A Cognitive IoE (Internet of Everything) Approach to Ambient-Intelligent Smart Space

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Gopal Singh Jamnal

Edinburgh Napier UNIVERSITY

SCHOOL OF COMPUTING

Jan 2019

DECLARATION

I hereby declare that the thesis entitled “A Cognitive IoE (Internet of Everything) approach to Ambient Intelligent Smart Space” submitted for the award of *Doctor of Philosophy* by Edinburgh Napier University is a record of bonafide work carried out by me under the supervision of Prof. Xiaodong Liu, CAVES (Center for algorithms,visualization and evolving systems) and Dr. Augusto Abreu Esteves, Center for Interaction Design, School of Computing, Edinburgh Napier University, UK.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Edinburgh

Date: **Signature of the Candidate**

ABSTRACT

At present, the United Nations figures claim that the current world population would rise from 7.6 billion to 8.5 billion in 2030 and 9.7 billion in 2050. Therefore by the 2050, 65 percent of world's population would be living in urban mega-cities and each megacity would be accommodating around 10 million inhabitants. Such massive urbanization of growing population would be known as 21st century's 'Urban Age'. On the other side, by 2020 the growing population of elderly people above 65 years old would be increasing by 25 percent in EU countries and by 30 percent in other developing nations including Asia and North America. As a result, the growth of massive population and elderly inhabitants in urban cities would require an assisted living environment for independent and comfortable living experiences. As can be expected, a persuasive demand of assisted living environment would be vital to the humankind. The goal of an assisted living environment is to support the aging population and inhabitants to live independently in their own home and communities with the support of trained services and personal digital assistants. Therefore, the continuous growing demand of assisted living environment targets to improve the inhabitants comfort level and efficiency to do their ADL (Activity Daily Living) routine tasks by enabling the cooperation among various IoT smart objects and sensors which will understand the environmental surroundings and the inhabitant's contextual needs in a proactive manner.

In this work, a Cognitive IoE (Internet of Everything) framework with ambient intelligence capability is proposed to observe the inhabitant activities with heterogeneous IoT network objects and sensors in a time series manner to perceive the inhabitant intentions and situations in the environment. The predictive regression model forecasts the inhabitant's activity patterns with regressive machine learning algorithms. The interconnected network objects (sensors and actuators) behave as agents to learn, think and adapt to contextual situations in the dynamic environment with no or minimum human intervention. Therefore, the first research challenge is to recognize the inhabitant's intentional-situation in the environment, and it is achieved by the Ambient Cognition Model(ACM). The ACM not only performs IoT data-fusion but also applies a statistical model for threshold and weight scheme to extract contextual information in a more systematic manner. The second research challenge of automating the predictive regression model to forecast the time series activity patterns of inhabitants is addressed within the Ambient-Expert Model(AEM). The hidden activity state patterns are identified, trained and tested with the supervised machine learning method of Hidden Markov Model, Recurrent-Neural Network, and Naive Bayes classifier. In addition, a recursive training mechanism of DATAWELL is integrated with the architecture to train(re-train) the model over new datasets and perform predictive analysis in a proactive manner.

Furthermore, the unified framework CAiSH (Cognitive Ambient Intelligent Smart Home), built upon the integration of ACM and AEM architectures to a provide an intelligent IoT framework for the ambient intelligence smart home environment. The trained model uses maximum likelihood posterior probabilities to forecast the inhabitant's intentional activity states. The CAiSH works as a proactive digital assistant to the inhabitant provide a development platform for autonomous and enhanced assisted living services in the cognitive IoE environment. The research has been carried out on time-series data sets, deploying IoT lab to generate and collect time series data for the training and testing purpose and providing hands-on research experience on IoT prototype deployment. Overall, 5499 datasets of 30 SA (Spot-Activities) and 9 IA (Intention- Activities) data sets have been engaged for the training and evaluation. The result outputs are evaluated with MAE (Mean-Square Error), MAPE (Mean Absolute Percentage Error) and MAE (Mean Absolute Error) metrics for the prediction accuracy measures.

Keywords: *Time Series Forecasting*, *Data Science for IoT systems*, *Cognitive Ambient Intelligent Smart Home*, *Activity Pattern Recognition*, *Cognitive IoTs*, *Ambeint Cognition Model*, *Real-time IoT system*, *Ambient Expert Model*

ACKNOWLEDGEMENT

With the deep sense of gratitude, I would like to thank my supervisors Prof. Xiaodong Liu and Dr. Augusto Esteves and PhD panel chair Prof. Ahmed Al-Dubai for all their help, support, expertise and understanding throughout my period of PhD study. I would also like to thank all staff in the School of Computing at Edinburgh Napier University, especially the members of the CAVES group, for providing me the valuable feedback and suggestions during my PhD study.

I am most indebted to my MOTHER and FATHER, whom I lost during my PhD journey. Their unconditional love, endless spiritual support, motivations and vision about life will always be remembered in my deep heart throughout my life to follow their life teachings. My family did hold me tightly during this emotionally difficult time and encouraged me to complete my journey with good faith.

Place: Edinburgh

Date: 12/01/2019 **Gopal Singh Jamnal**

LIST OF PUBLICATIONS

Conference Papers

1. Jamnal G and Liu X, (2017) A Cognitive-IoE Approach to Ambient-intelligent Smart Home, Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security, 2017.

2. Jamnal G, Liu X and Lu F, (2017) Home Automation : HMM-based fuzzy rule engine for Ambient Intelligent Smart Space, Proceedings of the 29th International Conference on Software Engineering and Knowledge Engineering , 2017.

Journal Paper

3. Jamnal G, Liu X, Agusto E, (2018) A Cognitive Ambient-Intelligent Smart Home, IEEE Transactions on Consumer Electronics (submitted).

Book Chapters

4. Jamnal G, Liu X and Ramachandran M, (2017) Cognitive Internet of Everything (CIoE), Research Handbook of Emerging Trends and Applications of the Internet of Things Advances in Wireless Technologies and Telecommunication - IGI Global (pp. 277-309).

Workshop Posters

5. Home Automation : HMM-based Fuzzy Rule Engine for Ambient Intelligent Smart Space, SICSA PhD conference, University of Dundee, 27-28 June, 2017.

6. Artificial Intelligence: Cognitive IoE(Internet of Everything) Approach to Ambient-Intelligent Smart Space, SICSA DemoFest , Dynamic Earth, Edinburgh, 2017.

LIST OF ABBREVIATIONS

AAL	Ambient Assisted Living
ACM	Ambient Cognition Model
AEM	Ambient Expert Model
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANFIS	Artificial Neuro Fuzzy Inference System
AMRUT	Atal Mission for Rejuvenation and Urban Transformation
ADL	Activity Daily Living
AmI	Ambient Intelligence
CoBra	Context Broker Agent
CAiSH	Cognitive Ambient Intelligent Smart Home
CIoE	Cognitive Internet of Everything
CPS	Cyber Physical System
HMM	Hidden Markov Model
IA	Intention Activity
IoT	Internet of Things
MQTT	Message Queueing Telemetry Transport
MSE	Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
M2H	Machine to Human
NB	NaiveBayes
P2P	People to People
RNN	Recurrent Neural Network
SHT	Smart Home Technologies
SVM	Support Vector Machine
SOCAM	Service Oriented Context Aware Management
SA	Spot Activity

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Research Background

The United Nations figures suggested that the world population would rise from 7.3 billion to 8.5 billion in 2030 and 9.7 billion in 2050. Therefore in 2050, sixty-five percent of the worlds population would be living in urban mega-cities and each megacity would be accommodating around 10 million inhabitants. Such massive urbanization of growing population would be known as a 21st century 'Urban Age' (Hui et al. 2017, Braun et al. 2017). On the other side, by 2020 the growing population of elderly people above 65 years, would be rapidly increasing by twenty-five percent in major EU countries and by thirty percent in other developing nations including Asia and North America. Therefore, such massive growth of the population and the elderly inhabitants in urban cities would require an assisted living environment for independent and comfortable living experience. As a result, a persuasive demand of assisted living environment would be vital to the humankind. The primary goal of assisted living environment would be supporting the aging population and inhabitants in urban megacities to live independently in their homes and communities with the support of proactive digital personal assistants. Therefore, the continuous growing demand of assisted living environment targets to improve the inhabitants comfort level and efficiency to do their ADL (Activity Daily Living) tasks by enabling the cooperation among various IoT smart objects which understand the environments, surroundings and their contextual needs in a proactive manner (Risteska Stojkoska and Trivodaliev 2017).

The rapid global urbanization is the key factor for smart cities development in the future with people-centric design approach in order to share resources effectively and intelligently. Although, the smart-city urbanization would be covering all aspect of inhabitant's age and physical abilities to perform their day to day tasks. However, the key element to providing tailor-made services or assistance to individual inhabitants in the urban age would be impossible without collecting and learning their personal activity-behaviors in public or private spaces such as smart homes, smart factories, and smart offices and even in public transport systems. Therefore, SHT (Smart Home Technologies) is in the EU's top priority list to create smart services, utilizing IoT technologies, comprising sensors, monitors, interfaces, appliances and network objects to enable automated, localized and remote controlled domestic environment (Hargreaves et al. 2017). Furthermore, SHT has the global market demand for smart appliances (including washing machines, fridge, dryers, ovens, and dishwashers) from USD 40 million to USD 26 billion by 2020 (International Energy Agency 2013). In Addition, Weber (2010) emphasized that IoT covering various aspect of internet connectivity in the physical world through distributed devices with sensing capabilities to manage, coordinate and deploy an intelligent set of services in our daily life. Hence, the innovation of IoT (Internet of Things) creates new opportunities for ICT(Information and Communication Technology) sectors services to seamlessly integrate IoT smart sensors-objects and electronic appliances in global infrastructure in order to provide better living experiences to humankind and society. As a result, IoT is becoming more things-oriented to link between physical and virtual worlds to serve humankind with enhanced services for independent assisted living experiences.

Furthermore, the IoT sensors technologies with the continuous research and development on wireless sensors network (WSN), Machine-to-machine (M2M) communication and Artificial intelligence have created a platform for various automated IoT services. According to Moore's law, the miniaturization of electronic components enabled the networked computer to be embedded into our day-to-day life and every possible aspect of the thing. As a result, things are now electronically connected with Internet and possible everywhere in the society, invisibly doing their task via machine-to-machine communication to serve humankind. In addition, CPS (Cyber-Physical System) seamlessly integrate physical components in connected cyberspace with the IoT technologies adoption. The term CPS, IoT and smart space are interchangeably used in ambient intelligent context, where monitor, control, and automation functions are accomplished through embedded connected sensors and actuators. The IoT smart technologies benefit various sectors of the society such as health care systems for elderly people and people living with various disabilities through connected monitoring services provided by a medical institution from a distance. Not only that, the transport services with embedded IoT sensors providing real-time road traffic information to navigate driver-less vehicles. Consequently, the smart IoT grids and meters are providing better decision capabilities for homes and industrial plants through collecting usage data to analyze and control the appliances for load balancing in eco-friendly manners.

To summarize, in the near future IoT technologies would enable homes to be ambient intelligent enough towards inhabitant's cognitive needs for assisted living experience and turn them into smart intelligent spaces. Particularly, an excellent smart home control system should not rely on the users' instructions instead it should be proactive in nature (Wanglei and Shao 2015). Therefore ambient intelligent systems are a sensational new information technology paradigm in which people are empowered for the assisted living environment through multiple IoTs sensors that are aware of inhabitant presence and context and highly sensitive, adaptive and responsive to their needs. Therefore, excellent ambient intelligent systems could be characterized by their ubiquity, transparency, and intelligence, which seamlessly integrated into the background and invisible to the surrounded users and inhabitants. However, few state of art IoT products (systems) were researched and proposed by industries and research institutions such as Google home, Alexa, i-Dorm, and CASAS but most of them were unable to embrace the pro-activeness into their architecture. As a result, ambient intelligence is being regressively researched for the next generation's IoT systems that would be able to fulfill the inhabitant's cognitive needs with its unobtrusively ambient intelligent service compositions. The ambient intelligent control system would be trained on performed (historical) activities data sets and empower the automated massive predictive analysis services which would aware of inhabitant's presence and context in a highly sensitive proactive manner. Therefore, the research is motivated to propose novel ambient intelligent models to build a seamless predictive system for IoT. The proposed system would be characterized by its activity identification and predictive analytics techniques to identify and forecast hidden ADL patterns. Along with, an associated rule-based system would be attached to perform task execution in a proactive manner like an expert system.

