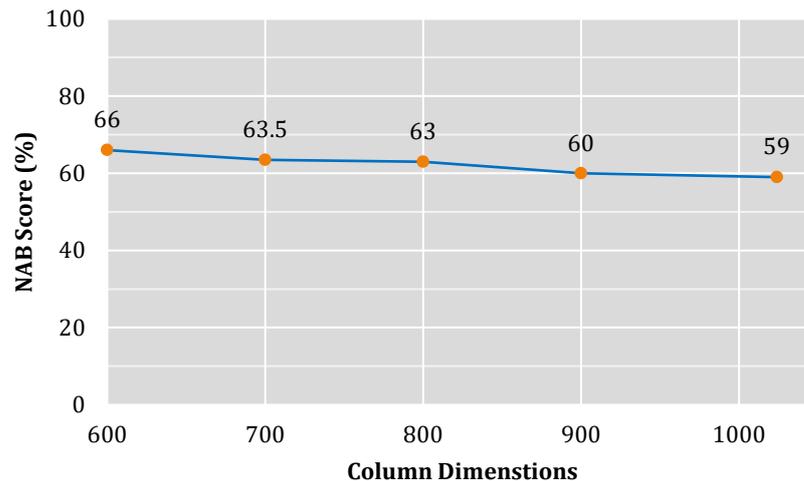


Figure 6-3 The NAB Score for the proposed model with different column dimensions

The performance of the proposed model is also tested when changing the density of active columns within a local inhibition area. This is tested by changing the 'numActiveColumnsPerInhArea' parameter and result is shown in Table 6-3. In this part, LocalAreaDensity parameter is set to -1 as a requirement.

Table 6-3 Proposed model with different 'numActiveColumnPerInhArea' values

<b>numActiveColumnPerInhArea</b>	<b>NAB Score (%)</b>
12	66
24	64
36	62.5

One significant factor in here is to have a balanced level of activation densities. As highlighted in a work by (Cui, Ahmad and Hawkins, 2017), high activation density would more likely contributes to high false positives error whereas low activation density on the other hand can contributes to false negative error as it will be much harder to detect patterns with low activation densities.

In Figure 6-4 the performance of the proposed model was evaluated with different boost strength (boostStrength), values ranging from 0 to 4.

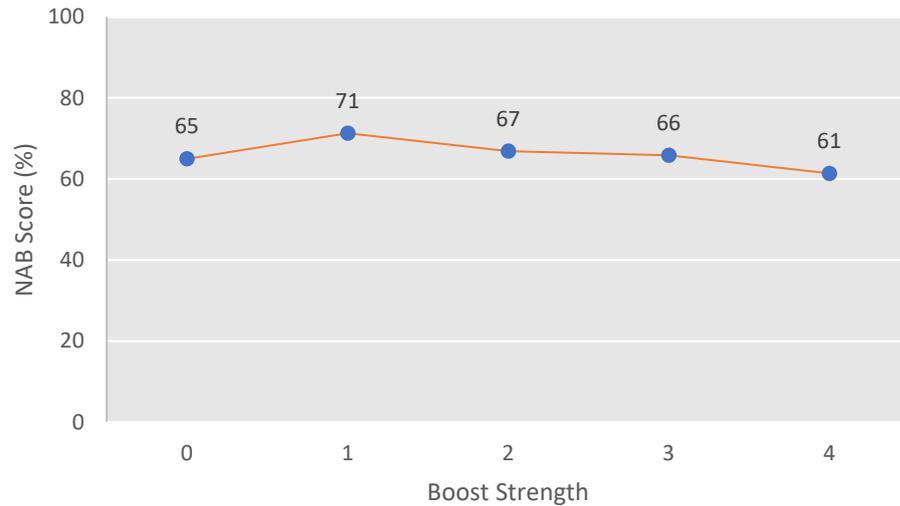


Figure 6-4 The NAB Score for the proposed model with different Boost Strength values

Figure 6-5, illustrates the behaviour of the proposed model with different values of the parameter ‘synPermActiveInc’. This parameter defines amount of permanence values of active synapses are to be incremented during the learning. Default value is 0.03, however the proposed model has shown a better performance when ‘synPermActiveInc’ is set to 0.01.

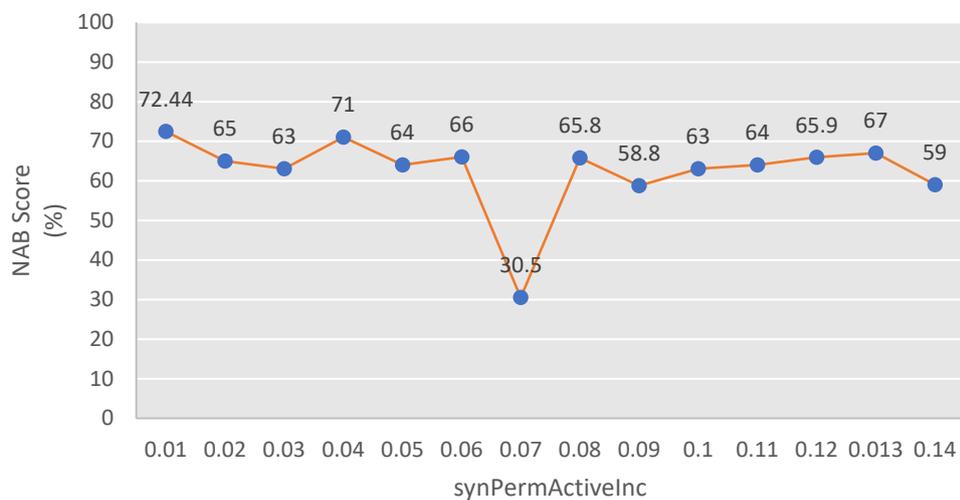


Figure 6-5 The NAB Score for the proposed model with different SynPermActiveInc values

Several performance passes were carried out to evaluate the behaviour of the proposed model with respect to the parameters ‘synPermActiveInc’,

‘potentialPct’, ‘potentialRadius’, ‘cellsPerColumn’, and ‘activationThreshold’, however no noticeable changes in model’s behaviour were detected.

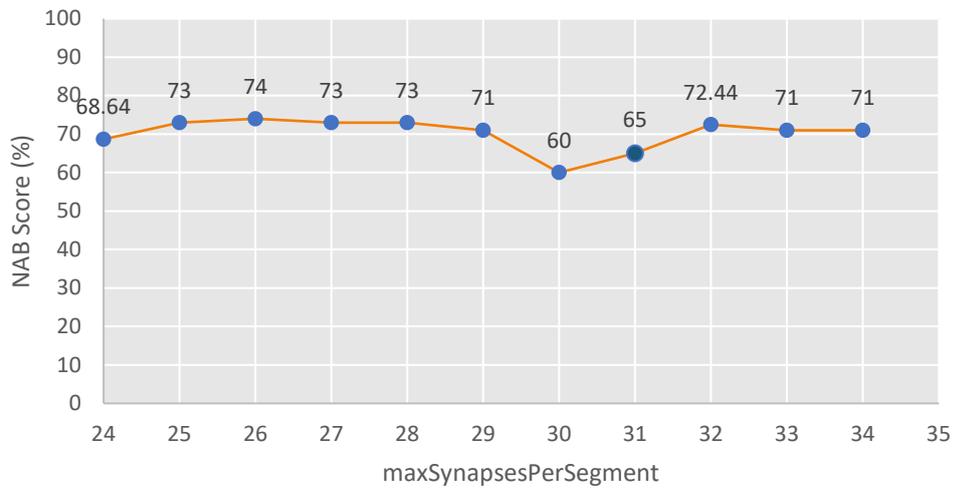


Figure 6-6 The NAB Score for the proposed model with different maxSynapsesPerSegment values

Further tests are conducted and different values of the parameter ‘maxSynapsesPerSegment’ have been tried (Figure 6-6). The level 1 of the proposed framework has achieved 74% NAB accuracy mark.

#### 6.2.1.2.2 Level Two

Once optimum parameter settings for SP and TM algorithms in level 1 are identified, further empirical experimentations are conducted on level 2 to optimise parameter settings for the SP and TM algorithms.

In this section, the same experimental steps as level 1 are followed. The noticeable changes in behaviour of the proposed model are detected particularly when the ‘synPermInactiveDec’ of SP algorithm, and ‘permanenceDecrement’ of TM algorithm are tested.

In Figure 6-7 and Figure 6-8, behaviour of the proposed model is illustrated when several values are tested for ‘synPermInactiveDec’ and ‘PermanenceDecrement’ parameters. These two parameters are aligned with inactive bits which values of their synapses are decremented in column and cell levels respectively. The idea of making synapses more or less permanent is coming from “Hebbian” learning

rules, where the permanence value of the synapses will be incremented if the synapses are active that contribute to the cell's activation, otherwise their permanence value will be decremented which means they did not contribute to the cell's activation. This process is also referred to the reinforcement of the dendritic segment in (Cui, Ahmad and Hawkins, 2016).

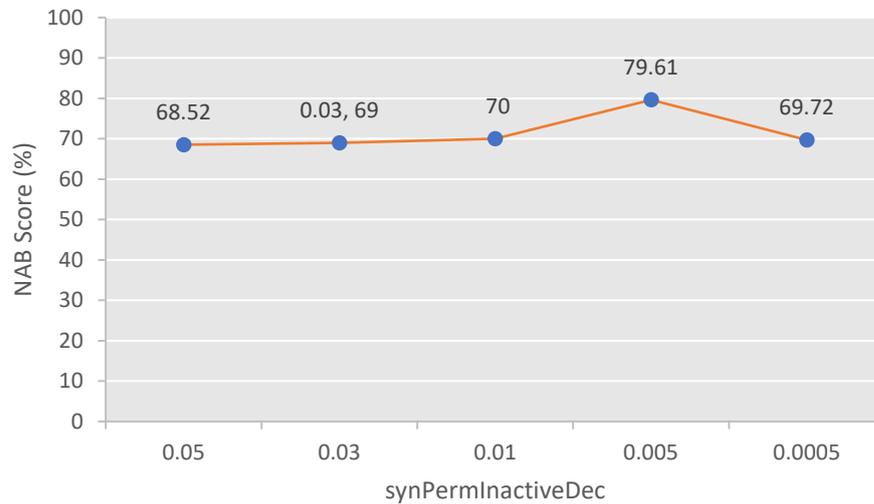


Figure 6-7 The NAB Score for the proposed model with different synPermInactiveDec values

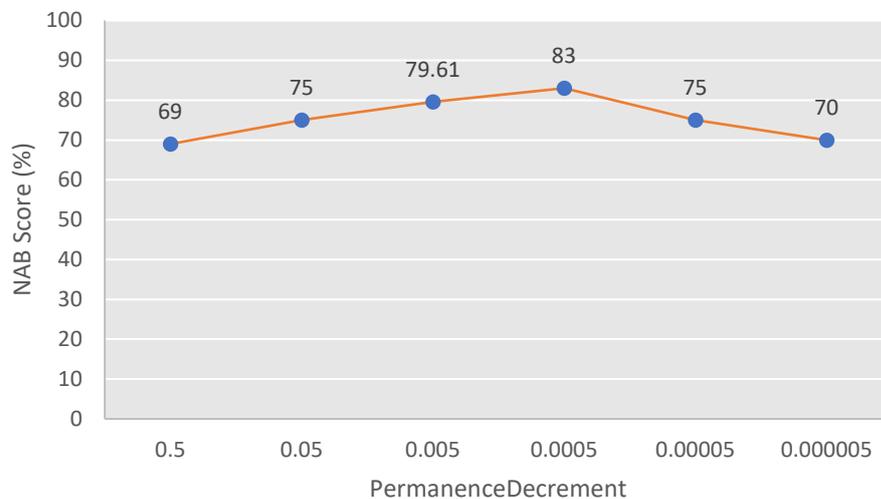


Figure 6-8 The NAB Score for the proposed model with different PermanenceDecrement values

Once the NAB score of the proposed model is computed using the optimised parameter settings, next its performance is evaluated using a traditional metric, F-measure beside recall and precision scores. Results are illustrated in Table 6-4.