1.2 The IoT Opportunity: Digital Everything and Digital Everywhere

In recent years, impressive hardware technologies have been developed that let mobile and embedded devices to better exploit the web-Internet features to ensure an enhanced interactive experience with the physical world. As earlier, Satyanarayanan (2001) suggested that great technology inventions are those, who dissolve themselves into everyday life and be invisible from human consciousness. Such research themes are making futuristic scenarios of Ambient Intelligence and smart environments into the reality of everyday lives by integrating research contribution from the fields of context-aware pervasive computing, wireless sensor network, IoT and artificial intelligence. The smart spaces research extends the functionality of ambient intelligence towards more proactive possibilities, where the smart environment not only monitors people for tasks execution but also understand their plans and intentions. In Addition, the EU report suggested that the pervasive computing will be the new wave in ICT innovation and by 2020 pervasive computing will be one major type of ICT system. (Ricci 2015)

Many researchers around the world are regressively researching on Context-aware IoT Systems to investigate and propose the intelligent IoT architectures for smart homes, smart cities, and urban computing projects. Furthermore,Taylor et al. (2015) stated that there will be a significant change in the electronics industries due to the vast growing popularity of Internet of Things (IoT) sensor technologies. The tremendous opportunities to the electronics industry are to integrate the various aspects of IoT technologies in smart homes, smart meters, smart cities, autonomous transport system to offer improved lifestyle and en-cash the ready market and demand of IoT systems.

Overall as mentioned earlier that IoT is an enabling technology to establish more advanced communication channels between Machine to Machine (M2M) such as smart

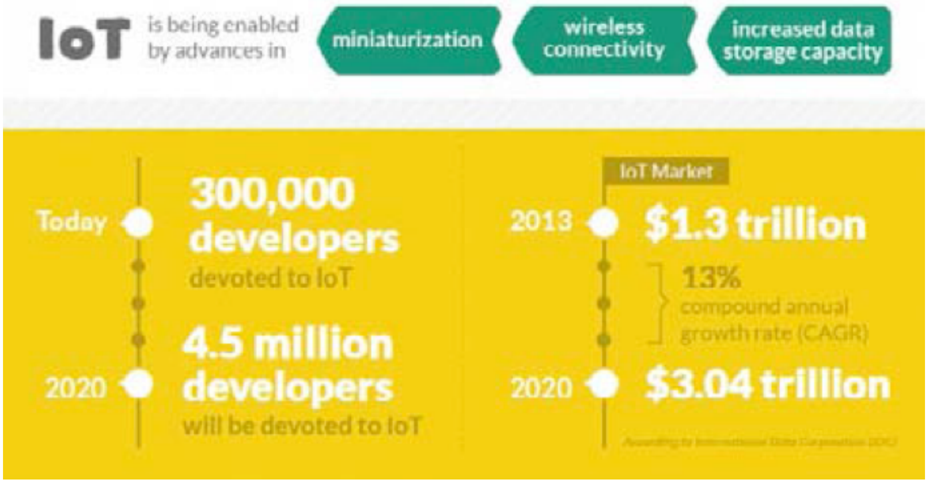


Fig. 1.1 Predicted growth of IoT Taylor et al. (2017)

wearables, automated smart homes, smart cities, smart manufacturing, health care systems and smart automotive industry.

In particular, M2M communication would be enhanced to respond inhabitant's everyday needs and desires through digital direction and control mechanism. Also Taylor et al. (2017) claimed that Imagine the day when the entire continent of Africa is complete, digitally connected and that day will come by 2025.

Several research institutes and blue-chip companies are investigating to capture this ready IoT market's demand. As per the figure 1.1, predicted growth in IoT would lead to cutting-edge innovations along with the boost in financial investment in IoT research around the world. In particular big IT giants such as IBM Watson, Samsung, Panasonic and Microsoft are working in large research groups and infrastructures to embrace the 2020 demand for advance IoT systems. The year 2011 when IBMs Watson won the Jeopardy championship trophy, did challenge the human cognitive intelligence on national TV to beat the human brains by rapid information extraction to find the right/relevant answer in much shorter time. Such incident created an alert indicating that knowledge engineering and context-aware computing could reveal inside patterns and relationships across large data sets to swiftly extract the key information to answer the specific question.

The IBM has already established a market trend for IoT and predicted that Internet of Things is at the threshold of a tremendous opportunity. As compared to the traditional computing systems, cognitive IoT systems are following the data mining approach to improve the learning environment and increasing the possibilities of what can be done with advanced analytics methods. As a result, that is making sensors capable of diagnosing and adapting to their environment without the need for humanitarian interventions. In addition, cognitive IoT systems provide the ability to combine multiple data streams that could identify hidden patterns and extract much more contextual information that would be otherwise available. The cognitive IoT and machine learning are further enabling enterprises to unlock IoT potential market in the future. An exploding amount of IoT sensors data streams require a new dynamic approach to collect and analyze the data to extract important contextual information out of it. Such a massive amount of information extraction from IoT sensors and devices could be applied to enhance existing knowledge, uncovering insights patterns, market trends and next generation decision making systems. Although IoT trends are driving digital transformation in consumer products such as wearables and connected electronics certainly at large part of the market share. The IDC (International Data Corporation) estimates that more than 80 percent of IoT spend in 2020 would be on B2B (Business to Business) applications and use cases systems. As a result, IoT technologies would be one of the primary drivers of the digital transformation that will undergo in the coming years, that would drive digital disruption in the physical world. Hence, advancement in IoT technologies would be a critical transformation for many industries to allow new business models to emerge and enables changes in workflow systems, productivity improvements, cost containment and enhanced customer experiences as the outcome results. (IBM Watson, 2018)

1.3 Key Challenges for Ambient Intelligent Environment

As Negnevitsky (2002) suggested that choosing the right tool for the right job is undoubtedly the most critical part of building an intelligent expert system for smart spaces. The process of building an intelligent system begins with gaining an understanding of the problem domain and determining the availability nature of input data from various IoT network objects. Particularly developing phase can be divided into six categories such as

i) problem assessment, ii) data and knowledge acquisition, iii) prototype development, iv) complete system development, v) revised evaluation and vi) Integration and maintenance of the system. However, by embracing the problem assessments as guidelines into fundamental architecture, still building an ambient-intelligent system is a very challenging task and need to dig deeper for solutions to tackle major gaps and loopholes in the system.

On the other side, Banafa (2017) argued that the main challenges of ambient intelligent systems include the Intelligent analysis of heterogeneous IoT sensors data and taking proactive actions. The accompanying artificial intelligent models enable complex event processing to analyze sensory data from real-time or near real-time series data streams for driving timely decision making and actions. However, the heterogeneous resources and smart objects must be standardized in dynamic architecture for flawless integration and composition for the value-added services to inhabitants. An AAL (Ambient Assisted Living) is built upon the ambient intelligent devices where inhabitant's presence, behaviours, and needs are captured through embedded heterogeneous IoT sensors in order to observe the environmental changes and respond them in an intelligent manner to carry out specific tasks (Technologies and Change 2014). However, the main criteria for an AAL system to be a smart IoT system are;

- **To extract contextual knowledge from heterogeneous IoT objects to anticipate ambient intelligence**

  The IoT system anticipates people's desires without conscious mediation. The context awareness must be embraced in the framework and deployed in IOT-AAL system. Various networked devices are integrated into the environment to collect and transform the raw data streams into contextual knowledge in order to reason in an intelligent way.Overall, the system recognizes the environment, related embedded devises, people and their situational context.

- **To forecast activity patterns in Complex IoT systems through regressive data analysis**

  The IoT architecture is composed of a big number of network objects (sensors and actuators) that must autonomously interact with the corresponding environment and the related users. Particularly, newly added IoT devices must start providing their functionalities. The processing and interaction with the different things will depend on the benefit provided by the different devices. In some cases, passive things will simply provide measurements. In other cases, objects may have larger memory, better processing, and advanced predictive analysis capabilities to make the system more intelligent and autonomous.

- **To develop an adaptive architecture with dynamic parameters for ADL task executions**
  The Internet of Things architecture should be adaptive in nature with reference to i) event-driven and ii) time driven. Some sensors produce data when an event occurs (e.g. door sensor, motion sensors); the rest produce data continuously in time series manner, based on specified periods (e.g. environment sensor).The IoT system follows event-driven approach with common rules for task executions. However, a system should be able to respond changes to people needs and the environment.

The current research, focused on the machine-learning application for predictive time series forecasting models to analyze and predict inhabitant ADL patterns. The integration of the anticipation and intelligence capabilities into IoT system would provide a platform, a better living experience, and services with automated rules for task execution in a smart home scenario.

1.4 Aim and Objectives of the Research

The research aims to develop a novel dynamic ambient intelligent framework including its related models and mechanism for Cognitive-IoE (Internet of Everything) based smart spaces. In the framework, a functionality of ambient intelligence would be extended towards more proactive possibilities, i.e., the smart IoE environment not only monitors people/devices for the eventual tasks and respond re-actively, instead it should be able to understand their intentions and situations in the surrounding environment. The dynamic framework consists of the sensors data (consistent/inconsistent state) generated from heterogeneous devices (sensors, RFIDs, network objects etc.) to extract inhabitants behavioral information and identify their intentions and situations. Later, such information is fed into machine-learning models that internally identify and predict the series of intention activities. The identified activity patterns will be mapped to the inference engine for particular rules set to achieve proactive task execution in the AAL(Ambient Assisted Living) environment. The research will demonstrate the utilization of IoT technologies integrated with machine learning models as two key areas of in-depth study : i) ACM (Ambient Cognition Model) for cognitive activities and situation identification and, ii) AEM (Ambient-Expert Model) for situation adaptation and decision making with predictive regression models for proactive task execution in the IoT environment.

The research aim would be delivered through the following research objectives:

- **Developing an approach to extract micro-level contextual information from IoT sensors unpretentiously and label discreet ADL (Activity Daily Living) states in a time series manner**
  The process starts with investigating the design and development methodologies of IoT technologies in ambient assisted living experience for the smart space environment. The knowledge extraction and modeling feature in existing CIoT frameworks and a comparison analysis between AAL System Architectures are the prime objectives. The data stream from embedded sensors in the environment leads a data-driven approach for deriving contextual information, therefore, a systematic approach of statistical model would be followed to normalize heterogeneous data sources. In particular, micro-level information would be combined to extract macro-level situations and intentions of inhabitant's activity states. The activities/situations identification problem in the ambient assisted living environment would be addressed in the proposed model.

- **Developing an approach to create and understand time series forecast model for activity pattern predictions with machine learning regression models**
  A supervised machine learning approach would be introduced to forecast time series activity patterns, which represents a measure that a specific set of activities reoccurs at constant time intervals. The training process would be applied on the model to

fed with time-series data to learn (train) and recognize (predict) the hidden ADL patterns of the inhabitant. The supervised machine learning regression models would be investigated, particularly the HMM (Hidden Markov Model), RNN (Recurrent Neural Network) and NB (Naive Bayesian). The self-optimization of parameter estimation abilities would enhance the probabilistic approach of such models in order to estimate maximum-likelihood activity patterns of inhabitant in IoT environment.

- **Developing an approach to integrate IoT eco-system's components as a unified framework for predictive task execution in a proactive manner**
  To provide assisted/independent living experiences, a number of IoT frameworks have been proposed. However, these frameworks could not sustain comprehensive and in-depth ambient intelligent services to the inhabitant due to the lack of focus on the "pro-activeness" aspect. Therefore, the third objective would focus on investigating and proposing a novel ambient intelligent framework for Cognitive IoT smart home to execute the various task in a proactive manner. Models and algorithms from the first and second objective would be integrated as a unified framework for automated task execution like a rule-based expert system.

- **Developing a non-intrusive IoT prototype to capture inhabitant's ambient activity patterns** A non-intrusive system to observe inhabitants activity would play a vital role to record data log of inhabitant's ADL in a time series manner. Despite a wide range of publicly accessed ADL datasets are available but they are only capable of handling event-based activities log and undermine the time series data log importance. Indeed, this is mainly due to their restricted IoT devices parameters that fail to capture activity states in time a series manner. In addition, most AAL products and devices are available as black box components, where further modifications are restricted for developers due to proprietary use by the device manufacturers. As a solution to the problem, the proposed IoT prototype would enable and provide full access control of functionalities and code modifications to the end user (developer) in a cost-effective manner. While in existing AAL products, no new functionalities, sensors, or actuators can be straightforwardly added to the original setup (Memon et al. 2014, Stengler et al. 2015)

1.5 Contribution to Knowledge

The research overcomes the existing limitation of the IoT systems by offering a dynamic ambient intelligent system to enhance assisted living experiences for inhabitants in IoT environment. Further, it fills the research gaps by providing a unified framework for cognitive IoT systems by integrating ambient intelligent models. The thesis embraced the machine learning data-driven approach to integrate and solve the IoT data streams problems in order to provide more scientific and systematic inhabitant activity patterns recognition compared to the traditional knowledge-driven approach. Accordingly, they result in a series of contributions : firstly, a unique ”Ambient Cognition Model” to provide an innovative means of activity identification model towards precise and systematic activity labeling task: secondly, a novel ”Ambient Expert Model” for predictive time series forecasting to perform task execution in a pro-active manner like a rule-based expert system. The proposed models are capable of describing the diversity in activities patterns and labeling them into discrete entities for the predictive time series forecasting analysis. Thirdly, the integration of both ACM and AEM models into a unified framework, enable an effective means of cognitive ambient intelligent smart home services for enhanced assisted living experience in IoT environment. More specifically, the contributions are described as follows:

1. ACM (Ambient Cognition Model) for discrete activities identification and labeling

The thesis proposes a novel technique for ADL (Activity Daily Living) identification and analysis in the cognitive ambient intelligent environment. The data from heterogeneous IoT sensors create a fundamental classification problem in cognitive IoT systems where developing a higher level contextual information is a tedious task. With an effective statistical model, ACM provides fundamental data preprocessing and activity identification ability to the IoT system. This consequently labels the identified and discreet activity states in order to provide the time-series data for the machine learning experiments.

The contribution is linked with publication in IoT-BDS conference paper. The proposed ACM architecture is embrace the novel technique for ADL identification and received valuable feedback and comments from the science community.

2. AEM (Ambient Expert Model) for automated predictive time series forecasting and task execution

In contrast with other existing IoT service operation specification framework and models, a novel framework is proposed to adopt the ADL time- series data sets in order to design and build ambient expert system for predictive analysis. A supervised machine learning approach has been adapted to forecast ADL patterns in time series fashion to perform the task in a proactive manner. The system not only observes inhabitant activity states but also forecast their activity intentions in term of most likelihood patterns. In addition, a rule-based system is leveraged to automate task executions. As an expert perform work in a specific situation, rule-based system mimic the decision making capability of an expert in if/else conditions. An antecedent and consequent relationship between IoT objects maintain the task coherency and formalization to leverage the ambient assisted living experience in the embedded IoT environment.

The contribution is linked with second publication for SEKE conference, which proposed a novel Ambient Expert Model for ADL forecasting in modern IoT systems. The proposed AEM architecture received valuable feedback from the critics.

3. CAiSH (Cognitive Ambient-Intelligent Smart Home) as a unified framework for enhanced AAL (Ambient Assisted Living)

Based on the above two intelligent models, a unified architecture is proposed, to integrate ACM and AEM components into a complete framework to address the requirements of the automated smart home. This architecture accommodates a proactive cross-domain ADL pattern recognition and prediction capabilities to the modern IoT systems. The framework of CAiSH incorporates the management of micro-level spot activities to the macro level complex activities state in order to identify the inhabitant intentions and situations proactively to perform the task via the rule engine in the ambient intelligent environment. A prototype implementation approach has been taken to deploy IoT sensors in order to capture the inhabitant activities and states in real time scenarios.

The third contribution to knowledge is submitted as a journal paper for Consumer electronic with the title of Cognitive Ambient-Intelligent Smart Home framework.The unified framework of CAiSH integrate the ACM and AEM architecture. The valuable feedback from reviewer has been embraced into the paper and considered for resubmission.

Overall, the statistical model for heterogeneous sensor activity identification and the data-driven machine learning regression models makes thesis contribution valuable to the modern intelligent IoT systems. The proposed models and frameworks enhance the assisted living experience in smart home scenarios and could also be further applied in other sectors of IoT where contextual information is vital for the system to function in a proactive manner.

1.6 Research Methodology

The thesis adopts a combination of research methods including literature review, quantitative measures and artifact production with use case study approaches.

Initially, a comprehensive review of philosophical literature is undertaken with regard to traditional context-aware computing, pervasive computing, IoT systems, data-driven approach for pattern recognition and machine learning algorithms applications and evaluation techniques. Alongside, various implementation use case study experiences are meticulously studied for the in-depth understanding of IoT systems to build a strong research knowledge base. Through in-depth review and analysis of the latest works of literature, several issues and limitations are found on existing context-aware IoT system specifications and relevant modeling techniques. These lead to the design and development of the series of novel approaches proposed subsequently. Later, following the data-driven approach, a real-time IoT (Internet of things) prototype has been developed for time-series data collection. A quantitative measure of system training and testing is applied in a scientific experiment in the MATLAB simulation. Developing a real-time IoT system and deployment given the enormous amount of implementation experience and challenges in the IoT systems. The supervised machine learning algorithms training with accuracy measures are conducted to evaluate the functionality, effectiveness, and efficiency of the proposed approaches. Consequently, research papers have been published for peer review based on the research outcomes at each milestone. This enables valuable assessments of the work from other researchers in terms of contribution and justification within the field

1.7 Structure of the Thesis

The thesis is organized as follows:

Chapter 1 outlines the introduction of the research with the opportunity and challenges, problem statement, motivation, aim, and objectives of the research, the contributions to knowledge and the statement of methodology.

Chapter 2 includes the in-depth literature review of existing cognitive IoT systems, context-aware pervasive computing, machine learning application for ADL pattern recognition and various agent-based systems for ambient intelligent environments are successfully conducted. The current state of machine learning approach of IoT system to provide assisted living experience summarized and critically analyzed.

Chapter 3 proposes a statistical model approach for ACM (Ambient Cognition Model) to enhance ADL identification at micro level spot activity to the complex level intention activity divisions. A real-time series IoT sensor data preprocessing has been a key player for further data analysis in cognitive IoT pattern recognition tasks.

Chapter 4 proposes an Ambient Expert Model (AEM) framework, based on the supervised machine learning approach for recognizing and predicting the most likelihood ADL patterns with the association of appropriate rule-sets for task execution in a pro-active manner.

Chapter 5 proposes a unified architecture of CAiSH (Cognitive Ambient Intelligent Smart Home), formally defines the situation, intention identification problem and proposes a pattern recognition and prediction model by leveraging machine learning techniques.

Chapter 6 presents the case study implementation of real IoT sensors deployment in the smart home. With the help of microcontrollers and IoT sensors, a low-cost IoT prototype development and deployment succeeded for information collection in the real-time data sets in a time-series manner.

Chapter 7 performs the supervised machine learning experiments on discreet time-series activities data sets. The pattern recognition and forecasting intention-activities in time-series fashion are trained and tested for their accuracy measures in MATLAB simulation. The evaluation process has been performed against accuracy measure against the real-time testing data sets.

Chapter 8 presents the conclusions of the research and the future work.

CHAPTER 2

Literature Review

2.1 Cognitive Internet-of-Things (CIoT) system

The Cognitive Internet-of-Things (CIoT) is a new state-of-art computing paradigm for interconnecting (physical or virtual) things/objects in a context-aware perception-action cycle. The interconnected objects (sensors, RFID, network objects) behave as agents to learn, think and adapt situations according to the dynamic contextual environment with no or minimum human intervention (Vlacheas et al. 2013, Wu et al. 2014). As a result, the world has entered in the Smart ERA, where every IoT network object plays an important and intelligent role in our daily lives. The concept of ambient intelligent smart space represents the embedded environment, surrounded by numerous sensors to monitor, sense and actuate the space. Every Internet-of-things(IoT) network object generates trace and vital information in the network, and every IoT network object has the capabilities of machine-to-machine (M2M) and machine-to-human(M2H) communications. The enormous amount of data generated from numerous IoT sensors and devices, provide the most interesting research platform of pattern recognition in the smart IoT ecosystem. The investigation in such sensory data could lead to the design and development of new type of ambient intelligent IoT systems and services to ease inhabitant daily lives with enhanced assisted living experience. However, the major challenges in cognitive IoT system are to extract the contextual knowledge from the IoT sensors data to identify inhabitant's activity intentions and the situation in the surrounding environment. The blend of machine learning and Internet-of-Things takes cognitive IoT research to the next level in order to innovate and implement the new ideas of smart cities, smart homes, smart healthcare systems and smart urban computing platforms.

The application of machine learning with the Internet of Things technologies provides a fundamental base for the research of cognitive-IoT system with ambient intelligent capabilities. Admittedly, Wu et al. (2014) claimed that without cognitive capability, IoT is just like an awkward stegosaurus: all brawn, no brains. To fulfill its potential and deal with growing challenges, it must take the cognitive capability into consideration and empower IoT with high-level intelligence as we say brain-empowered Internet of things or cognitive Internet of things. In contrast, as human mind takes input from outer world to sense, hear, touch taste and smell to make a decision and initiate sophisticated coordinated actions as results. In cognitive computing, the aim is to mimic human decision making capability with the help of computational theory. On the other hand, Cognitive Internet of Everything is a new network paradigm, where (physical/virtual) things or objects are interconnected and behave as agents. Particularly, with the minimum human intervention the things interact with each other following a context-aware perception-action cycle. The prime objective of Cognitive IoT system, is to bridge the gap between physical world (with objects, resources, etc.) and the social world (with human demand, social behavior, etc.) to develop an intelligent physical-cyber-social (iPCS) system by enabling smart resource allocation, automatic network operation, and intelligent service provisioning. (Wu et al. 2014)

However, in the context of providing an intelligent smart spaces for CIoT enabled environment, Vlacheas et al. (2013) emphasized on key characteristics such as ; i) consideration of perception-action life cycle in whole ecosystem, ii) appropriate normalization methods for handling massive data streams, iii) extraction of knowledge base for developing intelligent decision-making system and finally, iv) provision of on-demand service platforms. In the cognitive-IoT enabled environment, the interaction between machine to machine performed in a more smarter way to communicate and provide context-aware services with minimum and sometimes no human interventions. Particularly, the perception-action cycle plays an important role in anticipating environmental needs as input from physical sensors and providing appropriate services as a solution. While setting the key characteristics of an ideal CIoT system, Vlacheas et al. (2013) believed in a slightly different view and argued to focus on three major areas as i) hiding heterogeneity of network object connections, ii) dynamic services provision with resiliency, iii) proximity assessment through relevance matric for IoT application and dynamic management of scalable virtual objects. Beyond the definition of key characteristics of ambient smart applications for CIoT enable environment, a broker-centric agent architecture, known as 'CoBra' is proposed by (Chen et al. 2003), where broker agents play a key role to maintain contextual information with user-defined policies on behalf of the community of agents and devices. The CoBra system focused more on the aspects of: i) contextual knowledge base, ii) context acquisition for reasoning engine and iii) user-defined policy management, where the resource hungry task for reasoning contextual information has moved to centralized, rich resources brokers from resource-limited agents.

In comparison to other researchers work, Gu (2004) has a different state of mind to embrace context-awareness in modern Cognitive IoT systems , therefore a service-oriented context-aware middle-ware architecture 'SOCAM' has been proposed by them to support context reasoning such that high-level implicit contextual information is derived from low-level explicit contextual information, hence applications would have direct impact derived from low-level explicit contexts. The dealing with contextual information interpretation from various information sources, makes 'SOCAM' a very effective Service-oriented context-aware system, where the completed framework is subdivided into four different layers such as, i) contextual information provider, ii) information interpreter, iii) context knowledge database, and iv) context-aware services provision with location awareness. while on the other side, Lee et al. (2010) offered service-oriented architecture based contextual information management framework in manager and agent relationship as context reasoning task performed by the manager and Q-broker utilized for service composition, adaptation, and re-configuration. On the other hand, Hong et al. (2009) utilized historical information to analyze user preference and habits in order to provide more customized tailor made context-aware services, with subdivision of the main framework into four different aspects such as, i) data gathering, ii) context management, iii) preference management and iv) service application layer for more customized solutions as services outcome.

While, Taylor et al. (2015) have a different opinion that, cognitive computers are expected to learn through experience to find a correlation, create a hypothesis and remember outcome to learn from. Such cognitive computing mimic the human brains structure and synaptic plasticity. They proposed a SyNAPSE (System of neuromorphic adaptive plastic scalable electronics) project to create a system that not only analyses complex information from multiple sensors but also dynamically rewires itself according to the environment while keeping reduced computing complexity and power usage.

Therefore, we are living in a world where worldwide effort from the academic community, service providers, network operators and standard development organization etc. contributing their effort to make life easier. Briefly, cognitive Internet of Things enhances the current Internet of Things by mainly integrating the human cognition process into the system design. The advantages are multi-fold, e.g., saving peoples time and effort, increasing resource efficiency, and enhancing service provisioning, to just name a few of many more. As Wu et al. (2014) proposed framework of CIoT and Fundamental Cognitive tasks in figure 2.1, the CIoT serves as a transparent bridge between physical world (with general physical/virtual things, objects, resources etc.) with Cyber world for data semantic knowlege layer and social world for human demand, social behavior and service provisioning. The intelligent physical-cyber-social worlds work together to embrace five fundamental cognitive tasks such as Perception-action cycle, Massive data analytics, Semantic derivation and knowledge discovery, Intelligent decision making, and finally on demand service provisioning.