Table 6-4 Proposed model results

<b>NAB score (%)</b>	<b>F-measure (%)</b>	<b>Recall (%)</b>	<b>Precision (%)</b>
83	83	87	70

The proposed model with the default parameter settings recommended by the NuPIC shows a better performance than the original HTM model, the average NAB score of 64% compared to the performance of the HTM model with the NAB score of 62%. The proposed model even shows a much better performance using the optimised parameter settings by achieving the NAB score of 83%. Furthermore, a noticeable increment in the performance of the proposed model is observed in recorded values for F-measure, recall and precision in comparison with equivalent measures from the HTM model with default parameter settings (recommended by NuPIC).

The Listing 6-2 and Listing 6-3 illustrate the optimised parameter settings for the level 1 and level 2 respectively which the proposed model has produced the best results including NAB score and F-measure.

---

```

SpatialPooler (
    inputDimensions = (224,),
    columnDimensions = (600,),
    synPermConnected = 0.1,
    numActiveColumnsPerInhArea = 12.0,
    boostStrength = 1.0,
    synPermActiveInc = 0.01,
    synPermInactiveDec = 0.005,
    globalInhibition = True,
    potentialPct = 0.85,
    spVerbosity = 0,
    minPctOverlapDutyCycle = 0.001,
    dutyCyclePeriod = 1000,
    localAreaDensity = -1.0,
    potentialRadius = 16,
    stimulusThreshold = 0),

TemporalMemory (
    columnDimensions = (600,),
    cellsPerColumn= 32,
    initialPermanence = 0.21,
    connectedPermanence = 0.5,
    maxNewSynapseCount = 20,
    permananceIncrement = 0.1,
    permananceDecrement = 0.1,
    minThreshold = 12,
    activationThreshold = 16,
    maxSegmentsPerCell =200,
    maxSynapsesPerSegment =26,
    predictedSegmentDecrement = 0.0)

```

---

Listing 6-2 The best parameters found for the proposed model – level 1

```

SpatialPooler (
    inputDimensions = (884,),
    columnDimensions = (2048,),
    synPermConnected = 0.1,
    numActiveColumnsPerInhArea = 20.0,
    boostStrength = 3.0,
    synPermActiveInc = 0.01,
    synPermInactiveDec = 0.0005,
    globalInhibition = True,
    potentialPct = 0.85,
    spVerbosity = 0,
    minPctOverlapDutyCycle = 0.001,
    dutyCyclePeriod = 1000,
    localAreaDensity = -1.0,
    potentialRadius = 16,
    stimulusThreshold = 0),

TemporalMemory (
    columnDimensions = (2048,),
    cellsPerColumn= 32,
    initialPermanence = 0.21,
    connectedPermanence = 0.5,
    maxNewSynapseCount = 20,
    permananceIncrement = 0.1,
    permananceDecrement = 0.1,
    minThreshold = 12,
    activationThreshold = 16,
    maxSegmentsPerCell =128,
    maxSynapsesPerSegment =28,
    predictedSegmentDecrement = 0.0)

```

Listing 6-3 The best parameters found for the proposed model – level 2

### 6.2.2 k-Nearest Neighbour

The k-Nearest Neighbour (k-NN) is tested with different values of  $k$  and their results are illustrated in Figure 6-9. The value of  $k$  has an impact on the accuracy of the model and according to (Rajkumar and Reena, 2010), larger values for  $k$  is usually selected to reduce the effect of noise on the classification. In this section values ranging from 2-32 are tested for  $k$ . As a part of the testing, the Principal Component Analysis (PCA) is also used for dimension reduction of obtained parameters in order to observe the behaviour the k-NN with PCA and without it.

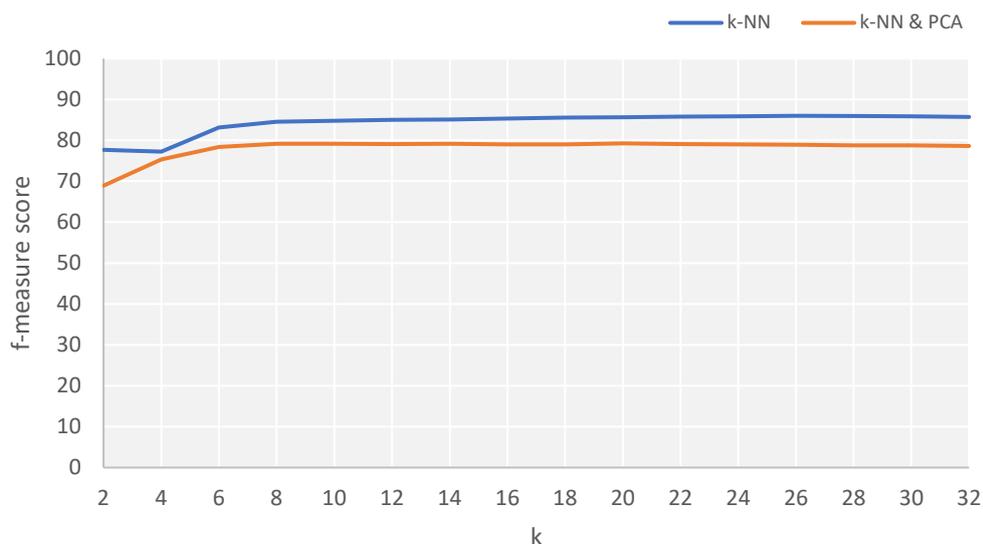


Figure 6-9 k-NN model F-measure results

The results show that the k-NN performs better without PCA. The outcome of the testing also shows that the k-NN on its own by using multiple vital signs has a better performance compared to when it is used with the PCA.

The k-NN has the best performance when  $k$  is set to 28 and the k-NN with PCA has produce the best performance when  $k$  was set to 20. Once the best F-measure for the k-NN with and without PCA is selected, then same parameter settings are used to compute the NAB scores. Table 6-5 illustrates the overall results for the k-NN algorithm with and without PCA.

Table 6-5 k-NN model results

	<b>NAB score (%)</b>	<b>F-measure (%)</b>	<b>Recall (%)</b>	<b>Precision (%)</b>
<b>k-NN</b>	68	86	90	82
<b>k-NN &amp; PCA</b>	62	78	78	80

### 6.2.3 Influence Outlierness

Two models using Influence Outlierness (INFLO) algorithm with Singular Value Decomposition (SVD) and without it are tested. The SVD is similar to PCA and it

is used to reduce the number of obtained attributes by removing unnecessary attributes. The results are shown in Figure 6-10 for both models.



Figure 6-10 INFLO model f-measure results for different k-values

The parameter settings for both models that scored best F-measure are selected to compute their NAB scores. The INFLO has shown the best performance when *k* is set to 26 and it produced the best performance with SVD when *k* is set to 26. The overall result illustrated in Table 6-6 indicates that INFLO algorithm with SVD achieved a better performance.

Table 6-6 INFLO model results

	NAB score (%)	F-measure (%)	Recall (%)	Precision (%)
<b>INFLO</b>	60	80	78	81
<b>INFLO &amp; SVD</b>	67	89	100	80

### 6.2.4 Histogram-Based Outlier Score

In this section, the Histogram-Based Outlier Score (HBOS) algorithm is tested and its performance is shown in Table 6-7.

Table 6-7 HBOS model results

	<b>NAB score (%)</b>	<b>F-measure (%)</b>	<b>Recall (%)</b>	<b>Precision (%)</b>
<b>HBOS</b>	67	41	34	85

### 6.2.5 Support Vector Machine

In this section, performance of one-class Support Vector Machine (SVM) is evaluated. The one-class type of SVM is used in here as the data row within the extracted datasets are labelled and classed to *anomaly* and *normal*. Hence this type of SVM is used to check the performance of SVM when it learns from one class of examples and later on test if new examples match the known ones. However, one class SVM was not able to perform well to detect the anomalies in a batch-processed fashion and it could only exceed the 34% NAB accuracy mark.

Table 6-8 SVM model results

	<b>NAB score (%)</b>	<b>F-measure (%)</b>	<b>Recall (%)</b>	<b>Precision (%)</b>
<b>HBOS</b>	34	0.54	0.27	90

## 6.3 Results Analysis and Discussion

This section discusses and analyses the anomaly detection performance of the nine algorithms and techniques that are experimented and tested in Section 6.2. The performance evaluation has been carried out using both an on-line learning and a traditional learning fashion. The NAB is used for on-line learning and other performance metrics such as recall, precision and F-measure are used as a traditional evaluation technique (for the algorithms that use batch processing). Figure 6-11 illustrates the NAB and F-measure scores for all evaluated models show that the proposed model outperforms state-of-the-art algorithms in on-line learning fashion.

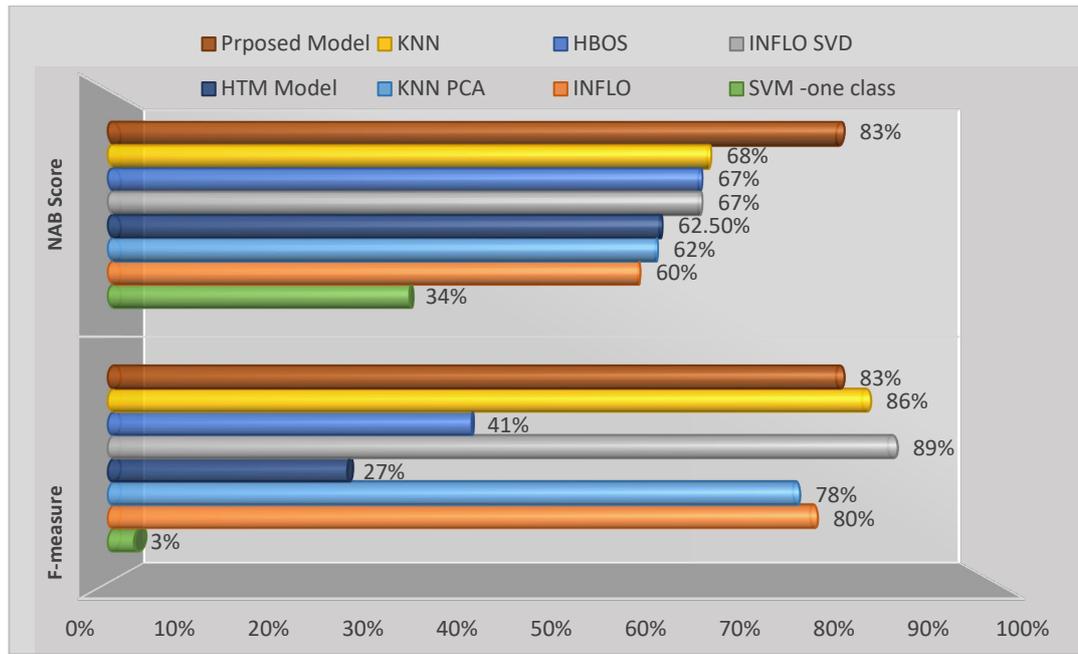


Figure 6-11 The Results for all evaluated models

The proposed framework achieved 83% NAB score which outperforms the HTM and k-NN algorithms by 15%, the HBOS and INFLO SVD by 16% and the k-NN PCA by 21% while the SVM scored 34%. The results prove that multiple HTM networks can achieve better performance when dealing with multi-dimensional data i.e. data collected from more than one source/sensor. The patterns learned from the data stream at each level of the hierarchy and the semantically improved representation of the data are reused when combined in novel ways at higher levels.

The results show that the proposed model in traditional fashion can get a relatively good accuracy on detecting abnormal patterns of the vital signs. The proposed model has outperformed the current HTM model by 56%, however the k-NN and INFLO SVD algorithms performed slightly better than the proposed model. It is important to note that the training phases of the proposed model and other models using batch processing are different. In the training phase, the cases are sent to the proposed model one by one where all the instance data are sent to the other algorithms that use batch processing all together.

Also, it is important to note that a core feature of the AAL health monitoring scenario is to detect anomalies as early as possible which the proposed model has

achieved this and it has shown a better performance compared to other algorithms. The proposed model has been tested in a traditional fashion to compare its performance with the batch processing algorithms and to get a better insight of the online learning algorithms when it is used in an off-line mode.

The result also shows that in the off-line mode, the algorithms such as k-NN and INFLO SVD which are using global and local anomaly detection methods respectively have a slightly better performance compared to algorithms such as the HTM that are abstracting temporal patterns of data points.

## 6.4 Summary

The work presented in this Chapter comprised of two core phases presenting the experimental design, and the performance evaluation of the model developed in Chapter 4. A detailed explanation of the experimental design and the performance comparison of the proposed model with several state-of-the-art anomaly detection algorithms are discussed and documented.