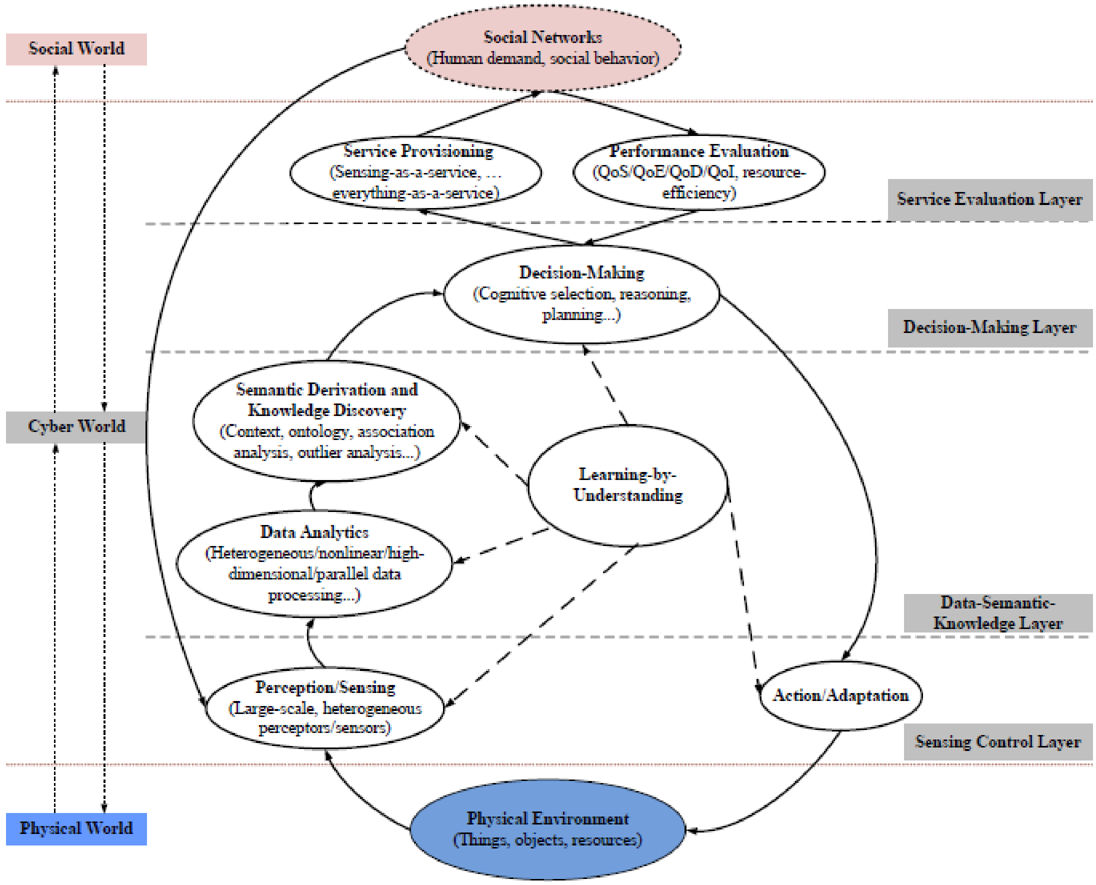


Fig. 2.1 Framework of the Cognitive Internet of Things (C-IoT) (Wu et al. 2014)

Likewise, Guo et al. (2011) emphasized that such embedded intelligence in Cognitive IoT systems aims to extract individual behaviors, space contexts, and social dynamics from off-the-shelf or emerging smart things. The various smart things weave themselves deeply into the fabric of everyday life. The diverse features of them, nevertheless, present us unprecedented opportunities to understand various aspects of interaction patterns between human and real-world entities, incorporating human-object interaction, human-environment interaction, and human to human interaction. The learned embedded intelligence cannot only, from the micro-scale, improves the quality of human life by anticipating user needs and environmental changes, but also from the macro-scale, provides real-time decision support for the crowd as well as urban planning managers. The rapid use interconnected network objects sensor, RFID, body-wearable sensors, are empowering cognitive IoT (internet of thing), where artificial intelligence provides foundation as build blocks to design such automated IoT systems, which can be applied in vast areas such as elderly health care system, Smart cities, smart transportation and manufacturing industries etc..In current scenario of Cognitive internet-of-things(CIoT) system, artificial intelligence play a key role for making system more ambient-intelligent to the environment. In contrast, the capability of the system to forecast future behavior or abnormality in the surrounding environment is remarkably becoming attractive to many applications. Therefore, in the complex environment, heterogeneous IoT sensors generate an enormous amount of data about various activities and events. As a result, the major challenge arises to extract the contextual information from the system to forecast the probability of the future event. Hence, the application of machine learning models provides the solutions to makes Cognitive IoT systems more ambient intelligent to forecast and understand the environmental situation by Forkan et al. (2015)

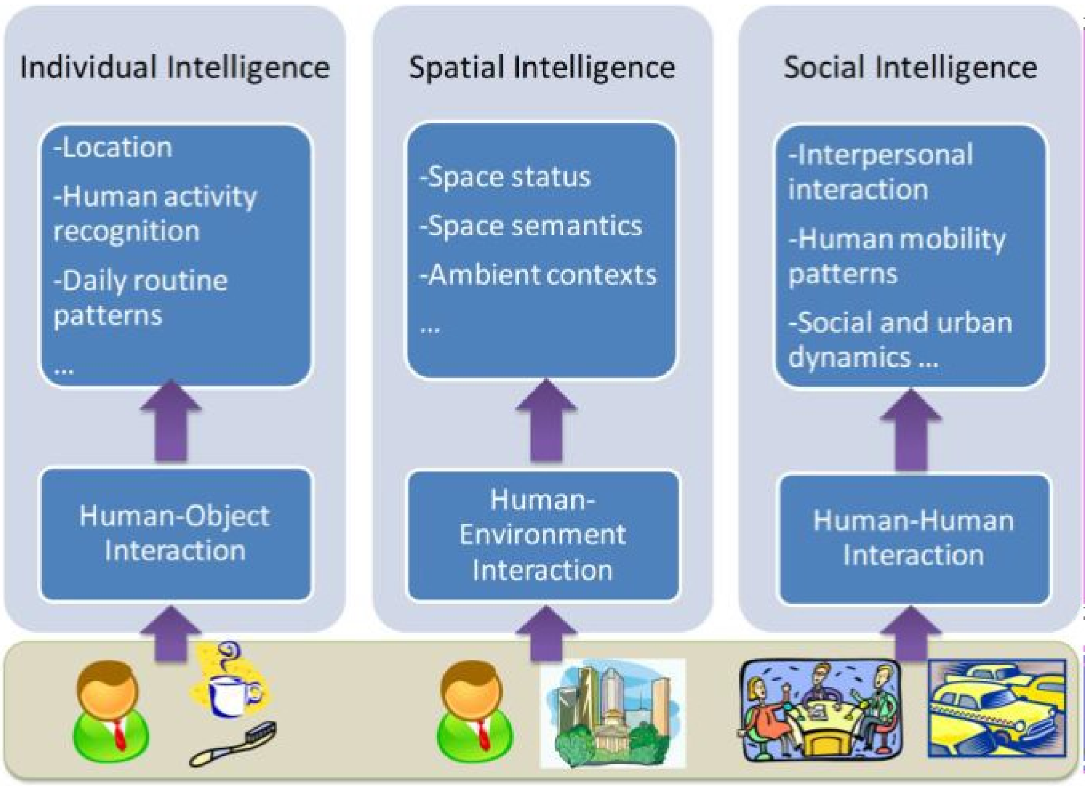


Fig. 2.2 The characteristics of embedded intelligence for CIoT

To summarize the context awareness in the Internet of things, a pro-active system is required which should have the intelligence to understand the inhabitant's intentions. In other words, the elimination of human in loop cycle for developing unobstructed ambient intelligent services, where the goal of the human-centric system changed to the human-supervised system (Tennenhouse 2000). The pro-activeness in modern cognitive IoT systems would create a valuable impact to anticipate users needs in IoT environment with a hypothetical approach to understand user's goals and intentions in context-aware intelligent systems (Want et al. 2003, Salovaara and Oulasvirta 2004). As a result, in near future context-aware, pervasive computing would be strong enough to provide pro-active services as a solution for the ambient intelligent environment without no or minimum human instructions (VanSyckel and Becker 2014).

However, choosing the right tool for right job is the most critical task for building a solution. Such as gaining knowledge from the work domain and understanding the contextual meaning of heterogeneous sensors data are the most import factor to build a dynamic ambient intelligent system. The (Negnevitsky 2002), outlined the development process of cognitive IoT system in various steps including i) problem assessment, ii) knowledge acquisition from raw data, iii) prototype development and iv) the revised evaluation of complete system with integration of maintenance process. The revised and learned lesson from previous researcher's work failures and gaps would be embraced into new research architecture. However, the challenge is big enough to spend few years to investigate and dig deeper for major problems of ambient intelligent IoT systems such as semantic description and relationship among heterogeneous sensors data stream.

2.2 Role of Ambient Intelligence in CIoT systems

In the current research scenario, ambient intelligence is still a relatively new and emerging field where several researchers are exploring the perceptions of ambient intelligence applications in a cognitive IoT environment. In modern days of IoT systems, ambient intelligence used as an umbrella concept for a multitude of network devices, objects, automated services information and communication technologies (ICT). Which are aiming to enhance the assisted living environment experience for inhabitants and supporting their mobility by means of innovative technologies (Braun et al. 2017). In addition, such ambient intelligent technologies nowadays used for supporting, assisting, preventing and improving wellness and health conditions for different categories of people, mainly the elderly ones (Stengler et al. 2015). A smart environment can be presented as a regular looking home by the installing different types of sensors, actuators and a diverse intelligent system where the information from the different sensor will help to provide a detailed overview about the environment status. In fact, the deduced context will be mainly used for providing, guiding and orienting the inhabitant's behaviors, and decisions toward safer, healthier and more comfortable living environment (Stengler et al. 2015). However, the aim of ambient intelligence in Cognitive IoT systems is to provide comfortable assisted lives to inhabitants, therefore different areas of society and business domain can get benefit from such automated systems. In broader term, an ambient intelligent system fulfill four major criteria such as : (i) to capture important information about inhabitant's ADL routines, (ii) to perform required data mining analytics to extract contextual knowledge from raw data streams, (iii) respond accordingly to the physical environment and , (iv) learn and update knowledge base for better results in the future (Friedewald et al. 2005).

In contrast, supporting people in independent living using ICT is a great opportunity to solve upcoming society challenges. Primarily these are related to the aging of the European population where ambient intelligent solutions aim to increase the quality-of-life that leads to increased demands for comfortable and luxuries life. As the web portal Technologies and Change (2014) claimed that ambient smart system aims to anticipate user needs, according to the situation they are, by means of semantically understanding the environment status and on-the-fly composition of different service components. This requires applications to be able to understand the context and situation the user is in. Such a theme has been addressed within the ambient intelligence, ambient assisted living and pervasive computing fields, leading to a number of solutions able to leverage contextual information coming from a number of sources.

Whereas, the purpose of IoT is to connect the multiple of network objects and sensors together to adopt environmental changes into the smart system. An ambient intelligent environment requires suitable machine learning techniques and a well-defined architecture to blend technologies from artificial intelligence to embedded systems to provide smart services. In addition, Sasidharan et al. (2014) argued that the domain of IoT application is modeled to detect and anticipate future potential situations using machine learning mechanisms and react on contextual situations based on their learning outcomes and reasoning, implementing actuation and automation to respond situation and adapting user preferences. They have explored the concept of visualizing Real-World Object (RWO) for a seamless integration between network objects to provide realistic and sophisticated services, they divided cognitive management into three layers such as Virtual Object (VO), CVO and Service level. In addition, Hassan et al. (2013) suggested the criteria of the ambient-intelligent system should be able to extract the knowledge base from large about of raw data sets, where information is dynamic in nature as compared to offline system. They also emphasized on Adaptive Fuzzy Inference System (AFIS), which follows the HMM and fuzzy logic principle to predict nonlinear time series data about forecasting stock prices indices. On the other hand, Liouane et al. (2016) suggested the structure of the multi-level stochastic process of recognizing activity daily living(ADL) patterns in pervasive IoT systems can be visualized as a tree-structured variant of Hidden Markov model. The relationship between log likelihood predicted state and real activity states are key impact factor of an adaptive Hidden Markov model for the pervasive and smart environment. (Liouane et al. 2016)

The current research work about ambient intelligent environment is not limited to smart homes but also have its important impact on the assisted living environment in healthcare scenario, smart cities etc.. However, the existing contribution in knowledge base for Cognitive IoT systems is well embraced in the physical and social layer but the analytical part is not investigated at in-depth level to provide the proactive ambient-intelligence and decision making capabilities. In the proposed work by Braun et al. (2017), Sasidharan et al. (2014), Hassan et al. (2013), Liouane et al. (2016), Wu et al. (2014) an universal design approach have been missing for Ambient Assisted Living (AAL) capabilities, where design and development of ambient intelligent systems, can be used in all sort of ambient assisted environment by every individual, no need for specialized design or further adaptation to make changes at a certain extent. The usefulness of ambient intelligence concepts in hospital, school, cities is evident a priori. The intelligence factors can be listed into four categories such as unobtrusive, personalized, adaptive and anticipatory. Therefore, cognitive IoT systems should be embedded in an environment in such a way that it should not intrude inhabitant consciousness unless required implicitly. However, being invisibly working in the environment, the system should be able to alter or tailored as per their needs to provide personalized services. In addition, adaptiveness of system provides a solution for behavioral changes in inhabitant actions and surrounding environmental factors. The ultimate meaning of ambient intelligent system finally embraced by anticipatory features, where the system can anticipate the inhabitant desires and environmental factors without any interference of inhabitant to a certain extent. As a result, intelligence comes into the system to differentiate between nature and unnatural activity state based on some initial information collected from continuous observation of inhabitant daily living activities. Also, in the case of the multi-resident environment, a system should be able to provide solutions based on individual preferences. (Burzagli and Emiliani 2017)

2.3 Activity Daily Living (ADL) Patterns

Activity recognition has been attracting increasing attention in a number of related research areas such as pervasive computing, intelligent environments, and robotics. It is also driven by growing real-world application needs in such areas as ambient assisted living and security surveillance. Activity recognition is the process whereby an actors behavior and his/her situated environment are monitored and analyzed to infer the undergoing activities. It comprises many different tasks including i) activity modeling, ii) environment behavior monitoring, iii) data processing and iv) pattern recognition. More specifically performing activity recognition task, it is necessary to; (a) create computational activity models in a way that allows software systems/agents to conduct reasoning and manipulation, (b) monitor and capture a users behavior along with the state change of the environment, (c) process perceived information through aggregation and fusion to generate a high-level abstraction of context or situation, (d) decide which activity recognition algorithm to use, and finally, (e) carry out pattern recognition to determine the performed activity. The interactive information between people and environments can be obtained from deployed sensors and the cameras.

However, monitoring inhabitant behavior in IoT environment along with changes in the environmental factors is a critical task in activity recognition. This monitoring process is responsible for capturing relevant contextual information for activity recognition to infer inhabitant's activity. In contrast, Prieto et al. (2017) suggested that home automation system primarily focuses on, inferring the environment situational change and inhabitant intention for his/her physical activity state to detecting abnormal activities patterns. later such information is used for automating home appliances to enhance their assisted living experience. According to the type of the sensing devices and data type of these monitoring facilities, there is currently two main activity recognition approaches; vision-based activity recognition and sensor-based activity recognition. Whereas Chen and Chen (2017) strongly argued that, when it comes to privacy and user acceptance, sensor-based activity recognition is more preferable over vision-based activity recognition. The vision-based activity recognition uses visual sensing facilities, e.g., camera-based and compared to sensor-based activity recognition exploits the emerging sensor network technologies to monitor inhabitants behavior along with their IoT enabled environment. (Khalil and Linz n.d., Chen and Chen 2017)

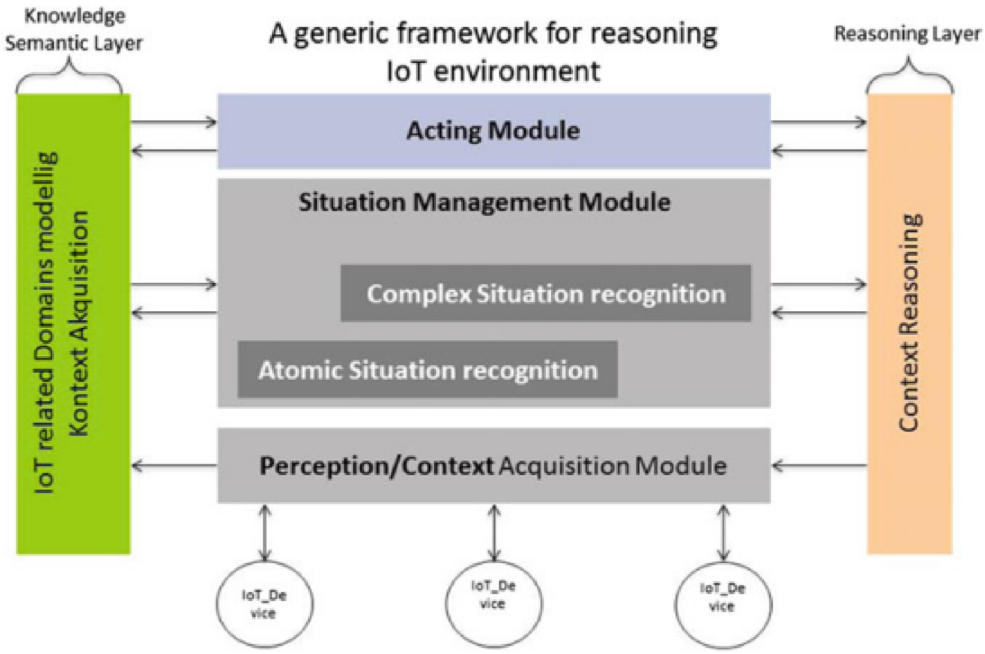


Fig. 2.3 Generic framework for the IoT system reasoning (Technologies and Change 2014)

On the other hand, Gu et al. (2011) claimed that human activity recognition are categorized in three different stages which provide information at i)action level, ii)activity daily living(ADL) level, and iii) higher context level. At the action level, physical movement of a user in the surrounding environment provides micro level information. The ADL level ensure the daily routine task information such as walking, sitting, reading and cooking. The higher contextual level information includes the overall anticipated activities of several ADL routines, which are merged into several coarse-grained categories, such as Information Leisure, Personal, Cleaning and Relaxing etc.. Such subset of information about action level, ADL level and Contextual informations provide the analytical research ground for machine learning experiments. Nowadays, machine learning models are focusing on logic-based and probabilistic model based approach to identify the hidden patterns of inhabitant's ADL routines. These days, the probabilistic model approach is gaining more popularity due to IoT sensor uncertainty in readings the noisy data collection and the human nature of performing nondeterministic activities. Furthermore, probabilistic model approach is categorized into static classification and temporal classification methods. In the static classification method, a variety of features are extracted from the sensor readings, and then a static classifier is applied to classify activities using naive Bayes, decision tree, k-nearest neighbor, and support vector machine classifiers etc.. Since, many activities share the common features, the temporal classification apply state-space techniques to infer the hidden activity states using Hidden Markov Model, Dynamic Bayes Naives and recurrent neural networks. However in temporal classification methods Hidden Markov Model achieve the better results with higher accuracy and less computational resources compare to other classifiers(Gu et al. 2011).

2.4 Data-Driven Approach for ADL Pattern Recognition

The sensor-based data drive approach are divided into two major machine learning techniques, supervised and unsupervised learning methods which primarily use for probabilistic and statistical reasoning. In supervised learning labeled data used for training the algorithms are then able to classify new dataset and compares results with testing data set for accuracy and overall output performance. The general procedure of supervised learning algorithm for activity recognition includes i) acquire sensor data representative of activities with labeled annotations , ii) determine the input data features and transform them into the application- dependent features, e.g., through data fusion, noise elimination, dimension reduction and data normalization, iv) divide them into a training set and a testing sets v) train the ML algorithm iteratively and finally (vi) test performance of the trained model against the testing sets. There are a wide range of algorithms and pre-trained models for supervised machine learning and activity recognition process, including Hidden Markov Model, NaiveBayes, recurrent neural network and Support Vector Machines. (Khalil and Linz, 2011)

On the other hand unsupervised learning, construct models for unlabeled data. The general procedure for unsupervised learning typically includes: (i) to acquire unlabeled sensor data, (ii) to aggregate and transforming the sensor data into features , (iii) model the data using either density estimation or clustering methods. Unlike the supervised learning, unsupervised learning has no correct answers . Algorithms are left to their own devices to discover the interesting structures in the data. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about it. The algorithms for unsupervised learning may include the use of graphical models and multiple clustering models including probabilistic reasoning. The main difference between unsupervised and supervised probabilistic techniques is that, instead of using a pre-established stochastic model to update the activity likelihood, unsupervised learning algorithms keep a trace of their previous observed experiences and use them to dynamically learn the parameters of the stochastic activity models. This enables them to create a predictive model based on the observed agents activity profiles. (Khalil and Linz, 2011)

2.5 Time Series Forecasting Methods for Ambient Intelligent System

In general, the time series forecasting is a technique for the prediction of events through a sequence of time intervals. The technique is used across many fields of study, from the geology of behavior to economics. The techniques predict future events by analyzing the trends of the past, on the assumption that future trends will hold similar to historical trends. Time series forecasting is performed in a variety of applications including weather forecasting, earthquake prediction, astronomy, statistics, mathematical finance, econometrics, pattern recognition, signal processing, control engineering etc. . Time series forecasting is sometimes just the analysis of experts studying a field and offering their predictions. In many modern applications, time series forecasting uses machine learning algorithms of the recurrent neural network (RNN), support vector machines, logical regression , hidden Markov models and NaiveBayes Classifiers etc.. In particular, forecasting starts with a historical time series data, where analysts examine the historical events and check for the patterns of time decomposition, such as trends, seasonal patterns, cyclic patterns and regularities in the data sets. Many areas within organizations including marketing, finance, and sales use some form of time series forecasting to evaluate technical costs and consumer demand in the future markets.

Such forecasting ability implemented in the smart home environment by Wu et al. (2013), where the spatial feature and temporal features of activities play important parts in activity forecasting. The former can be used to recognize where the corresponding activity of interest takes place whereas the latter can be used to determine whether some changes occur in the activities and use them to discover useful patterns. Naturally, a spatial feature refers to how an activity interacts within smart homes, e.g. what appliances are involved in and temporal features refer to when and how often an activity occurs in smart homes, e.g daily occurrence duration. The machine learning has been providing building blocks for modern IoT systems to solve real-world problems of classification, decision making, and pattern forecasting tasks. (Ramos et al. 2008)

In addition, building an adaptive system for ambient intelligent space have used common approaches to create learning model which include machine learning techniques Hidden Markov model, support vector machine, nave Bayes classifier, neural network and fuzzy logic and Adaptive Neuro-Fuzzy Inference System (Forkan et al. 2015) . Likewise, Walek and Janošek (2014) applied fuzzy approach, but were unable to acknowledge temporal attributes into the fuzzy adaptive system. Every machine learning technique is unique in its implementation and has own pros and cons. For instance, ANFIS (Adaptive Neuro-Fuzzy Inference System) system is not suitable for creating an event-based system because of temporal information of each activity instance cannot be applied in system training to recognize inhabitant activity patterns due to performance constraints Lee et al. (2013). Instead, in such activity recognition models, a method of sequential pattern mining could be more effective for discovering user action sequences. Subsequently, Cho (2016), proposed a system to forecast users next location utilizing k-Nearest Neighbour (kNN) and decision tree (DT) algorithms which are used for recognizing the current location from the information of T_n and S_n the time and transportation mode when user visits the n^{th} location, respectively. However, Cho (2016) claimed that K-means clustering is not suitable for the real-world system because it requires the prior knowledge of numbers of clusters (K). Instead, using the G-Means (Gaussian-Means) method is more suitable , which is based on a statistical test for the hypotheses that a subset of data follows a Gaussian distribution. Furthermore, G-means clustering method is useful for automatically finding the number of clusters that satisfies the Gaussian distribution.

2.5.1 Fuzzy Logic application for handling uncertainty and vagueness

In a human cognition system, some values and experiences are not crisp in nature but rather more vague and ambiguous in their terms at some level of uncertainty. Such vagueness-fuzziness in the cognitive IoT systems and pervasive-computing environment creates uncertainty and ambiguity in sensor's generated data. The fuzzy set theory apply membership functions to define that what degree of an element belongs to what fuzzy set values. In particular, fuzzy-logic embraces the idea that all things admit the degree of similarity/fuzziness to each other and can be described in a linguistic variable such as low, medium and high. The formulation of such multi-valued logic for approximate reasoning in a formal system called fuzzy logic. With the help of domain expert knowledge and user experience, fuzzy values can be formulated in their membership values for low, medium and high to reflect the fuzziness in the data-points. The fuzzy logic provides a mathematical formulation for dealing with imprecision and uncertainty in term of a degree of membership functions whereas crisp values of classical binary logic fail. For instance, 15 C (degree) temperature could be transformed into fuzzy sets and inferred as cold with the fuzzy value 0 to 5, and 4 to 9 'lukewarm', and 8 to 18 as warm. Therefore, fuzzy logic provide the formulated way to deal with knowledge symbolically expressed by a human observer. The Fuzzy logic supports the logical operations including intersection, union, complement, and modifier of fuzzy sets to manage information overlapping much easier (Ye et al. 2012, Negnevitsky 2002, Zaheeruddin and Garima 2006).

Furthermore, in the fuzzy theory, fuzzy set A of universe X can be defined by function $\mu_A(x)$, known as membership-function of element x in the subset A.