Additionally, a context model is described to specify the age group, gender and medical condition in order to define a threshold for normal and abnormal values of vital signs that were fed to the algorithms. A total of 247 datasets extracted from MIMIC-III in Chapter 5 were split into learning and inference sets including records of the patients who have survived during their ICU stays and patients who have died during their ICU stays. Each dataset consists of 24 hours of vital signs readings that are used for learning and inference steps.

As a secondary contribution of this thesis, the performance evaluation is carried out where both an on-line learning and a traditional learning fashion are applied. The Numenta Anomaly Benchmark (NAB) is used for on-line learning and other performance metrics such as recall, precision and F-measure are used as a traditional evaluation technique for the algorithms that use batch processing. The behaviour of the proposed model was reviewed through the optimisation of its parameter settings.

The results of several anomaly detection algorithms and techniques are also analysed and discussed on the extracted datasets. The proposed framework achieved 83% NAB score which outperforms the HTM and k-NN algorithms by 15%, the HBOS and INFLO SVD by 16% and the k-NN PCA by 21% while the SVM scored 34%. The results prove that multiple HTM networks can achieve better performance when dealing with multi-dimensional data i.e. data collected from more than one source/sensor. The patterns learned from the data stream at each level of the hierarchy and the semantically improved representation of the data are reused when combined in novel ways at higher levels.

Indirectly, this Chapter also brings together the model of the artefact developed through this thesis. We have had an opportunity to review the behaviour of the proposed model through optimisation of its parameter settings. The results presented in this Chapter have satisfy the requirements of the on-line learning model in AAL health monitoring scenario.

## Chapter 7

# CONCLUSIONS AND FUTURE WORK

This thesis has presented a novel framework inspired by the HTM theory to monitor the health condition of an elderly patient by detecting abnormal patterns of several vital signs. The framework incorporates different functions of CLA which enables continual learning and recognises new and changed behaviour without manual intervention. The detection of the abnormal patterns in vital signs is essential in health monitoring and assistance scenario, and it is driven by improving care and survival of the elderly patients who are living independently.

In recent years, several ML approaches have attempted to detect anomalies in a single or multiple vital sign including the BP, HR and SpO<sub>2</sub>. To the author's knowledge, the proposed framework is the first implementation of HTM hierarchy with multiple regions applied to model the spatio-temporal patterns of multiple vital signs collected from elderly people to prevent a cardiac arrest. The vital signs, or in general clinical variables measured at different time moments, are usually in numeric or integer formats, and are irregularly sampled in time. In addition, there are several modifiable and non-modifiable factors that must be considered when modelling the patterns of the vital signs. More generally, what

is considered as an anomaly for a patient, might not be an anomaly for another patient, consequently a fixed threshold will not be an ideal solution.

The research investigation motivated by two hypothesis and five research questions to investigate these hypotheses.

**HYP 1:** The hierarchical and temporal principles of the HTM theory would outperform the accuracy of current state-of-the-art machine learning algorithms in detecting anomalous points preceding a health critical event.

**HYP 2:** HTM theory can be used in AAL health monitoring scenario to prevent a health critical event by modelling the patterns of vital signs.

The following research questions were formulated to investigate these hypotheses:

**RQ1:** Can HTM be applicable for detecting anomalous points preceding a health critical event by modelling spatio-temporal patterns of multiple vital signs?

**RQ2:** How hierarchical organisation of the HTM improves abstraction of semantical information from multiple vital signs?

**RQ3:** What are the Machine Learning techniques and algorithms that can be applied in health monitoring scenario to prevent a critical health event?

**RQ4:** Does the use of temporal information improve the anomaly detection in health monitoring scenarios?

**RQ5:** What are the data mining tasks required to prepare datasets for the evaluation of Machine Learning algorithms in the context of AAL health monitoring?

In Chapter 2, a literature review is conducted to answer RQ3, RQ4 and RQ5. Chapter 2 began with a research into the principles underlying the AmI paradigm providing the health care services, to explore the complexity of diagnosis and prognosis in healthcare where multiple vital sign parameters are obtained from the patients. Previous works in relation to presentation of the temporal

dimension of data and correlations among multiple time series data are studied. The research revealed that normally a single vital sign does not contain sufficient information to be used for detecting anomalies, hence, to overcome this issue, the correlation between multiple vital signs can be used. Further research in the field of healthcare confirmed the noticeable changes in vital signs including BP, HR are occurring prior to a critical event and are usually used by the professional caregivers to diagnose symptoms and to check if the patients have a medical condition. From this research the basis of a model of temporal patterns was devised and used in the proposed framework in order to discover patient's specific anomalies independently. Further research was conducted on ML techniques and algorithms in context of health diagnosis and prognosis. Current ML techniques applied in health monitoring scenarios fall short from reaching the full potential of anomaly detection. This is due to a number of facts that AAL in health monitoring scenario must: (i) be adaptable to the new normal patterns, (ii) apply contextual anomaly detection approach, to detect anomalous data points, (iii) detect an anomalous point as early as possible preceding a cardiac arrest and (iv) employ a common learning technique that can be used for clinical decision support to discover patient specific anomalies independently.

More specifically, existing algorithms are commonly applying statistical, nearest neighbour, and density-based approaches where time dimension is not considered.

Chapter 3 contained a further research to answer RQ1 and RQ2. This Chapter described the HTM theory inspired by recent advances in neuroscience and the interaction of pyramidal neurons in the neocortex of mammalian brains. The HTM is a system based on memory, it consists of different regions that are structured in a hierarchy. This research explored the role of the regions which are the main elements of memory, and typically are located and organised in a level of the HTM network. More specifically, a region receives and sends information up and down the hierarchy. As information ascends the hierarchy, the output of regions in lower levels will be converged, and the regions on higher levels of hierarchy make use of the information to produce a bigger picture of an outside world which makes the learning and prediction of HTM possible.

In fact, it is noticed that the HTM takes into consideration three factors that are crucial for intelligence: time, feedback and the physical hierarchical structure of the brain. Time is an important parameter particularly in AAL health monitoring scenario and it also plays a very important role in the HTM for learning, inference and prediction.

Chapter 3 also contained an exploration of the algorithmic implementation of the HTM theory, the CLA. The CLA are a set of algorithms that implements functions of the HTM theory where three representations of the SDR in a region can be computed when applying the encoder, SP and TM algorithms. The findings of this research on the CLA revealed that the input to the HTM network, are encoded to the SDR using an appropriate encoder. Input data to an HTM region can come from a sensory data or from another region lower in the hierarchy. There are numerous types of encoders available which can be used for encoding different type of data. The SDR created by an encoder are fed to the SP algorithm. The SP algorithm operates on the columns level of the region which the HTM learns the spatial patterns of the input space. Next, the SDR from the SP are sent to the TM algorithm where it operates on active columns by activating and deactivating the cells within the columns to learn transition between spatial patterns. The output of the TM algorithm could be used by different types of anomaly detection algorithms or classification algorithms to detect and predict unexpected patterns respectively.

These features of the HTM theory make this state-of-the-art algorithm adaptive to different modalities, universal in using the same learning principles, enables continual learning and recognises new and changed behaviours without manual intervention.

However, the current implementation of the HTM does not have a hierarchy of regions and only implement one region. Hence, as a contribution of this thesis, a novel framework is designed and implemented in Chapter 4 by proposing an efficient hierarchy of regions and by arranging of the regions within the HTM network.

Chapter 4 reflects on the work conducted in Chapter 2 and 3. This Chapter described the design and development of a novel and adaptive framework by applying the HTM theory and ideas underlying combination of hierarchy networks in HTM. This Chapter covered the challenges in monitoring health condition of elderly people in AAL scenario to prevent critical events.

A number of challenges were tackled in this Chapter to ensure the online capability of the proposed framework to learn and adapt to new normal patterns, while abstracting the semantics of vital signs through regions of TM and SP algorithms. In addition, the two layered hierarchy is proposed to detect abnormal patterns in vital signs by modelling temporal and correlation patterns between the vital signs. Four regions are constructed in level-one, to abstract temporal pattern of individual vital signs rather than applying a series of concatenated multiple vital signs (HTM implementation of one region). In addition, to detect an anomalous point as early as possible (10 hours preceding cardiac arrest), the high order sequence memory was applied to enable the learning of long-range temporal dependencies and statistic for a particular vital sign that can appear in different temporal contexts (time of day).

As a result, a two-layered hierarchy framework consists of encoders, SP, TM was proposed to suite the requirements of the AAL in particular the proposed scenario to prevent a cardiac arrest.

Chapter 5 described the approaches that are commonly used in ML literature for data acquisition task such as the experimental wearable sensor data, medical databases, and simulated sensor data. As a secondary contribution of this thesis, a set of datasets is generated to be used for training and testing of the proposed framework. More specifically, 7 tables out of 26 tables from MIMIC-III database are selected for further data processing. The relationship between tables was used to filter and select 5 parameters including: timestamp, HR, SBP, DBP and SpO<sub>2</sub>. A total record of 247 older patients are selected who are diagnosed with hypotension condition. In addition, it was ensured that records of distinctive patients consist of the measurement of the vital signs for at least 24 hours and they have balanced records of patients of different gender (50% male, 50%

female). The generated datasets are used in Chapter 6 for experimentation to evaluate the performance of the proposed framework and to compare its performance with several state-of-art algorithms.

Chapter 6 described the experimentations that were conducted to test the hypothesis of this thesis that whether the HTM theory is able to improve the accuracy of anomaly detection in multiple vital signs patterns.

The present findings confirm that a core requirement of the AAL health monitoring application is to enable early detection of abnormalities through real-time patient monitoring to prevent a critical event. On this basis, the performance of the proposed framework was evaluated and compared with various state-of-the-art algorithms using NAB as an anomaly detection metric.

The statistical analysis of data indicated that the proposed framework achieved 83% NAB score which outperforms the HTM and k-NN algorithms by 15%, the HBOS and INFLO SVD by 16% and the k-NN PCA by 21% while the SVM scored 34%. The results prove that multiple HTM networks can achieve better performance when dealing with multi-dimensional data i.e. data collected from more than one source/sensor.

The patterns learned from the data stream at each level of the hierarchy and the semantically improved representation of the data are reused when combined in novel ways at higher levels. On this basis, the results support the hypothesis.

## 7.1 Contribution to Knowledge

The following list is a summary of the contributions to knowledge achieved in this thesis.

- A novel HTM based framework to prevent the cardiac arrest by early detection of abnormal patterns of vital signs for elderly patient's health monitoring and assistance. The novel framework consists of two-layer HTM networks consists of the layer one for modelling the temporal patterns of four vital signs, including: SBP, DBP, HR and SpO<sub>2</sub>. Also, a higher layer to detect abnormal temporal spatial patterns of the anomalies detected from the regions in layer 1.
- 247 datasets of older patients are generated from MIMIC-III database consists of the measurement of multiple vital signs for at least 24 hours which it has a balanced record of patients for different genders (50% male).
- Performance testing of the proposed framework and various state-of-the-art algorithm using both an on-line and a traditional anomaly detection metrics. The NAB was used for on-line learning and metrics such as recall, precision and F-measure were used to test the performance of the algorithms that use batch processing techniques.

## 7.2 Future Work

The encoder is one of the important components of the HTM network that converts the native format of data (i.e. datetime, number, category) into a binary SDR. However, the current encoders are developed for specific data types without considering the application domain and scenario. In AAL health monitoring application, time dimension for vital signs is extracted to give a different semantical meaning of similar data points that occur in different times of day. Hence, future research should consider the potential effects of encoders more carefully on the accuracy of HTM in anomaly detection of vital signs. For instance,

an encoder can be developed that creates an SDR for a single vital sign by encoding and concatenating vital signs with a time-stamp. This potentially results in semantically improved representation of data that can be used by SP and TM for the anomaly detection task.

Future studies could investigate the appropriate classifiers in the AAL health monitoring scenario to generate and raise an alarm to the caregiver to react to patient health degradation. Hence the proposed HTM-based framework can be expanded to use a classifier for classifying the risk level of detected anomalies according to the user preferences and degree of sudden changes in vital signs. Commonly health caregivers in AAL health monitoring scenarios are referred to GP, nurse, healthcare assistant, family members and relatives of the elderly patient who have different roles and characteristics. Hence adding an alarm feature to AAL health monitoring application potentially improve the quality of the health service by sending alarms to caregivers in real-time and automatically recommends corresponding care guidelines so that appropriate action can be taken.