$\mu_A(x) : X \to [0, 1]$

where

$\mu_A(x) = 1$ if x is totally in A;

$\mu_A(x) = 0$ if x is not in A.

$0 < \mu_A(x) < 1$ if x is partly in A.

The value between 0 and 1, represent the degree of membership ($\mu_A(x)$), known as membership value, of element x in set A. A membership degree(μ) is thus defined as the degree to which an individual is considered to be the instance of a class. The value in interval (0, 0.5) means the statement is less likely to be true and [0.5, 1) means the statement is more likely to be true. For instance, an individual x and two fuzzy sets A and B that it may belong to: $\mu_A(x) = 0.9$ stands for x that very likely to be the instance of A; $\mu_A(x) = 0.2$ stands for x is very unlikely to be the instance of B.
$\mu A \cap B(X) = \mu_A(x) \cap \mu_B(x) = min \mu_A(x), \mu_B(x) = \mu_B(x) = 0.2$
$\mu A \cup B(X) = \mu_A(x) \cup \mu_B(x) = max \mu_A(x), \mu_B(x) = \mu_A(x) = 0.9$
It means that the degree of x belonging to the intersection of A and B is the minimum of A (x) and B (x), which is 0.2; and the degree of x belonging to the union of A and B is the maximum of A (x) and B (x), seen as 0.9.

Fuzzy rule

As fuzzy logic embraces human linguistic variables, so that it captures human knowledge into fuzzy rules as a conditional statement such as IF (x is A) THEN (y is B) where x and y are linguistic variables and A, B are linguistic values. Fuzzy rules are divided into two parts as antecedent the (if part) and consequent the (then part). As compared with a classical rule-based system where, if the antecedent is true then consequent will also be true, is not fully applicable in the fuzzy rule-based system. Instead, in a fuzzy system where the antecedent is true to some degree of membership, then the consequent will also be true in some degree.

Algorithm 1 Fuzzy rule set

```
Fuzzy Rule:1
IF (Speed is Fast)
THEN stopping distance is long
Fuzzy Rule:2
IF (Speed is Slow)
THEN stopping distance is short
```

Here we noticed that, antecedent (speed) and consequent (stopping distance) are in fuzzy values ranges. Fuzzy rule-based systems give a single crisp output by aggregating all outputs fuzzy sets into a single output of fuzzy set, and then defuzzify the resulting fuzzy set into a single/crisp number. The most commonly used fuzzy inference system was invented in 1975 by Professor Ebrahim Mamdani from London University to control a steam engine and boiler Combination. Mamdani -style inference system shown in figure 2.4 are divided into four parts such as (i) fuzzification of input values (ii) rule evaluation (iii) aggregation of the rule output and (iv) defuzzification (Negnevitsky 2002).

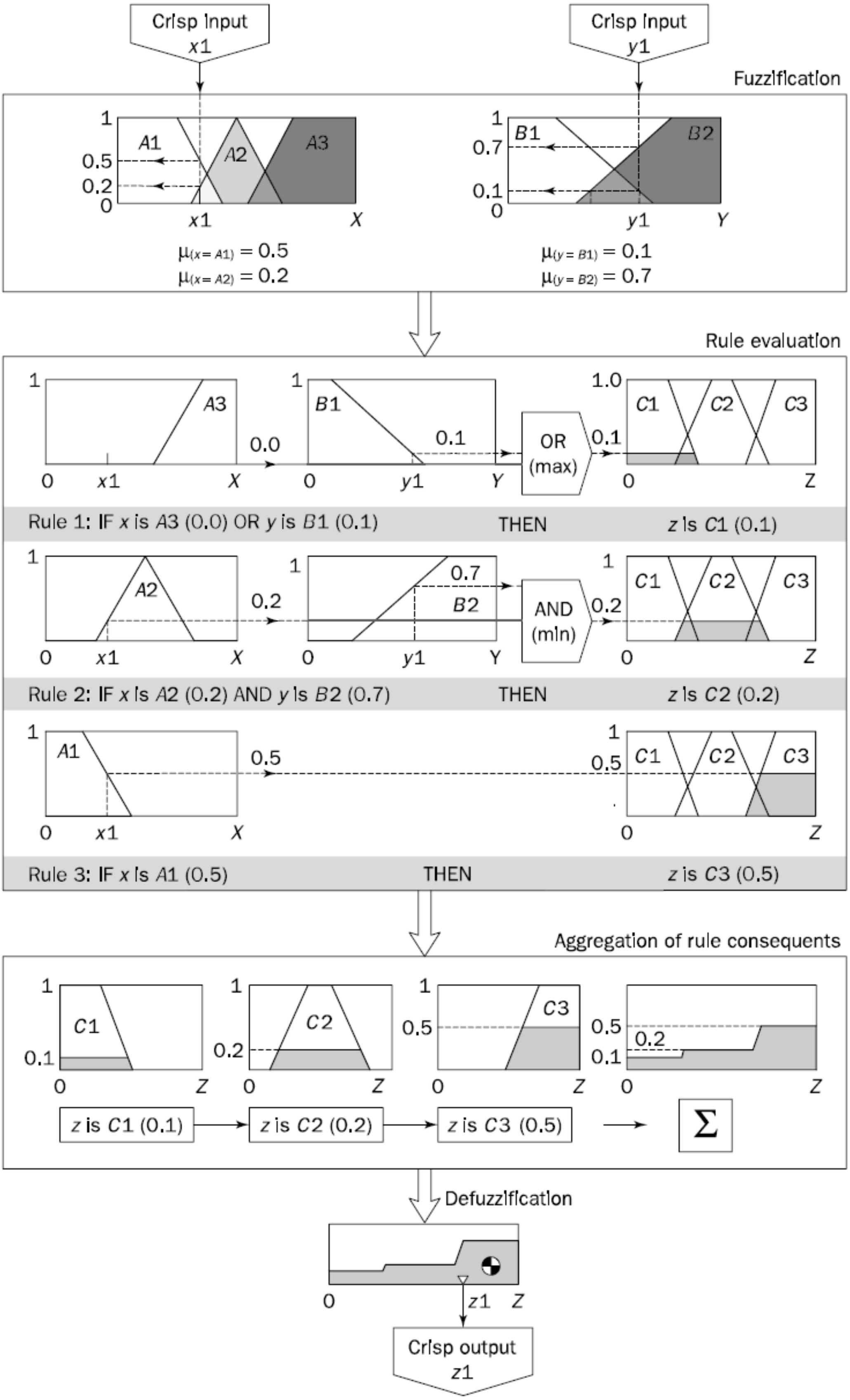


Fig. 2.4 Mamdani Fuzzy Inference System

2.5.2 Recurrent Neural Network to Forecast Patterns

A Recurrent Neural Network (RNN) is a variant of the Artificial Neural Network (ANN) where the complex connections between processing units (nodes) construct a cycle network. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Unlike non-cyclic feedforward MLP (Multilayer Perceptron) neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. The forward pass of an RNN is the same as that of a multilayer perceptron with a single hidden layer, except that activations arrive at the hidden layer from both the current external input and the hidden layer activation from the previous time-step. Consider the figure 2.5 , a length T input sequence x presented to an RNN with$I$ input units, H hidden units, and K output units. Let x_i^t be the value of input i at time t, and let a_j^t and b_j^t be respectively the network input to unit j at time t and the activation of unit j at time t. For the hidden units we have; (Graves n.d.)

$$A_h^t = \sum_{i=1}^{I} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} b_h'^{t-1} \tag{2.1}$$

Nonlinear, differentiable activation functions are then applied exactly as for an MLP:

$$B_h^t = 0_h(a_h^t) \tag{2.2}$$

The complete sequence of hidden activations can be calculated by starting at $t = 1$ and recursively applying A_h^t and B_h^t, , incrementing t at each step.

The fundamental feature of a Recurrent Neural Network (RNN) is that the network contains at least one feedback connection, so the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences, e.g., perform sequence recognition/reproduction or temporal association/prediction. RNNs are used to model Time series because the feedback mechanism creates a memory i.e. an ability to process the Time dimension. Memory is important because many Time series problems (such as Traffic modeling) need a long-term / historical modeling of Time values. Long Short-Term Memory Networks (LSTMs) is a special kind of RNN, capable of learning long-term dependencies especially because they are capable of remembering information over long time frames. The figure below summaries feed-forward neural networks.

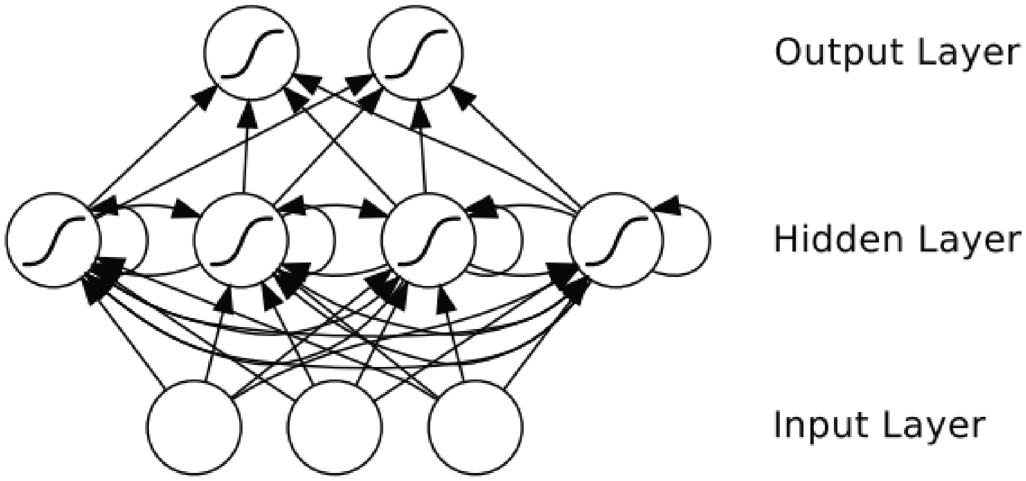


Fig. 2.5 A Recurrent Neural Network

In the Neural network system back propagation algorithm is devised to determine the derivatives of individual node weights and their iterative calculation in the network. The back propagation follows the iterative approach of adjusting weights in order to minimize error. Once training stage over, ANN saves the weight values for last time for the testing stage to get results according to the input data and conserved weights values. (Negnevitsky 2002, Wu et al. 2013)

$$\delta_h^t = \theta'(a_h^t)\left(\sum_{k=1}^{K} \delta_k^t w_{hk} + \sum_{h'=1}^{H} \delta_{h'}^{t+1} w_{hh'}\right)$$

where

$$\delta_j^t \stackrel{\text{def}}{=} \frac{\partial \mathcal{L}}{\partial a_j^t}$$

Fig. 2.6 The back propagation algorithm for Neural Network

2.5.3 Adaptive Neuro-Fuzzy Inference System

The adaptive neuro-fuzzy computing is a judicious integration of merits of the neural network and fuzzy logic approaches. This incorporates the generic advantages of an artificial neural network like massive parallelism, robustness, and learning in data-rich environments into the system. The modeling of imprecise and qualitative knowledge, as well as the formulation of uncertainty is possible through the use of fuzzy sets theory. Beside these generic advantages, the neuro-fuzzy approach also provides the corresponding application specific merits. It consists two major component namely, fuzzy inference system and adaptive neural network. In order to incorporate the capability of learning form input/output data sets in fuzzy inference system, a corresponding adaptive neural network is generated. An adaptive network is a multilayer feed-forward network consisting of nodes and directional links through which nodes are connected (Zaheeruddin and Garima 2006).