The lifestyle of patients is influenced by a number of factors such as weekly alcohol consumption, Body Mass Index (BMI), smoking, regular exercise which can be monitored and modified to keep track of their health status. The lifestyle changes can associate to symptoms and can be effective in a threshold value of the vital signs such as BP and HR. Hence, it will be important that future work considers a different blend of vital signs and to consider the impact of risk factors on anomaly detection in AAL health monitoring application.

Future work apart from looking for spatiotemporal patterns of the vital sign for anomaly detection, it should look for patterns of patient's daily activity (e.g. sleeping, walking, exercising), as changes in daily activity associate to potential changes in vital signs. Such a piece of knowledge is valuable in creating a context model that improve the performance of the proposed HTM-based framework by avoiding false detection of anomalies (misdiagnosis).

More work is required to create datasets that capture complex patterns of vital signs to evaluate the ability of the proposed framework to recognise new normal

patterns of vital signs. It is common that the normal patterns of vital signs change over time due to changes to the lifestyle of patients. Hence, more data are required that incorporate various changing normal patterns of vital signs to test the ability of the proposed framework to recover and recognise the new normal patterns.

For closer biological likelihood, to investigate the feasibility of adding a feedback channel to the proposed framework. Adding the ability of top-down process could potentially improve the distribution of information throughout the networks.

# REFERENCES

Aggarwal, C. C. (2014) *Managing and mining sensor data, Managing and Mining Sensor Data*. doi: 10.1007/978-1-4614-6309-2.

Ahmad, S. and Hawkins, J. (2015) ‘Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory’, *CoRR*, abs/1503.0, p. 1503.07469. doi: 10.1063/1.4918346.

Ahmad, S. and Hawkins, J. (2016) ‘How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites’, pp. 1–23. Available at: <http://arxiv.org/abs/1601.00720>.

Ahmad, S. and Lewis, M. (2017) ‘Temporal Memory Algorithm Implementation and Pseudocode’.

Al-Shaqi, R., Mourshed, M. and Rezgui, Y. (2016) ‘Progress in ambient assisted systems for independent living by the elderly.’, *SpringerPlus*. Springer International Publishing, 5, p. 624. doi: 10.1186/s40064-016-2272-8.

Alotaibi, N. N. and Sasi, S. (2015) ‘Predictive Model for Transferring Stroke In-Patients to Intensive Care Unit’, pp. 848–853.

Altun, K., Ã, B. B. and Tunc, O. (2010) ‘Comparative study on classifying human activities with miniature inertial and magnetic sensors’, 43, pp. 3605–3620. doi: 10.1016/j.patcog.2010.04.019.

Amiribesheli, M., Benmansour, A. and Bouchachia, A. (2015) 'A Review of Smart Homes in Healthcare', *Journal of Ambient Intelligence and Humanized Computing*, 6(4), pp. 495–517.

Analysis, S. C. et al. (2017) Hospital Adult Critical Care Activity.

Atallah, L., Lo, B. and Yang, G. Z. (2012) 'Can pervasive sensing address current challenges in global healthcare?', *Journal of Epidemiology and Global Health*. Ministry of Health, Saudi Arabia, 2(1), pp. 1–13. doi: 10.1016/j.jegh.2011.11.005.

Augusto, J. C. (2005) 'Temporal reasoning for decision support in medicine', *Artificial Intelligence in Medicine*, 33(1), pp. 1–24. doi: 10.1016/j.artmed.2004.07.006.

Avci, A. et al. (2010) 'Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey', *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pp. 1–10. Available at: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5759000>.

Ayadi, A. et al. (2017) 'Outlier detection approaches for wireless sensor networks : A survey'. Elsevier B.V., 129, pp. 319–333. doi: 10.1016/j.comnet.2017.10.007.

Banaee, H., Ahmed, M. U. and Loutfi, A. (2013) 'Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges.', *Sensors (Basel, Switzerland)*, 13(12), pp. 17472–17500. doi: 10.3390/s131217472.

Banaee, H., Ahmed, M. U. and Loutfi, A. (2015) 'Descriptive modelling of clinical conditions with data-driven rule mining in physiological data', *HEALTHINF 2015 - 8th International Conference on Health Informatics, Proceedings; Part of 8th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2015*, (Healthinf), pp. 103–113. doi: 10.5220/0005220901030113.

Batal, I. et al. (2011) 'A pattern mining approach for classifying multivariate temporal data', *Proceedings - 2011 IEEE International Conference on*

*Bioinformatics and Biomedicine*, *BIBM* 2011, pp. 358–365. doi: 10.1109/BIBM.2011.39.

Benetis, R. *et al.* (2006) ‘Nearest and reverse nearest neighbor queries’, *The VLDB Journal*, 15, pp. 229–250. doi: 10.1007/s00778-005-0166-4.

Blackburn, R. *et al.* (2018) ‘Statin prescribing for prevention of cardiovascular disease amongst people with severe mental illness : Cohort study in UK primary care’, *Schizophrenia Research*. The Authors, 192, pp. 219–225. doi: 10.1016/j.schres.2017.05.028.

Bobak, I. (2017) ‘HIERARCHICAL TEMPORAL MEMORY FOR REAL-TIME ANOMALY DETECTION’.

Bose, R. (2009) ‘Sensor networks motes, smart spaces, and beyond’, *Pervasive Computing*, *IEEE*, pp. 84–90. Available at: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5165565](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5165565) (Accessed: 11 November 2013).

Bouckaert, R. R. and Frank, E. (2004) ‘Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms’, in Dai, H., Srikant, R., and Zhang, C. (eds) *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–12.

Casiglia, E., Tikhonoff, V. and Mazza, A. (2005) ‘Systolic and pulse hypertension’, 1, pp. 85–94.

Celler, B. G. and Sparks, R. S. (2015) ‘Home telemonitoring of vital signs--technical challenges and future directions.’, *IEEE journal of biomedical and health informatics*, 19(1), pp. 82–91. doi: 10.1109/JBHI.2014.2351413.

Chandola, V., Banerjee, A. and Kumar, V. (2009) ‘Anomaly Detection: A Survey’, *ACM Computing Surveys*, 41. doi: 10.1145.

Chen, Z. *et al.* (2018) ‘Development of a personalized diagnostic model for kidney stone disease tailored to acute care by integrating large clinical , demographics

and laboratory data : the diagnostic acute care algorithm - kidney stones ( DACA-KS )'. *BMC Medical Informatics and Decision Making*, pp. 1–14.

Chester, J. G. and Rudolph, J. L. (2011) 'Vital Signs in Older Patients: Age-Related Changes', *Journal of the American Medical Directors Association*, 12(5), pp. 337–343. doi: 10.1016/j.jamda.2010.04.009.

Churpek, M. *et al.* (2013) 'Derivation of a cardiac arrest prediction model using ward vital signs', *PMJ*, 40(7), pp. 2102–2108. doi: 10.1097/CCM.obo13e318250aa5a.Derivation.

Clement DL, De Buyzere M, De Bacquer DA, et al (2003) 'Prognostic value of ambulatory blood pressure recordings in patients with treated hypertension', *The new england journal of medicine original*, 348, pp. 2407–15.

Cook, D. J., Augusto, J. C. and Jakkula, V. R. (2009) 'Ambient intelligence: Technologies, applications, and opportunities', *Pervasive and Mobile Computing*. Elsevier B.V., 5(4), pp. 277–298. doi: 10.1016/j.pmcj.2009.04.001.

Cui, Y., Ahmad, S. and Hawkins, J. (2016) 'Continuous Online Sequence Learning with an Unsupervised Neural Network Model', *Neural Computation*, 28(11), pp. 2474–2504. doi: 10.1162/NECO\_a\_00893.

Cui, Y., Ahmad, S. and Hawkins, J. (2017) 'The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding', *Frontiers in Computational Neuroscience*, 11(November), pp. 1–15. doi: 10.3389/fncom.2017.00111.

Devineni, A. (2015) *How our brains learn*. Available at: <http://www.brains-explained.com>.

Epidemiology, C. (2016) 'An algorithm for identification and classification of individuals with type 1 and type 2 diabetes mellitus in a large primary care database', pp. 373–380.

Fischer, A. and Mougeot, M. (2018) 'Aggregation using input–output trade-off', *Journal of Statistical Planning and Inference*. Elsevier B.V., 200, pp. 1–19. doi: 10.1016/j.jspi.2018.08.001.

Forkan, A. R. M. *et al.* (2015) 'A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living', *Pattern Recognition*. Elsevier, 48(3), pp. 628–641. doi: 10.1016/j.patcog.2014.07.007.

Forkan, A. R. M. and Khalil, I. (2016) 'A probabilistic model for early prediction of abnormal clinical events using vital sign correlations in home-based monitoring', *2016 IEEE International Conference on Pervasive Computing and Communications, PerCom 2016*. doi: 10.1109/PERCOM.2016.7456519.

Forkan, A. R. M. and Khalil, I. (2017a) '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*. Elsevier Ireland Ltd, 139, pp. 1–16. doi: 10.1016/j.cmpb.2016.10.018.

Forkan, A. R. M. and Khalil, I. (2017b) 'PEACE-Home: Probabilistic estimation of abnormal clinical events using vital sign correlations for reliable home-based monitoring', *Pervasive and Mobile Computing*. Elsevier B.V., 38, pp. 296–311. doi: 10.1016/j.pmcj.2016.12.009.

Gennings, C., Ellis, R. and Ritter, J. K. (2012) 'Linking empirical estimates of body burden of environmental chemicals and wellness using NHANES data', *Environment International*. Elsevier Ltd, 39(1), pp. 56–65. doi: 10.1016/j.envint.2011.09.002.

Ghayvat, H. *et al.* (2019) 'Smart Aging System : Uncovering the Hidden', *sensors*. doi: 10.3390/s19040766.

Goldstein, M. and Dengel, A. (2012) 'Histogram-based Outlier Score ( HBOS ): A fast Unsupervised Anomaly Detection Algorithm', (1), pp. 59–63.

- Goldstein, M. and Uchida, S. (2016) 'A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data', (April), pp. 1–31. doi: 10.7910/DVN/OPQMVF.
- Grguri, A., Mošmondor, M. and Huljeni, D. (2019) 'The SmartHabits: An Intelligent Privacy-Aware c', *Sensors*. doi: 10.3390/s19040907.
- Gustafsson–sebg209, S. (2011) 'Hierarchical Temporal Memory', *Ida.Liu.Se*, pp. 1–68. doi: 10.1109/IEMBS.2006.260909.
- Hamerly, G. and Elkan, C. (2004) 'Learning the k in kmeans', *Advances in neural information processing ...*, 17, pp. 1–8. doi: 10.1.1.9.3574.
- Hands, C. *et al.* (2013) 'Patterns in the recording of vital signs and early warning scores: compliance with a clinical escalation protocol', *BMJ Qual Saf*, 22(9), pp. 719–726. doi: 10.1136/bmjqs-2013-001954.
- Haque, S., Rahman, M. and Aziz, S. (2015) 'Sensor Anomaly Detection in Wireless Sensor Networks for Healthcare', *Sensors*, 15(4), pp. 8764–8786. doi: 10.3390/s150408764.
- Harper, P. R. (2005) 'A review and comparison of classification algorithms for medical decision making', *Health Policy*, 71(3), pp. 315–331. doi: 10.1016/j.healthpol.2004.05.002.
- Hawkins, J. and Ahmad, S. (2016) 'Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex', *Frontiers in Neural Circuits*, 10(March), pp. 1–13. doi: 10.3389/fncir.2016.00023.
- Hawkins, J., Ahmad, S. and Cui, Y. (2017) 'A Theory of How Columns in the Neocortex Enable Learning the Structure of the World', *Frontiers in Neural Circuits*, 11(October), pp. 1–18. doi: 10.3389/fncir.2017.00081.
- Hawkins, J., Ahmad, S. and Dubinsky, D. (2011) 'Cortical Learning Algorithm and Hierarchical Temporal Memory', *Numenta Whitepaper*, pp. 1–68. doi: 10.1109/IEMBS.2006.260909.

Hawkins, J. and Blakeslee, S. (2005) 'On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines.', *Times Books*, pp. 1–174.