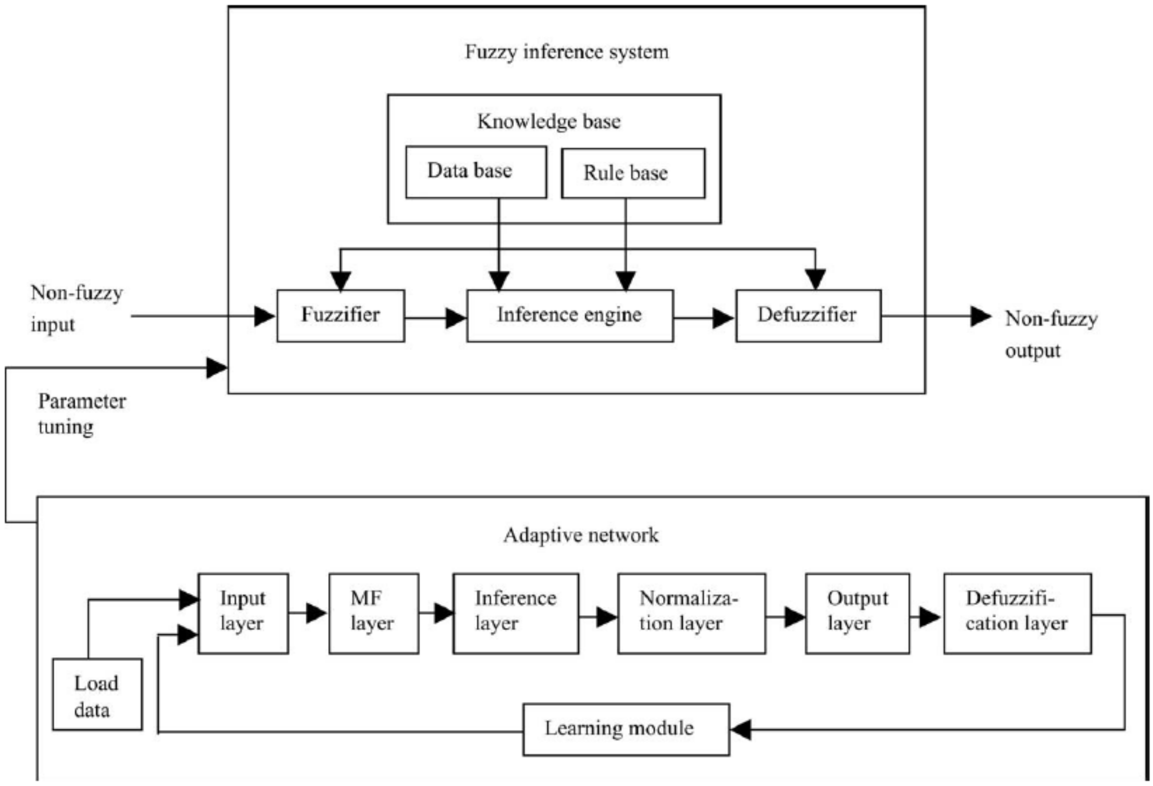


Fig. 2.7 Conceptual Diagram of ANFIS

In addition to fuzzy set theory, neural networks are used to tune the membership functions of fuzzy systems for complex systems. With the help of fuzzy sets theory, neural network can better apply back propagation algorithms to adjust the each node weight and provide deep insight to a design of better neural networks.Therefore, the neuro-fuzzy hybrid systems integrate the advantages of fuzzy systems for dealing with explicit knowledge which can be explained and well understood, subsequently neural networks are used for dealing with implicit knowledge which can be acquired by learning (Singh et al. 2012).

Using a given input/output data sets, the toolbox function ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back-propagation algorithm alone or in combination with a least squares type of method. Agreeing to the structure of the consequent parts and the inference method to compute the output of the model, rule-based model can be classified mainly in four groups such as (i) fuzzy relational models (ii) linguistic models (iii) neural network based models (iv) and TakagiSugenoKang (TSK) fuzzy models. The most commonly used systems are Mamdani-type and TakagiSugeno-type which is also known as TakagiSugenoKang-type . In Mamdani-type fuzzy inference system, both premise (if) and consequent (then) parts and ifthen rule are fuzzy propositions whereas in the TakagiSugeno-type fuzzy inference system, the premise part of a fuzzy rule is a fuzzy proposition and consequent part is a mathematical function, usually a zero- or first-degree polynomial function(Singh et al. 2012).

The node functions in the same layer are of the same function family as described such that Layer 1: every node *i* in this layer is a square node with a node function:

$$O_i^1 = \mu_A i(x) \tag{2.3}$$

Where (x) is input to node (i), is the membership grade of a fuzzy set and it specifies the degree to which the given input x satisfies the quantifier A and is Gaussian membership function :

$$\mu_A i(x) = exp[-\frac{x - c_i}{a_i})^2] \tag{2.4}$$

Where a_i and c_i are parameter sets , the parameters in this layer are referred to as premise parameters.

Layer 2: every node in this layer is a circle node label whose output is product of all incoming inputs :

$$w_i = \mu_A i(x) \times \mu_B i(x), i = 1, 2 \tag{2.5}$$

Each node output represents the firing strength of a rule.

Layer 3: The $i^t h$ node calculates the ratio of the $i^t h$ rules firing strength to the sum of all rules firing strengths:-

$$\overline{w_i} = \frac{w_i}{w_1 + w_2}, i = 1, 2 \tag{2.6}$$

Outputs of this layer will be called normalized firing strengths such that Layer 4 including adaptive nodes:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{2.7}$$

Parameters in this layer will be referred to as consequent parameters such that Layer 5: including a single labeled encircled with function of summation.

$$Overalloutput = O_i^5 = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{2.8}$$

(Singh et al. 2012)

2.5.4 Clustering for Activity Classification

The smart home system can learn about inhabitant's activity routines from the previous and current behavior of operation towards household appliances with the help of sensors data. The clustering techniques to improve the learning ability of smart home system helps to understand the inhabitant behavior. Particularly, the K-means clustering algorithms is a distance-based algorithm, which use the euclidean distance as the similarity index, namely that the closer of the two objects, the greater its similarity. The principle of K-means clustering algorithms is simple and easy to implement, with good clustering speed and effectiveness. However, Its main disadvantage is to determine in advance the value of K (k data point as the initial cluster center) in the algorithm. Whereas the human beings may perceptually make a judgment to the external environment variable, which is very strong, moderate, and very weak. Therefore, ANFIS could use the maximum, minimum and intermediates values as three initial cluster center of the K-means algorithms such as:

- In the data input, the maximum, minimum and intermediates values are selected as three initial cluster centers c_1, c_2, c_3.
- For other data object in the data set , below equation is used to cluster the Euclidean distance with all cluster centers. According to the principle of nearest neighbor, the data will be put to the nearest cluster, wherein x_j represent the j^{th}data points and $c_j$the i^{th}cluster center.

$$d_{ij} = |x_j - c_j|(i = 1, 2, 3) \tag{2.9}$$

- The equation above and below are used to re-calculate the cluster centers, where d_i represents the mean value for all the data points in the i^{th} cluster; n represents the number of data points in the i^{th} cluster; x_j represents the data point in the i^{th} cluster. The new cluster center is X_j with X_j meeting min $f(x_j)$.

$$d_i = \frac{\sum x_j}{n}(x_j \in c_i)(i = 1, 2, 3)f(x_j) = |x_j - d_i|(1 = 1, 2, 3) \tag{2.10}$$

- Steps 2 and step 3 are to be repeated, until E value in below equation is less than a given threshold and the algorithm ends.

$$E = \sum_{i=1}^{3} \sum_{x_j \in c_i} (x_j - c_j)^2 (i = 2, 3) \tag{2.11}$$

(Wanglei and Shao 2015)

In addition, Bourobou and Yoo (2015) also suggested that in order to detect repeated patterns or anomalous user behavior from varied and complex user activities, K-means clustering algorithm shows the best performance in terms of the temporal complexity and cluster set flexibility even for the very large amount of data in the IoT smart home environment. On the other hand, the second step describes the training of smart environments for predicting and recognizing user activities inside his/her personal space in order to mitigate the issues related to that activity recognition in the real world. The Allens temporal relations based Artificial Neural Network (ANN) gives the highest accuracy for user activity recognition. The additional use of an efficient feature selection approach of J48 decision tree, improves both the average accuracy and the run-time performance. This hybrid method approach is effective with the activity recognition challenges, considering the restrictions and features of the IoT based smart home environment.

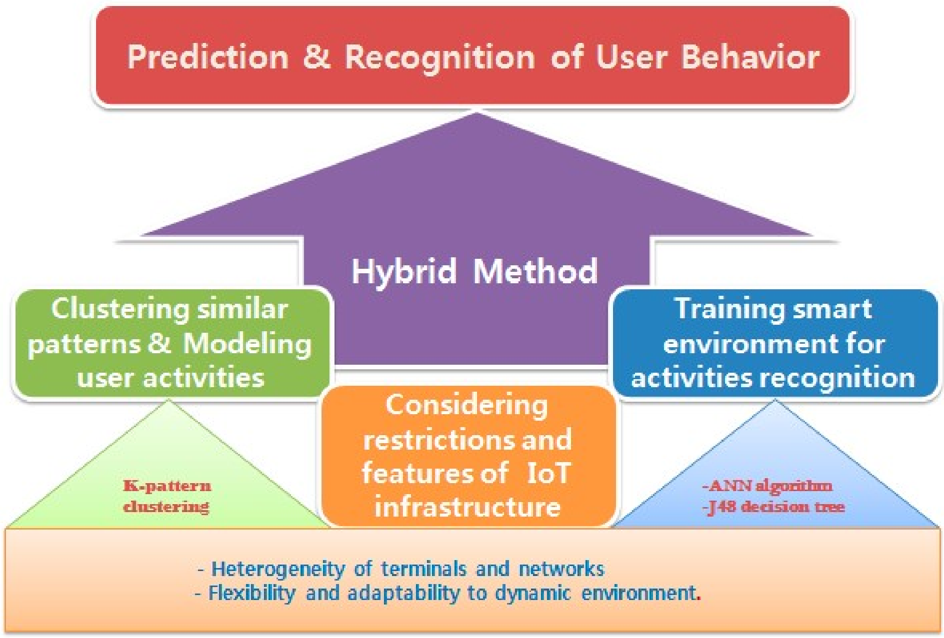


Fig. 2.8 The IoT Architecture of Hybrid methods (Bourobou and Yoo 2015)

In most cases, the existing clustering algorithms have some ambiguity in processing noisy data. Indeed, this noise makes it difficult to include an object into a certain cluster because it affects the results of the algorithms. In contrast, the K-means clustering algorithm has the ability to overcome this drawback. On the other hand, some works integrate user behavior through activity recognition. Detecting user activities usually implies the collection of observation sequence in order to recognize new events. Also, Bourobou and Yoo (2015) explained the pseudo code for methodology analysis is described in the following figure 2.9. The set of frequent activity patterns and the number of clusters are the input, while the set of clusters is the output of the algorithm as shown below figures.

```
Algorithm 1: K-PATTERN CLUSTERING (NC, cP)
---------------------------------------------------------------
Input: NC- Number of Clusters        - Initially is zero
       C1set- set of Cluster Centers
       P1set- set of Input Patterns
Output: Set of Clusters
Read the Input dataset
begin
    for each Pattern P in dataset do
        if NC= 0 then
            C1 ← P1                  - First Pattern as Cluster Center
              i ← 1                  - Index of the Pattern
            NC ← 1
        else
              i ← i+1
            Get next Pattern Pi
            Assign Pi to Cluster
            Cluster (NC, Ci, Pi)
return Cluster
```

Fig. 2.9 Process of Forming Frequent Activity Patterns

2.5.5 Naive-Bayes Classifier

The Naive Bayes is the simplest probabilistic model in generative modeling approach. The Naive Bayes model assumes all data points are independently and identically distributed, that is it does not take into account any temporal relations between data points. The model factorizes the joint probability over the data points as follows :

$$p(Y_{1:T}, X_{1:T}) = \prod_{t=1}^{T} p(x_t^{\rightarrow}|y_y)p(y_t). \quad (2.12)$$

The term $p(y_t)$ is the prior probability over an activity, that is how probable an activity is to occur without taking any observation into account. The observation distribution $p(\vec{x_t}|y_t)$ represents the probability that the activity y_t would generate observation vector $\vec{x_t}$.If we were to model the distribution of the observation vector exactly, we would need to consider all possible combinations of values in the vectors dimensions. This would require 2_N parameters per activity, with N being the number of sensors used. This easily results in a large number of parameters, even for small numbers of N. Instead, we apply the naive Bayes assumption, which means we model each sensor reading separately, requiring only N parameters for each activity. The observation distribution, therefore, factorizes as

$$p(\vec{x_t}|y_t = i) = \prod_{n=1}^{N} p(x_t^n|y_t = i) \tag{2.13}$$

Where each sensor observation is modeled as an independent Bernoulli distribution. Using naive Bayes is a very strong model assumption, which most likely does not represent the true distribution of the data (i.e. it is very likely that two sensors are dependent on each other with respect to a particular activity. However, naive Bayes has been shown to give very good results in many domains, despite its overly simplistic assumption.
(Rish 2001, Khalil and Linz n.d.)

2.5.6 Hidden Markov Model for Discreet Activities Pattern Recognition

To begin with, Chen et al. (2016) suggested that, users contextual information has two parts, first user own activities info and second related context information, by combining both of them, a knowledge-base can be build to infer, understand situation and intentions of users. Likewise in the smart home scenario, it can be correlated with human activity recognition process such as first sensor data is collected in time series manner and secondly probability of activity transition from one state to another can be estimated. However, it has been seen that human does not change daily activities habits easily in a continuous manner, at-least certain period of times due to other environmental factors. For instance, students on top floor go out of the classroom, will have a higher probability to walk downstairs while not walk upstairs after class. Under such circumstances, Hidden Markov model can fit into this scenario to predict human activities in more accurate manner. Let $Q = q_1, q_2, q_3, , q_N$, the set of all possible activity-states and $V = v_1, v_2, v_3, , v_M$ is the set of all possible event -observations, therefore a Hidden Markov Model(HMM) can be defined as 3-triple such that

$\lambda = (A, B, \pi)$ where $A = [a_{ij}]$ represent the state transition probabilities from i^{th} toj^{th} state

$B = b_j(k)$ represents the event-observation probability distribution.