He, Z. *et al.* (2017) 'Comprehensive transcriptome analysis of neocortical layers in humans, chimpanzees and macaques', *Nature Neuroscience*. Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., 20, p. 886. Available at: <https://doi.org/10.1038/nn.4548>.

Huaman, M. A. *et al.* (2015) 'The Decade of the Dendritic NMDA Spike', 143(5), pp. 951–959. doi: 10.1017/S0950268814002131.Tuberculosis.

Iram, S. *et al.* (2012) 'Computational Data Analysis for Movement Signals Based on Statistical Pattern Recognition Techniques for Neurodegenerative Diseases', *13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcastin*. Available at: [https://www.academia.edu/2958098/Computational\\_Data\\_Analysis\\_for\\_Movement\\_Signals\\_Based\\_on\\_Statistical\\_Pattern\\_Recognition\\_Techniques\\_for\\_Neurodegenerative\\_Diseases](https://www.academia.edu/2958098/Computational_Data_Analysis_for_Movement_Signals_Based_on_Statistical_Pattern_Recognition_Techniques_for_Neurodegenerative_Diseases).

J.F.Allen (1984) 'Towards a general theory of action and time', *Artificial Intelligence*, 23, pp. 123–154. doi: 10.1016/0004-3702(84)90008-0.

Jih, W., Hsu, J. Y. and Tsai, T. (2006) 'Context-Aware Service Integration for Elderly Care in A Smart Environment', *American Association for Artificial Intelligence*.

Johnson, A. E. W. *et al.* (2016) 'Data Descriptor : MIMIC-III , a freely accessible critical care database', pp. 1–9.

Johnson, M. E. *et al.* (2016) 'Early real-world evidence of persistence on oral anticoagulants for stroke prevention in non-valvular atrial fi brillation : a cohort study in UK primary care'. doi: 10.1136/bmjopen-2016-011471.

Kaur, H. and Wasan, S. K. (2006) 'Empirical Study on Applications of Data Mining Techniques in Healthcare', *(B3) Journal of Computer Sciences*, 2(2), pp. 194–200. doi: 10.3844/jcssp.2006.194.200.

Khusainov, R. *et al.* (2013) 'Real-time human ambulation, activity, and physiological monitoring: taxonomy of issues, techniques, applications, challenges and limitations.', *Sensors (Basel, Switzerland)*, 13(10), pp. 12852–12902. doi: 10.3390/s131012852.

Kim, S. H. *et al.* (2016) 'HeartCast: Predicting acute hypotensive episodes in intensive care units', *Statistical Methodology*. Elsevier B.V., 33, pp. 1–13. doi: 10.1016/j.stamet.2016.07.001.

Ko, J., Lu, C. and Srivastava, M. (2010) 'Wireless sensor networks for healthcare', *Proceedings of the ...*, 98(11). Available at: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5570866](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5570866) (Accessed: 2 May 2014).

Kofod-Petersen, A. (2007) A Case-Based Approach to Realising Ambient Intelligence among Agents.

Kumari, M. and Godara, S. (2011) 'Comparative Study of Data Mining Classification Methods in Cardiovascular Disease Prediction', *Ijcst*, 4333, pp. 304–308. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.219.6038>.

Kuryloski, P. *et al.* (2008) 'DexterNet: An Open Platform for Heterogeneous Body Sensor Networks and Its Applications'.

Kuryloski, P. and Giani, A. (2009) 'DexterNet: An open platform for heterogeneous body sensor networks and its applications', ... *Sensor Networks*, ... Available at: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5226911](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5226911) (Accessed: 11 November 2013).

Kyriacos, U. *et al.* (2014) 'Monitoring vital signs: Development of a Modified Early Warning Scoring (Mews) system for general wards in a developing country', *PLoS ONE*, 9(1). doi: 10.1371/journal.pone.0087073.

Lavin, A. and Ahmad, S. (2015) 'Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark', *14th International Conference on Machine Learning and Applications (IEEE ICMLA'15)*.

Lavrač, N. (1999) 'Selected techniques for data mining in medicine', *Artificial Intelligence in Medicine*, 16(1), pp. 3–23. doi: 10.1016/S0933-3657(98)00062-1.

Lehman, L. H. *et al.* (2013) 'Methods of Blood Pressure Measurement in the ICU\*', *Critical Care Medicine*, 41(1), pp. 34–40. doi: 10.1097/CCM.0b013e318265ea46.

Lewin, D. *et al.* (2010) 'Assisted living technologies for older and disabled people in 2030', 44(March).

Lewis, M. *et al.* (2018) 'Locations in the Neocortex: A Theory of Sensorimotor Object Recognition Using Cortical Grid Cells', *bioRxiv*, p. 436352. doi: 10.1101/436352.

Liu, F. and Chang, H. (2017) 'Physiological links of circadian clock and biological clock of aging', *Protein & Cell*. Higher Education Press, 8(7), pp. 477–488. doi: 10.1007/s13238-016-0366-2.

Liu, J. *et al.* (2015) 'Tracking Vital Signs During Sleep Leveraging Off-the-shelf WiFi', *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc '15*, pp. 267–276. doi: 10.1145/2746285.2746303.

McNaull, J. *et al.* (2012) 'Data and Information Quality Issues in Ambient Assisted Living Systems', *Journal of Data and Information Quality*, 4(1), pp. 1–15. doi: 10.1145/2378016.2378020.

Mike, R. and Bunker, J. (2014) 'Hypertension: diagnosis, assessment and management', 28(42), pp. 50–59.

Mitri, N. *et al.* (2017) 'Irregular Breathing Detection in CPAP Assisted Patients Using Hierarchical Temporal Memory', *IEEE*, pp. 1–6.

Moody, G. B. and Lehman, L. H. (2009) 'Predicting acute hypotensive episodes: The 10th annual PhysioNet/Computers in Cardiology Challenge', *Computers in Cardiology, 2009*, pp. 541–544. doi: 10.1126/scisignal.2001449.Engineering.

Morris, C. J. *et al.* (2013) 'Original article Day / Night Variability in Blood Pressure: Influence of Posture and Physical Activity', 26(June). doi: 10.1093/ajh/hpt026.

Mshali, H. *et al.* (2018) 'A survey on health monitoring systems for health smart homes', *International Journal of Industrial Ergonomics*. Elsevier B.V, 66, pp. 26–56. doi: 10.1016/j.ergon.2018.02.002.

Nehmer, J. *et al.* (2006) 'Living Assistance Systems - An Ambient Intelligence Approach -'.

Ni, Q., Hernando, A. B. G. and de la Cruz, I. P. (2015) The elderly independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development, *Sensors* (Switzerland). doi: 10.3390/s150511312.

Oh, H., Lee, K. and Seo, W. (2016) 'Temporal patterns of change in vital signs and Cardiac Arrest Risk Triage scores over the 48 hours preceding fatal in-hospital cardiac arrest', *Journal of Advanced Nursing*, 72(5), pp. 1122–1133. doi: 10.1111/jan.12897.

Padmanabhan, S. *et al.* (2018) 'Approach to record linkage of primary care data from Clinical Practice Research Datalink to other health-related patient data: overview and implications', *European Journal of Epidemiology*. Springer Netherlands, 4. doi: 10.1007/s10654-018-0442-4.

- Palaniappan, S. and Awang, R. (2008) 'Intelligent heart disease prediction system using data mining techniques', *2008 IEEE/ACS International Conference on Computer Systems and Applications*, pp. 108–115. doi: 10.1109/AICCSA.2008.4493524.
- Pani, A. K. and Bhattacharjee, G. P. (2001) 'Temporal representation and reasoning in artificial intelligence: A review', *Mathematical and Computer Modelling*, 34(1–2), pp. 55–80. doi: 10.1016/S0895-7177(01)00049-8.
- Patel, S. *et al.* (2012) 'A review of wearable sensors and systems with application in rehabilitation.', *Journal of neuroengineering and rehabilitation*. BioMed Central Ltd, 9(1), p. 21. doi: 10.1186/1743-0003-9-21.
- Pedersen, L. H., Henriksen, T. B. and Hammer, B. (no date) 'Selective serotonin reuptake inhibitors in pregnancy and congenital malformations: population based cohort study', pp. 1–6. doi: 10.1136/bmj.b3569.
- Pham, T. M. *et al.* (2018) 'Population-calibrated multiple imputation for a binary / categorical covariate in categorical regression models', (September), pp. 1–17. doi: 10.1002/sim.8004.
- Polat, K. and Güneş, S. (2006) 'The effect to diagnostic accuracy of decision tree classifier of fuzzy and k-NN based weighted pre-processing methods to diagnosis of erythematous-squamous diseases', *Digital Signal Processing: A Review Journal*, 16(6), pp. 922–930. doi: 10.1016/j.dsp.2006.04.007.
- Purdy, S. (2016) 'Encoding Data for HTM Systems', *Pacing and Clinical Electrophysiology*, 18(1), pp. 125–125. doi: 10.1111/j.1540-8159.1995.tb02489.x.
- Rafe, V. and Farhoud, R. H. (2013) 'A Survey on Data Mining Approaches in Medicine', 4(5), pp. 196–202.
- Raji, A., Jeyasheeli, P. G. and Jenitha, T. (2016) 'IoT based classification of vital signs data for chronic disease monitoring', *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–5. doi: 10.1109/ISCO.2016.7727048.

Rajkumar, A. and Reena, G. S. (2010) 'Diagnosis Of Heart Disease Using Datamining Algorithm', *Global Journal of Computer Science and Technology*, 10(10), pp. 38–43.

Ramos, C., Augusto, J. C. and Shapiro, D. (2008) 'Ambient Intelligence—the Next Step for Artificial Intelligence', *IEEE Intelligent Systems*, 23(2), pp. 15–18. doi: 10.1109/MIS.2008.19.

Rashidi, P. and Mihailidis, A. (2013) 'A survey on ambient-assisted living tools for older adults.', *IEEE journal of biomedical and health informatics*, 17(3), pp. 579–90. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24592460>.

Rech, J. and Althoff, K. (2004) 'Artificial Intelligence and Software Engineering : Status and Future Trends'.

Rich, E. and Knight, K. (1991) *Artificial Intelligence*. McGraw-Hill.

Rodriguez, M. A., Kotagiri, R. and Buyya, R. (2018) 'Detecting performance anomalies in scientific workflows using hierarchical temporal memory', *Future Generation Computer Systems*. Elsevier B.V., 88, pp. 624–635. doi: 10.1016/j.future.2018.05.014.

Salem, O. *et al.* (2014) 'Online Anomaly Detection in Wireless Body Area Networks for Reliable Healthcare Monitoring', *IEEE Journal of Biomedical and Health Informatics*, 18(5), pp. 1541–1551. doi: 10.1109/JBHI.2014.2312214.

Salih, A. S. M. and Abraham, A. (2013) 'A Review of Ambient Intelligence Assisted Healthcare Monitoring', *Mirlabs.Org*, 5, pp. 741–750. Available at: [http://www.mirlabs.org/ijcisim/regular\\_papers\\_2013/Paper153.pdf](http://www.mirlabs.org/ijcisim/regular_papers_2013/Paper153.pdf).

Sartakhti, J. S., Zangooui, M. H. and Mozafari, K. (2012) 'Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA)', *Computer Methods and Programs in Biomedicine*. Elsevier Ireland Ltd, 108(2), pp. 570–579. doi: 10.1016/j.cmpb.2011.08.003.

Satyanarayanan, M. (2001) 'Pervasive Computing : Vision and Challenges'.

Schmidt, A. (2005) 'Interactive Context-Aware Systems Interacting with Ambient Intelligence', *Ambient intelligence*, pp. 159–178. Available at: [http://www.neurovr.org/emerging/book5/09\\_AMI\\_Schmidt.pdf](http://www.neurovr.org/emerging/book5/09_AMI_Schmidt.pdf).

Skubic, M., Guevara, R. D. and Rantz, M. (2015) 'Automated health alerts using in-home sensor data for embedded health assessment', *IEEE Journal of Translational Engineering in Health and Medicine*, 3(March). doi: 10.1109/JTEHM.2015.2421499.

Spruston, N. (2008) 'Pyramidal neurons: Dendritic structure and synaptic integration', *Nature Reviews Neuroscience*, 9(3), pp. 206–221. doi: 10.1038/nrn2286.

Srinivas, K., Rani, B. and Govrdhan, A. (2010) 'Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks', *International Journal on Computer Science and Engineering*, 02(JANUARY 2010), pp. 250–255. doi: 10.1.1.163.4924.

Steventon, A., Bardsley, M. and Billings, J. (2012) 'Effect of telehealth on use of secondary care and mortality : findings from the Whole System Demonstrator cluster randomised trial', 3874(June), pp. 1–15. doi: 10.1136/bmj.e3874.

Swersky, L. *et al.* (2016) 'On the evaluation of outlier detection and one-class classification methods', *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 1–10. doi: 10.1109/DSAA.2016.8.

Tran, J. *et al.* (2018) 'Patterns and temporal trends of comorbidity among adult patients with incident cardiovascular disease in the UK between 2000 and 2014 : A population-based cohort study', pp. 1–23.

Varas, C. *et al.* (2018) 'Methodological challenges when evaluating potential off-label prescribing of drugs using electronic health care databases : A case study of

dabigatran etexilate in Europe', 2017(August 2017), pp. 1–11. doi: 10.1002/pds.4416.

Wang, Z., Yang, Z. and Dong, T. (2017) 'A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities and Monitor Vital Signs in Real Time', *Sensors*, 17(2), p. 341. doi: 10.3390/s17020341.

Welch, C., Bartlett, J. and Petersen, I. (2014) 'Europe PMC Funders Group Application of multiple imputation using the two-fold fully conditional specification algorithm in longitudinal clinical data', 14(2), pp. 418–431.

Wood, A. *et al.* (2006) 'ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring', *University of Virginia* .... Available at: <http://ursuletz.com/~control/docs/papers/tr06-alarmnet.pdf> (Accessed: 14 April 2014).

Wood, A. D. *et al.* (2008) 'Context-Aware Wireless Sensor Networks for Assisted Living and Residential Monitoring', (August), pp. 26–33.

Wu, J., Zeng, W. and Yan, F. (2016) 'Hierarchical Temporal Memory method for time-series-based anomaly detection', *IEEE 16th International Conference on Data Mining Workshops Hierarchical*, 273, pp. 535–546. doi: 10.1109/ICDMW.2016.64.

Zghebi, S. (2018) 'A longitudinal observational study on the patterns of comorbid clinical conditions in patients with and patients without Type 2 diabetes Clinical care and other categories posters : Type 2 diabetes', (April).

Zhang, S. *et al.* (2010) 'Development and evaluation of a compact motion sensor node for wearable computing', *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Ieee, pp. 465–470. doi: 10.1109/AIM.2010.5695916.

Zhao, C. *et al.* (2015) 'Advances in Patient Classification for Traditional Chinese Medicine : A Machine Learning Perspective'. Hindawi Publishing Corporation, 2015.

# APPENDICES

Appendix A.....	195
Appendix B.....	202
MIMIC-III, Description of service type .....	217

## Appendix A

### Terminology – Spatial Pooler and Temporal Memory

**Synapse:** in the SP algorithm, synapses on column's dendritic segment connect to bits in the input space, a synapse can have three different states including:

- Connected – permanence is above the threshold.
- Potential – permanence is below the threshold.
- Unconnected – does not have the ability to connect

**Permanence value:** indicates the amount of growth between a column in the SP and one bit in the input space.

**Permanence threshold:** the default connected threshold, any synapse that its permanence value is above the connected threshold is a “connected synapse”.

**Input vector, input space:** refer to binary bits in encoder SDRs.

**Encoder:** it converts the native format of data (i.e. datetime, number, category) into a binary SDR that can be fed into an HTM algorithms. The binary SDR is a vector of one and zero bits, for a given input value in such a way as to capture the important semantic characteristics of the data. Similar input values should produce highly overlapping SDRs.

connectedSynapses(c)

It is a subset of potentialSynapses(c) which the permanence value is equal or greater than “synPermConnected” value.

potentialSynapses(c)

It is a list of potential synapses and their permanence values for this column.

Boost(c)

The boost factors are used to increase the overlap of inactive columns to improve their chances of becoming active, and hence encourage participation of more columns in the learning process. Columns whose active duty cycle drops too much below that of their neighbours are boosted depending on how infrequently they have been active. Columns that has been active more than the target activation level have a boost factor below 1, meaning their overlap is suppressed

**global inhibition** - entails picking the top active columns with the highest overlap score in the entire region. At most half of the columns in a local neighbourhood are allowed to be active. Columns with an overlap score below the 'stimulusThreshold' are always inhibited.

**Local inhibition** - is performed on a column by column basis. Each column observes the overlaps of its neighbours and is selected if its overlap score is within the top active columns in its local neighbourhood. At most half of the columns in a local neighbourhood are allowed to be active. Columns with an overlap score below the 'stimulusThreshold' are always inhibited.

**Receptive field** - the input space that are visible to a column and a column can be potentially connect to. it is controlled with “potentialRadius” parameter.

**inhibitionradius** - The inhibition radius determines the size of a column's local neighbourhood. A cortical column must overcome the overlap score of columns in its neighbourhood in order to become active. This radius is updated every learning round. It grows and shrinks with the average number of connected synapses per column.

**activeDutyCycle** - a sliding average representing for a column to define how often the column has been active after inhibition.

**overlapDutyCycle** - a sliding average representing how often column *c* has had significant overlap with its inputs

## **Pseudocode**

### Spatial Pooler

The main task of the Spatial Pooler (SP) is to convert the region's input into a sparse pattern which is used in the HTM system to learn sequences and make predictions requires starting with sparse distributed patterns.

Initialization of parameters for Spatial Pooling algorithm is first phase in SP algorithm, prior to receiving any input, the SP initialization is computed by allocating a list of initial potential synapses for each column. The SP links each column to a random set of binary inputs from the input space (potential pool). Each input has synapse with an assigned random permanence value, this permanence value is selected to be in small range around permanence threshold, this enables potential synapses to become in "connected" or "disconnected" state after a small number of training iteration.

After initializing the Spatial Pooler, three phases will be carried out to achieve the main task of the spatial pooler including:

- Compute the overlap with the current input for each column
- Inhibition
- Learning

---

**Phase 1: Compute the overlap with the current input for each column.**

---

**for *c* in columns then**

---

```
overlap(c) = 0
for s in connectedSynapses(c) then
    overlap(c) = overlap(c) + input (t, s.sourceInput)
end for
if overlap(c) < minOverlap then
    overlap(c) = 0
else
    overlap(c) = overlap(c) * boost(c)
end if
end for
```

---

---

**Phase 2: Calculates columns remain as winners after the inhibition.**

---

```
for c in columns then
    minLocalActivity = kthScore(neighbors(c), desiredLocalActivity)

    if overlap(c) > 0 and overlap(c) ≥ minLocalActivity then
        activeColumns(t).append(c)
    end if
end for
```

---

---

**Phase 3: Updates the permanence values of winning column's synapses as well as their boost values and inhibition radius if it is necessary.**

---

```
for c in activeColumns(t) do
    for s in potentialSynapses(c) do
        if active(s) then
            s.permanence += permanenceInc
            s.permanence = min(1.0, s.permanence)
        else
            s.permanence -= permanenceDec
            s.permanence = max(0.0, s.permanence)
        end if
    end for
end for
```

---

```
for c in columns do
  minDutyCycle(c) = 0.01 * maxDutyCycle(neighbors(c))
  activeDutyCycle(c) = updateActiveDutyCycle(c)
  boost(c) = boostFunction(activeDutyCycle(c), minDutyCycle(c))

  overlapDutyCycle(c) = updateOverlapDutyCycle(c)
  if overlapDutyCycle(c) < minDutyCycle(c) then
    increasePermanences(c, 0.1*connectedPerm)
  end if
  inhibitionRadius = averageReceptiveFieldSize()
end for
```

---

### Temporal Memory

The pseudocode for the Temporal Memory is divided into three sequential phases. This version of pseudocode implements inference and learning operations. Phase 3 is only for learning, however phase 1 and phase 2 contain some learning-specific operations when learning is activated (Hawkins, Ahmad and Dubinsky, 2011).

---

**Phase 1: calculates the activeState for each cell that is in a winning column.**

---

```
for c in activeColumns(t) do
  buPredicted = false
  IcChosen = false
  for i = 0 to cellsPerColumn - 1 do
    if predictiveState(c, i, t-1) == true then
      s = getActiveSegment(c,i,t-1, activeState)
      if s.sequenceSegment == true then
        buPredicted = true
        activeState(c,i,t) = 1
      if segmentActive(s,t-1,learnState) then
```

---

```
        IcChosen = true
        learnState(c, i, t) = 1
    end if
end if
end if

if buPredicted == false then
    for i = 0 to cellsPerColumn - 1 do
        activeState(c,i, t) = 1
    end for
end if
if IcChosen == false then
    I,s = getBestMatchingCell(c, t-1)
    learnState(c,i, t) = 1
    sUpdate = getSegmentActiveSynapses (c, i, s, t-1, true)
    sUpdate.sequenceSegment = true
    segmentUpdateList.add(sUpdate)
end if
end for
end for
```

---

---

**Phase 2: calculates the predictive state for each cell**

---

```
for c, i in cells do
    for s in segments(c,i) do
        if segmentActive(s, t, activeState) then
            predictiveState(c,i,t) = 1

            activeUpdate = getSegmentActiveSynapses (c, i, s, t, false)
            segmentUpdateList.add(activeUpdate)
            predSegment = getBestMatchingSegment (c, i, t-1)
            preUpdate = getSegmentActiveSynapses (c, i, predSegment, t-1, True)
            segmentUpdateList.add(predUpdate)
        end if
    end for
end for
```

---

Phase 3: carries out learning. Update the internal variables.

---

**for** c, i **in** cells **do**

**if** learnState(s, i, t) == 1 **then**

        adaptSegments (segmentUpdateList (c, i), true)

        segmentUpdateList(c, i).delete()

**else if** predictiveState(c, i, t) == 0 **and** predictiveState(c,i, t-1) == 1 **then**

        adaptSegments (segmentUpdateList(c,i), false)

        segmentUpdateList(c, i).delete()

**end if**

**end for**

---

## Appendix B

### Proposed Framework

In this appendix the pseudocode and algorithms code of the proposed framework will be documented.

#### **1. Pseudocode**

The proposed framework is inspired by the HTM networks. The proposed framework is constructed as a two-level hierarchy, and each level in the hierarchy consists of several regions. The two-level hierarchy is used as information ascends in the hierarchy, the output of regions in lower levels will be converged, and the regions on the higher level of the hierarchy make use of the data patterns to produce a bigger picture of the patient's health condition.

1.1. Level 1

---

**Pseudocode** - Level 1 of proposed framework.

---

```
for each region do           //try this for vital signs: SBP, DBP, HR and SpO2
  for file in Input_Learning_set do
    for i=1 to learningNumber for learning region do
      region.learningmode = 1
      region.inferencemode = 0
      timeOfDay = READ timeOfDay
      vitalsign = READ vitalsign
      sdr_date = ENCODE (timeOfDay)
      sdr_vitalsign = ENCODE (vitalsign)
      sdr_data = CONCATENATE (sdr_date, sdr_vitalsign)
      sdr_spatial_pooler = COMPUTE SpatialPooler(sdr_data)
      sdr_temporal_memory = COMPUTE TemporalMemory(sdr_spatialpooler)
      anomaly_score = COMPUTE anomaly(sdr_temporal_memory)
      PRINT anomalyScore, vitalsign, timeOfDay
    end for
  end for
for file in Input_Inference_set do
  for i=1 to inferenceNumber for inference region do
    region.learningmode = 0
    region.inferencemode = 1
    timeOfDay = READ timeOfDay
    vitalsign = READ vitalsign
    sdr_date = ENCODE (timeOfDay)
    sdr_vitalsign = ENCODE (vitalsign)
    sdr_data = CONCATENATE (sdr_date, sdr_vitalsign)
    sdr_spatial_pooler = COMPUTE SpatialPooler(sdr_data)
    sdr_temporal_memory = COMPUTE TemporalMemory(sdr_spatialpooler)
    anomaly_score = COMPUTE anomaly(sdr_temporal_memory)
    PRINT anomalyScore, vitalsign, timeOfDay
  end for
end for
```

---

**end for**

---

1.2. Level 2

---

**Pseudocode - Level 2 of proposed framework.**

---

```
for each region do           //try this for vital signs: SBP, DBP, HR and SpO2
  for file in Input_Learning_set do
    for i=1 to learningNumber for learning region do
      region.learningmode = 1
      region.inferencemode = 0
      timeOfDay = READ timeOfDay
      HR_AnomalyScore = READ HR_AnomalyScore
      SBP_AnomalyScore = READ SBP_AnomalyScore
      DBP_AnomalyScore = READ DBP_AnomalyScore
      SpO2_AnomalyScore = READ SpO2_AnomalyScore
      sdr_date = ENCODE (timeOfDay)
      sdr_HR_AnomalyScore = ENCODE (HR_AnomalyScore)
      sdr_SBP_AnomalyScore = ENCODE (SBP_AnomalyScore)
      sdr_DBP_AnomalyScore = ENCODE (DBP_AnomalyScore)
      sdr_SpO2_AnomalyScore = ENCODE (SpO2_AnomalyScore)
      sdr_data = CONCATENATE (sdr_date,sdr_HR_AnomalyScore,sdr_
          SBP_AnomalyScore, sdr_DBP_AnomalyScore,sdr_SpO2_AnomalyScore)

      sdr_spatial_pooler = COMPUTE SpatialPooler(sdr_data)
      sdr_temporal_memory = COMPUTE TemporalMemory (sdr_spatialpooler)
      anomaly_score = COMPUTE anomaly(sdr_temporal_memory)
      PRINT anomalyScore,HR_AnomalyScore,SBP_
          AnomalyScore,DBP_AnomalyScore, SpO2_AnomalyScore,timeOfDay
    end for
  end for
for file in Input_Inference_set do
  for i=1 to inferenceNumber for inference region do
    region.learningmode = 0
    region.inferencemode = 1
```

---

```
timeOfDay = READ timeOfDay
HR_AnomalyScore = READ HR_AnomalyScore
SBP_AnomalyScore = READ SBP_AnomalyScore
DBP_AnomalyScore = READ DBP_AnomalyScore
SpO2_AnomalyScore = READ SpO2_AnomalyScore
sdr_date = ENCODE (timeOfDay)
sdr_HR_AnomalyScore = ENCODE (HR_AnomalyScore)
sdr_SBP_AnomalyScore = ENCODE (SBP_AnomalyScore)
sdr_DBP_AnomalyScore = ENCODE (DBP_AnomalyScore)
sdr_SpO2_AnomalyScore = ENCODE (SpO2_AnomalyScore)
sdr_data = CONCATENATE (sdr_date,sdr_HR_AnomalyScore,sdr_
    SBP_AnomalyScore, sdr_DBP_AnomalyScore,sdr_SpO2_AnomalyScore)

sdr_spatial_pooler = COMPUTE SpatialPooler(sdr_data)
sdr_temporal_memory = COMPUTE TemporalMemory (sdr_spatialpooler)
anomaly_score = COMPUTE anomaly(sdr_temporal_memory)
PRINT anomalyScore,HR_AnomalyScore,SBP_
    AnomalyScore,DBP_AnomalyScore, SpO2_AnomalyScore,timeOfDay
end for
end for
end for
```

---

## 2. Software code

For the implementation of the proposed framework/model, the NuPIC version 1.0.5 platform with NuPIC Bindings version 1.0.0 and Python version 2.7.3 are used.

In the proposed framework, each region is configured to firstly learn the complex patterns from the learning set. The regions begin the inference process once the learning phase is accomplished. Below is the proposed framework is constructed as a two-level hierarchy.

## 2.1. Level 1

Below is a copy of algorithms code that is implemented for four regions in level 1. The output from level-one consists of anomaly scores that are computed for each vital sign. The anomaly score from level-one will be feed-forwarded into level-two for further processing.

A similar programming model is used for each region. Code below can be amended for each data stream, in this case, HR, SBP, DBP and SpO2 values.

```
import shutil

# The anomaly calculation modules
from nupic.algorithms.anomaly import Anomaly
from nupic.algorithms import anomaly_likelihood as AL

# These are the python implementation of the spatial pooler and the
temporal memory

# These are the same but the faster, C++ implementation
from nupic.algorithms.spatial_pooler import SpatialPooler as SP
from nupic.algorithms.temporal_memory import TemporalMemory as TM

# The encoders
# from nupic.encoders import ScalarEncoder
#from nupic.encoders import SDRCategoryEncoder
from nupic.encoders.date import DateEncoder
from nupic.encoders import random_distributed_scalar

import numpy as np

# The standard python datetime module
import datetime
import csv
import glob
import os

# Helper function to find the index of an active bit
def find_idx(s):
    return [i for i, x in enumerate(s) if int(x) == 1]

# Helper function to get an SDR out of the temporal memory
def get_sdr(tm_cellsPerCol, cells):
    return set([x/tm_cellsPerCol for x in cells])

## initializing the encoder- it is putted in here because, if I
putted in loop then it does not produce a correct SDR
```

```

e_rds =
random_distributed_scalar.RandomDistributedScalarEncoder(resolution=
1,w=21,n= 220)

# This dictionary holds the model parameters including the spatial
pooler, temporal memory and the encoders parameters.
settings = {

    'sp': SP(inputDimensions=(224,),
             columnDimensions=(600,),
             synPermConnected=0.1,
             numActiveColumnsPerInhArea=12.0,
             boostStrength = 1.0,
             synPermActiveInc=0.01,#0.001
             synPermInactiveDec=0.005,
             globalInhibition=True,
             potentialPct=0.85,
             seed=1956,
             spVerbosity=0,
             minPctOverlapDutyCycle=0.001,
             dutyCyclePeriod=1000,
             localAreaDensity=-1.0,
             potentialRadius=16,
             stimulusThreshold=0,
             wrapAround= False),

    'tm': TM(columnDimensions = (600,),
             cellsPerColumn= 32,
             initialPermanence=0.21,
             connectedPermanence=0.5,
             maxNewSynapseCount=20,
             permananceIncrement=0.1,
             permananceDecrement=0.005,
             minThreshold=12,
             activationThreshold=16,
             maxSegmentsPerCell=200,
             maxSynapsesPerSegment=26,
             predictedSegmentDecrement=0.0,
             seed=1960),

    # predictedSegmentDecrement`: A good value is just a bit larger
    than(the column-level sparsity * permananceIncrement). So, if
    column-level sparsity is 2% and permananceIncrement is 0.01, this
    parameter should be
    # something like 4% * 0.01 = 0.0004).
    'w': 3,
    'n': 600,
    'timeOfDay': (1, 6)
}

# This class is our HTM/CLA model consisting of several encoders for
each column, a spatial pooler and a temporal memory
class HTMModel(object):

    # initializing
    prevPredictedColumns = np.array([])
    next_t_columns = []
    anomaly = Anomaly(slidingWindowSize=None, mode='pure',
binaryAnomalyThreshold=None)

```

```

# This class constructor takes the dataset length (datasetLen)
so it can calculate the moving average needed for the anomaly score
calculation
def __init__(self, datasetLen, filename):

    self.patient_condition = filename.split("_")[6]
    self.patient_age = filename.split("_")[3]
    self.patient_gender = filename.split("_")[2]
    self.patient_status = filename.split("_")[4]

    # initializing these variables based on the parameter
dictionary
    self.n = settings['n']
    self.activeColumns = []
    self.sp = settings['sp']
    self.tm = settings['tm']
    self.date_sdr = []
    self.sensor_sdr = []

    datasetLen = datasetLen
    estimationSamples = int(datasetLen * 0.1)
    # for learning phase we increase this value
    #int(datasetLen * 0.1) - estimationSamples
    #learningPeriod = estimationSamples
    learningPeriod = int(datasetLen * 0.2)
    historicWindowSize = int(datasetLen * 0.2)

    self.al = AL.AnomalyLikelihood(
        claLearningPeriod = None,
        learningPeriod = learningPeriod, # If you get an error
                                         about this, try to
                                         change it to
                                         (LearningPeriod =
                                         learningPeriod)

        estimationSamples = estimationSamples,
        historicWindowSize = historicWindowSize,
        reestimationPeriod = estimationSamples

    )

# This method takes a row of the dataset and the row number (i)
def compute(self, i, row, learn, infer=True):

    # Selecting some data out of the dataset
    p_service = row[17]
    c_service = row[18]
    sensors = row[4]
    date = row[0]
    # Creating the date SDR
    de = DateEncoder(timeOfDay=settings['timeOfDay'])
    now = datetime.datetime.strptime(date, "%Y-%m-%d %H:%M:%S")
    sdr_date = de.encode(now)
    self.date_sdr = sdr_date

    sdr_data = e_rds.encode(float(sensors))
    print self.date_sdr, date, sensors
    # Then, we concatenate the previous big SDR with and date sdr.
    sdr = np.concatenate((sdr_date, sdr_data))
    print sdr.astype('int16')

```

```

        # Creating an empty active array to store the output of the
        spatial pooler
        activeArray = np.zeros(self.n, dtype="uint32")

        # LEARNING and TESTING - Feeding the SDR to the spatial
        pooler and the boolean flag indicates that we want the spatial
        pooler to learn from this. The output of this is stored in the
        activeArray
        self.sp.compute(sdr, learn, activeArray)

        #http://nupic.docs.numenta.org/stable/quick-
        start/algorithms.html
        #activeColumnIndices = np.nonzero(activeArray)[0]
        #print activeColumnIndices
        # The activeArray is a binary vector, so to get the actual
        indices of the active bits, we use this helper function
        self.activeColumns = set(find_idx(activeArray))

        # Then we feed that to the temporal memory. The temporal
        memory will not output anything, but we can get the active
        cells and the predictive cells later on from this
        `self.tm` object
        self.tm.compute(self.activeColumns, learn)

        # This calculates the raw anomaly score
        anomalyScore =
self.anomaly.compute(list(self.activeColumns),
list(self.prevPredictedColumns))

        # getting the predictive cells and converting them to an SDR
        and storing it in the variable `self.prevPredictedColumns`
        predictedColumns = get_sdr(self.tm.getCellsPerColumn(),
self.tm.getPredictiveCells())
        self.prevPredictedColumns = predictedColumns

        #print predictedColumns
        # Calculating the likelihood probability of the anomaly score
        likeScore = self.al.anomalyProbability(sdr, anomalyScore,
date)

        # Calculating the log likelihood probability of the anomaly
        score
        logScore = self.al.computeLogLikelihood(likeScore)

        # We have 3 anomaly metrics we can use and experiment with:
        the raw anomaly score, and the likelihood probability and
        the log of that.
        # From my experience the loglikelihood is the best one.
        However, this might be dependant on the dataset.
        # Finally we return these scores with the actual label.
        return (date, round(float(anomalyScore), 2),
round(float(likeScore), 2), round(float(logScore), 2),
sensors, p_service, c_service)

def main():

###make sure you decrease the learning period value

```

```

datasetfile_learning_1 =
glob.glob(r'C:\Framework_V11\Framework\Processed_Data_SOURCE\PROCESS
ED_DATA_FOR LEARNING\*.csv')
datasetfile_testing =
glob.glob(r'C:\Framework_V11\Framework\Processed_Data_SOURCE\PROCESS
ED_DATA_FOR TESTING\*.csv')
##Learning Phase
for file in datasetfile_learning_1:
    head, tail = os.path.split(file)
    print tail
    with open(file) as f:
        datasetLen = len(f.readlines()) - 1

        reader = csv.reader(open(file, 'r'))
        next(reader) # skipping the header row

        # Here we create an instance of our HTM model
        model = HTMModel(datasetLen,filename=tail)
        filename_to_write = tail.replace('_metavision.csv', '')
        writer =
csv.writer(open("C:\Framework_V11\Framework\step_one\HR_CLASS\output
_learning\\"+filename_to_write+"_HRresult.csv", 'wb'))
        writer.writerow( ('timestamp', 'anomalyScore',
'anomalyLikelihood', 'logLikelihood', 'HR','p_service','c_service')
)
        # reading the dataset row by row and feeding the rows to the
model
        for i, row in enumerate(reader, start=1):
            # feeding the model the row number and the row content

            result = model.compute(i, row,True)
            # writing the results
            writer.writerow(result)
##Testing Phase
for file in datasetfile_testing:
    head, tail = os.path.split(file)
    with open(file) as f:
        datasetLen = len(f.readlines()) - 1

        reader = csv.reader(open(file, 'r'))
        next(reader) # skipping the header row

        # Here we create an instance of our HTM model
        model = HTMModel(datasetLen,filename=tail)
        filename_to_write = tail.replace('_metavision.csv', '')
        writer =
csv.writer(open("C:\Framework_V11\Framework\step_one\HR_CLASS\output
_testing\\"+filename_to_write+"_HRresult.csv", 'wb'))
        writer.writerow( ('timestamp', 'anomalyScore',
'anomalyLikelihood', 'logLikelihood', 'HR','p_service','c_service')
)
        # reading the dataset row by row and feeding the rows to the
model
        for i, row in enumerate(reader, start=1):
            # feeding the model the row number and the row content

            result = model.compute(i, row,False)
            # writing the results

```

```

        writer.writerow(result)

if __name__ == '__main__':
    main()

```

## 2.2. Level 2

Level-two is responsible for extracting the correlation patterns between anomaly scores computed for each vital sign in level-one. The code below implements one region in level-two where its output will be a final anomaly score which is used as a predictor to detect abnormal behaviour of vital signs prior to cardiac arrest.

```

import cPickle as pickle
import TextReader
import shutil
from nupic.serializable import Serializable

# The anomaly calculation modules
from nupic.algorithms.anomaly import Anomaly
from nupic.algorithms import anomaly_likelihood as AL

# These are the python implementation of the spatial pooler and the
temporal memory
# These are the same but the faster, C++ implementation
from nupic.algorithms.spatial_pooler import SpatialPooler as SP
from nupic.algorithms.temporal_memory import TemporalMemory as TM

# The encoders
# from nupic.encoders import ScalarEncoder
from nupic.encoders import SDRCategoryEncoder
from nupic.encoders.date import DateEncoder
from nupic.encoders import random_distributed_scalar

# Can't live without Numpy
import numpy as np

# The standard python datetime module
import datetime

# The standard python csv module
import csv

import glob
import os
# Helper function to find the index of an active bit
def find_idx(s):
    return [i for i, x in enumerate(s) if int(x) == 1]

# Helper function to get an SDR out of the temporal memory
def get_sdr(tm_cellsPerCol, cells):
    return set([x/tm_cellsPerCol for x in cells])

```

```

encoder_rds =
random_distributed_scalar.RandomDistributedScalarEncoder(n= 220,
w=21,resolution=0.01)

# This dictionary holds the model parameters including the spatial
pooler, temporal memory and the encoders parameters.
settings = {
    'sp': SP(inputDimensions=(884,),
            columnDimensions=(2048,),
            synPermConnected=0.1,
            numActiveColumnsPerInhArea=20.0, ### framework_v11 - 2%
of 2048
            boostStrength = 3.0,
            synPermActiveInc=0.01,
            synPermInactiveDec=0.0005,
            globalInhibition=True,
            potentialPct=0.85,
            seed=1956,
            spVerbosity=0,
            minPctOverlapDutyCycle=0.001,
            dutyCyclePeriod=1000,
            localAreaDensity=-1.0,
            potentialRadius=16,
            stimulusThreshold=0,
            wrapAround= False),
    'tm': TM(columnDimensions = (2048,),
            cellsPerColumn= 32,
            initialPermanence=0.21,
            connectedPermanence=0.5,
            maxNewSynapseCount=20,
            permananceIncrement=0.1,
            permananceDecrement=0.1,
            minThreshold=12,
            activationThreshold=16,
            maxSegmentsPerCell=128,
            maxSynapsesPerSegment=28,
            predictedSegmentDecrement=0.0,
            seed=1960),
    #'w': 3,
    'n': 2048, ### framework_v8_v1 - this value is changed from 600
to 400 to match columnDimensions
    'encoder_n': 6,
    'timeOfDay': (1, 6)
}

# This class is our HTM/CLA model consisting of several encoders for
each column, a spatial pooler and a temporal memory
class HTMModel_score(object):

    # initalizing
    prevPredictedColumns = np.array([])
    anomaly = Anomaly(slidingWindowSize=None, mode='pure',
binaryAnomalyThreshold=None)

    # This class constructor takes the dataset lenght (datasetLen)
so it can calculate the moving average needed for the anomaly score
calculation
    def __init__(self, datasetLen,filename):

```

```

        # initializing these variables based on the parameter
        dictionary
        self.n = settings['n']
        self.activeColumns = []
        self.sp = settings['sp']
        self.tm = settings['tm']
        self.date_sdr = []

        datasetLen = datasetLen
        estimationSamples = int(datasetLen * 0.1)
        learningPeriod = int(datasetLen * 0.2)
        historicWindowSize = int(datasetLen * 0.2)

        ##must read >>>
https://github.com/numenta/nupic/blob/50c5fd0dc94f2fffb205544ed11fe82ad5bb0de18/src/nupic/algorithms/anomaly\_likelihood.py#L154-L155
        self.al = AL.AnomalyLikelihood(
            claLearningPeriod = None,
            learningPeriod = learningPeriod, # If you get an error
            about this, try to change it to (LearningPeriod = learningPeriod),
            they changed the name of this argument in later versions of NuPIC
            estimationSamples = estimationSamples,
            historicWindowSize = historicWindowSize,
            reestimationPeriod = estimationSamples
        )

        # This method takes a row of the dataset and the row number (i)
        def compute(self, i, row, learn, infer=True):
            hr = row[4]
            bps = row[9]
            bpd = row[14]
            spo = row[19]
            sensor_hr_score = row[1]
            sensor_hr_log = row[3]
            sensor_bps_score = row[6]
            sensor_bps_log = row[8]
            sensor_bpd_score = row[11]
            sensor_bpd_log = row[13]
            sensor_spo_score = row[16]
            sensor_spo_log = row[18]
            p_service = row[20]
            c_service = row[21]

            date = row[0]
        # Creating the date SDR
            de = DateEncoder(timeOfDay=settings['timeOfDay'])
            now = datetime.datetime.strptime(date, "%Y-%m-%d %H:%M:%S")
            sdr_date = de.encode(now)
            self.date_sdr = sdr_date

            rds_hr_score = encoder_rds.encode(float(sensor_hr_score))
            rds_bps_score = encoder_rds.encode(float(sensor_bps_score))
            rds_bpd_score = encoder_rds.encode(float(sensor_bpd_score))
            rds_spo_score = encoder_rds.encode(float(sensor_spo_score))

        ## # Then, we concatenate the previous big SDR with and date
        sdr.

```

```

        sdr =
np.concatenate((sdr_date,rds_hr_score,rds_bps_score,rds_bpd_score,rds_spo_score))
        ##sdr =
np.concatenate((sdr_date,rds_hr_log,rds_bps_log,rds_bpd_log,rds_spo_log))

        # Creating an empty active array to store the output of the
spatial pooler
        activeArray = np.zeros(self.n, dtype="uint32")
        ##print activeArray

        # Feeding the SDR to the spatial pooler and the boolean flag
indicates that we want the spatial pooler to learn from this. The
output of this is stored in the activeArray
        self.sp.compute(sdr, learn, activeArray)
        ##print activeArray
        # The activeArray is a binary vector, so to get the actual
indices of the active bits, we use this helper function
        self.activeColumns = set(find_idxs(activeArray))
        ##print self.activeColumns
        # Then we feed that to the temporal memory. The temporal
memory will not output anything, be we can get the active cells and
the predictive cells later on from this `self.tm` object
        self.tm.compute(self.activeColumns, learn)

        # This calculates the raw anomaly score
        anomalyScore =
self.anomaly.compute(list(self.activeColumns),
list(self.prevPredictedColumns))

        # getting the predictive cells and converting them to an SDR
and storing it in the variable `self.prevPredictedColumns`
        predictedColumns = get_sdr(self.tm.getCellsPerColumn(),
self.tm.getPredictiveCells())
        self.prevPredictedColumns = predictedColumns

        # Calculating the likelihood probability of the anomaly score
likeScore = self.al.anomalyProbability(sdr, anomalyScore,
date)

        # Calculating the log likelihood probability of the anomaly
score
        logScore = self.al.computeLogLikelihood(likeScore)

        # We have 3 anomaly metrics we can use and experiment with:
the raw anomaly score, and the likelihood probability and the log of
that.
        # From my experience the loglikelihood is the best one.
However, this might be dependant on the dataset.
        # Finally we return these scores with the actual label.
        return (date, round(float(anomalyScore),2),
round(float(likeScore),2),
round(float(logScore),2),p_service,c_service,hr,bps,bpd,spo)

def main():

```

```

    datasetfile =
glob.glob(r'C:\Framework_V11\Framework\step_two\scores_combined_step
_three\*.csv')

##Learning
    for file in datasetfile:
        head, tail = os.path.split(file)
        print file
        with open(file) as f:
            datasetLen = len(f.readlines()) - 1

            reader = csv.reader(open(file, 'r'))
            next(reader) # skipping the header row

            # Here we create an instance of our HTM model
            model = HTMModel_score(datasetLen,filename=tail)
            filename_to_write = tail.replace('.csv', '')
            #print filename_to_write
            writer =
csv.writer(open("C:\Framework_V11\Framework\step_two\scores_combined
_step_five_anomalyscore_only\\"+filename_to_write+".csv", 'wb'))
            writer.writerow( ('timestamp', 'anomalyScore',
'anomalyLikelihood', 'logLikelihood', 'p_service', 'c_service',
'hr', 'bps', 'bpd', 'spo2') )
            # reading the dataset row by row and feeding the rows to the
model

            for i, row in enumerate(reader, start=1):
                # feeding the model the row number and the row content
                result = model.compute(i, row, True)
                # writing the results
                writer.writerow(result)

##Testing

    datasetfile_testing =
glob.glob(r'C:\Framework_V11\Framework\step_two\scores_combined_step
_three_testing\*.csv')
    for file in datasetfile_testing:
        head, tail = os.path.split(file)
        with open(file) as f:
            datasetLen = len(f.readlines()) - 1

            reader = csv.reader(open(file, 'r'))
            next(reader) # skipping the header row

            # Here we create an instance of our HTM model
            model = HTMModel_score(datasetLen,filename=tail)
            filename_to_write = tail.replace('.csv', '')
            #print filename_to_write
            writer =
csv.writer(open("C:\Framework_V11\Framework\step_two\scores_combined
_step_five_anomalyscore_only_testing\\"+filename_to_write+".csv",
'wb'))

```

```
        writer.writerow( ('timestamp', 'anomalyScore',
'anomalyLikelihood', 'logLikelihood', 'p_service', 'c_service',
'hr', 'bps', 'bpd', 'spo2') )
        # reading the dataset row by row and feeding the rows to the
model

    for i, row in enumerate(reader, start=1):
        # feeding the model the row number and the row content
        #print row[0]
        result = model.compute(i, row, False)
        # writing the results
        writer.writerow(result)

if __name__ == '__main__':
    main()
```

## MIMIC-III, Description of service type

Description of service type that are stored in SERVICES table. Each service is listed in the table as an abbreviation with a description for each one of them.

Table 9-1 Description of each service type (A. E. W. Johnson *et al.*, 2016)

<b>Service</b>	<b>Description</b>
CMED	Cardiac Medical - for non-surgical cardiac related admissions
CSURG	Cardiac Surgery - for surgical cardiac admissions
DENT	Dental - for dental/jaw related admissions
ENT	Ear, nose, and throat - conditions primarily affecting these areas
GU	Genitourinary - reproductive organs/urinary system
GYN	Gynecological - female reproductive systems and breasts
MED	Medical - general service for internal medicine
NB	Newborn - infants born at the hospital
NBB	Newborn baby - infants born at the hospital
NMED	Neurologic Medical - non-surgical, relating to the brain
NSURG	Neurologic Surgical - surgical, relating to the brain
OBS	Obstetrics - concerned with childbirth and the care of women giving birth
ORTHO	Orthopaedic - surgical, relating to the musculoskeletal system
OMED	Orthopaedic medicine - non-surgical, relating to musculoskeletal system
PSURG	Plastic - restoration/reconstruction of the human body (including cosmetic or aesthetic)

PSYCH	Psychiatric - mental disorders relating to mood, behaviour, cognition, or perceptions
SURG	Surgical - general surgical service not classified elsewhere
TRAUM	Trauma - injury or damage caused by physical harm from an external source
TSURG	Thoracic Surgical - surgery on the thorax, located between the neck and the abdomen
VSURG	Vascular Surgical - surgery relating to the circulatory system