$b_j(k) = P(o_t = v_k | i_t = q_j)$ is the probability of generation observation v_k in the activity-state of q_j and at time t. Where, π represents the initial state distribution.

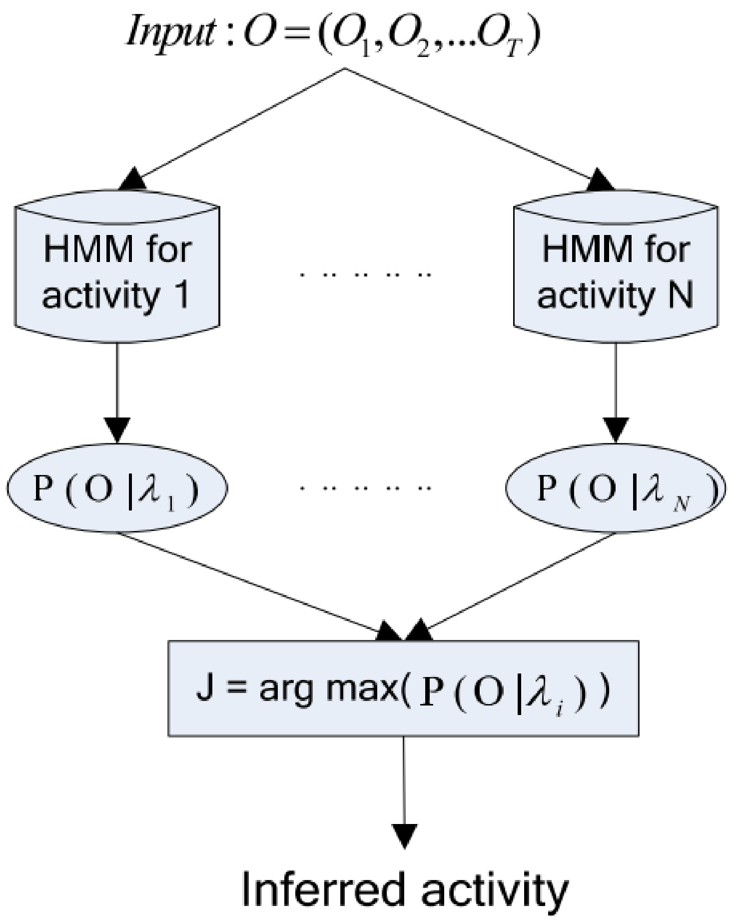


Fig. 2.10 Activity Pattern Recognition Process in Hidden Markov Model

In the real time-series activity sequences, it is an interesting factor to process sequential data to show a dynamic behavior of an object such as analysis of DNA genes, protein modeling, gesture recognition and behavior analysis. etc. As a result, in supervised machine learning methods, the classification problem of such temporal data can be mathematically tacked by Hidden Markov model(HMM) with more efficient manner Lotte and Congedo (2007). The discrete time-series activity dataset provides the input parameter to feed into HMM for training purpose to forecast most likelihood activity pattern in the system. The Hidden Markov Model (HMM), work on Markovian assumption such that the temporal correlation between consecutive time-slices relies on two independent assumptions. The first one is, hidden activity state at time t, namely (Y_t) depends only on previous hidden activity state (y_{t-1}). The other one is, the observable activity state at time t, namely $X_t^{\rightarrow}$, depends only on the hidden state activity Y_t at the time slice.
Therefore, HMM can be mathematically described by three parameters: the initial state (y_1), transition distribution $p(y_k|y_{k-1})$, and emission probability $p(x_k|y_k)$, then the joint distribution of the variables can be formulated as follow :

$$p(x, y) = p(y_1)p(x_1|y_1)\prod_{k=2}^{K} p(y_k|y_{k-1})p(x_k|y_k) \tag{2.14}$$

Parameters Estimation and Expectation-Maximization with Baum-welch algorithm
The parameter estimation and standard training of HMM is obtained by the EM (Expectation Maximization) algorithm known as ”Baum-Welch” that is more efficient and used for many other probabilistic models, such as the Finite Mixture Models (FMM) and the Gaussian Mixture Models (GMM) etc.. A HMM has two stochastic processes where observations are maintained by multiple distributions with switching between the states. EM computes HMM parameters by maximizing the log-likelihood $(L = logP(O|))$ of the observed time-series event data sets (O) such that; $ML = argmaxlogP(O|)$.
(Huda et al. 2014)

Decoding Likelihood ADL Patterns

The Viterbi Algorithm is commonly used for decoding as the likelihood sequences can be identified with Viterbi algorithm. The algorithm is named after Andrew Viterbi, the application of algorithms is to maximize problem involving probability. In other words, Viterbi algorithms can be well understood as a max-product algorithm, whose primary function is to find the most likely subset of latent variable among a large number of datasets. The application of Viterbi algorithms is well embraced in all graphical model such as Bayesian network, hidden Markov model and conditional random fields to find the best match with higher maximum probability subsequences of observations in the large dataset.

$$q = argmaxlog(P(q|O, \lambda)) \tag{2.15}$$

2.5.7 Spatial and Temporal Reasoning

The supervised and unsupervised machine learning approach for activity recognition and prediction in IoT environment have enormous scope to make living environment better such as improvement in elderly health monitoring, intrusion detection, optimizing resources for energy savings and providing an optimized assisted living to inhabitants. In particular multidimensional time series data provide the ground for activity classification and pattern recognition algorithms by Trabelsi et al. (2013). However, Cook et al. (2009) argued that a very little can be done in smart home environment without an explicit or implicit reference to where and when the meaningful events occurred. For an intelligent system to make sensible decisions it has to be aware of where the users are and have been for some period of time. Spatial and temporal data reasoning can very useful to analyze trajectories of people within a specific environment and understand their situations and intentions for a specific set of activities. However, few use case scenarios can describe the spatial-temporal relationship in more explanatory manners such as the following;

Prevent hazardous situation at home

(I) The Cook et al. (2009) suggested that, consider a scenario in which the AMI environment sense the cooker has been left turned on and a sequence of sensor signals (e.g., movement sensors combined with RFID sensors) detected the location of the user moving from the kitchen to a living area and then finally going to the bedroom. Afterwards, the bed occupancy sensor (a pressure pad) detects the person is lying on bed and fallen asleep. In such situation IoT system could raise an alarm if the occupancy in bedroom and the kitchen happen at the same time more than 10 minutes. As a result, the cooker unattended situation can be triggered through some rule based system as an alert to prevent hazardousness in case of fire. (Cook et al. 2009)

(II) Lets consider another situation in which the doorbell has been rung and the resident does not respond within 5 min. However, the AMI system detects that the person is at home and knows the resident is not hearing impaired. This can be identified as a potential emergency and may trigger a procedure where caregivers are notified and will try to contact the individual visually or by telephone.

An alternative formalism for reasoning about time is based on Allen's temporal logic. Allen suggested that it is more common to describe scenarios by time intervals than by time points, and defined thirteen relations that comprise a temporal logic: before, after, meets, meet-by, overlaps, overlapped-by, starts, started-by, finishes, finished-by, during, contains, and equals. These temporal relations play a beneficial role in prediction and anomaly detection for ambient environments.
(Cook et al. 2009)

Furthermore, Nazerfard et al. (2010) proposed a temporal model for discovering temporal feature and relation of activity patterns from sensor data called, DTFRA (Discovering of Temporal Features and Relation of Activities) which discovers the usual start times of each activity patterns in form of a normal mixture distribution. Using a k-means clustering technique. DTFRA discovers a similar representation for the duration of activities using a normal mixture distribution. The temporal information that is discovered by algorithms can be beneficial in many different applications such as reminder systems, anomaly detection, context-aware system networks. The activity patterns in the smart environment also include a timestamp. The timestamp indicates when a particular activity has occurred, or more specifically when the specific sensor was triggered. Just like the association rule mining, adding the concept of temporal features to the activity patterns can be quite useful, and in some cases it is very important to be implemented.

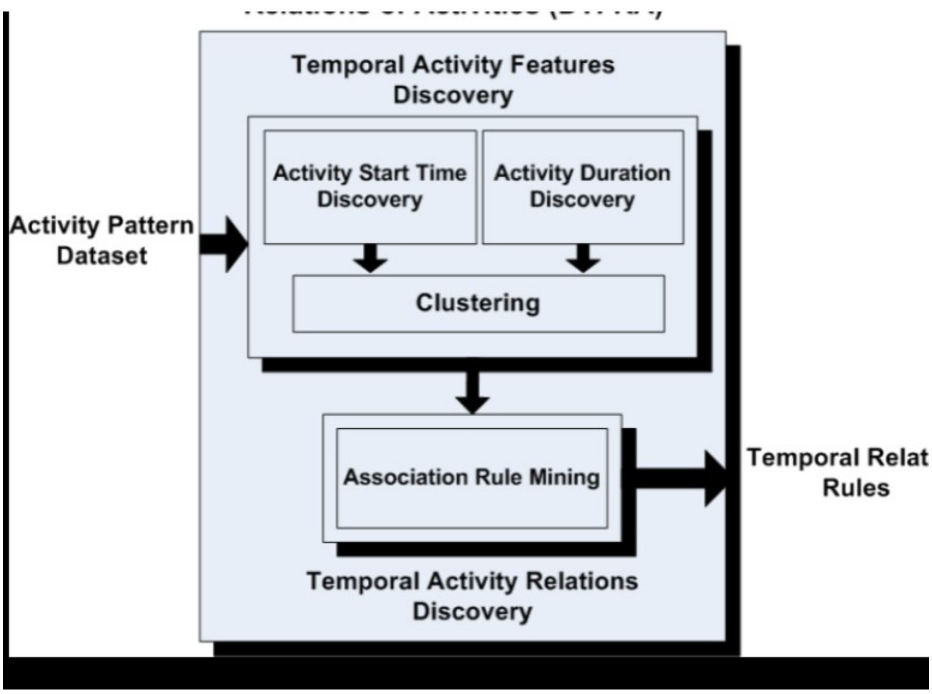


Fig. 2.11 DTFRA Architecture (Nazerfard et al. 2010)

Temporal Activity Feature Discovery

The Nazerfard et al. (2010) suggested to consider two temporal features for every activity such as (i) the start time of the activity and (ii) the duration of the activity. In their proposed work, the start times of every activity instance are clustered together in order to obtain a canonical representation for a specific set of activity.

Cluster #	Start Time (hh:mm)	Duration (hh:mm)
1	[8:37 – 10:29]	[0:02 – 0:04]
2	[6:32 – 9:12]	[0:06 – 0:08]
3	[18:56 – 22:18]	[0:02 – 0:06]

Fig. 2.12 Discovering Start Time and Duration of Activity

Here, K-Means clustering algorithm can be applied to construct a mixture model for each activity ai . If it denoted the start time of an activity instance a_i as t_i , then the probability that t_i belongs to a certain cluster k with parameters $\Theta_k = (\mu, \sigma)$, Which can be expressed as a normal Probability Density Function :

$$prob(t_i|\Theta_k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(t_i-\mu)^2}{2\sigma^2}} \tag{2.16}$$

Temporal Activity Relations Discovery

In addition, Nazerfard et al. (2010) argued that after the canonical forms of start time and duration have been discovered, the next task could be to discover the order of activities. The input to this stage is the features discovered in the previous stage such as the canonical start time and duration. The output of this stage is a set of temporal relations between activities. The temporal relations will determine the order of activities with respect to their start time such as for a specific time point, what activity would be the most probable. Such results can be useful in a variety of activity prediction scenarios, such as home automation.

Cluster #	Start Time	Duration	Next Activity	Conf
1	[8:37 – 10:29]	[0:02 – 0:04]	Eating	0.90
2	[6:32 – 9:12]	[0:06 – 0:08]	Eating	0.81
			Meal Preparation	0.11
3	[18:56 – 22:18]	[0:02 – 0:06]	Personal Hygiene	0.50
			Sleeping Not in Bed	0.25
			Eating	0.25

Fig. 2.13 Discovering Temporal Relation of Activity

2.6 Urban Computing: IoT projects in fast developing nations

Urban computing is an interdisciplinary field where computer sciences meet the conventional city related fields, like transportation, civil engineering, environment, economy, ecology and sociology in the context of urban spaces. The tasks of urban computing include improving urban planning, easing traffic congestion, reducing energy consumption, and reducing air pollution. Different tasks can be fulfilled by combining different IoT sensors data sources with different data acquisition, management and analytics techniques on different layers of the framework. In specific problem scenario, Sun et al. (2014) suggested that traffic congestion on citys road has impacted on the city development seriously, and became the crux which constrained the city development. In such circumstance, an IoT- intelligent urban traffic management system is very much required to tackle such increasing problem on urgent basis. As a result, an IoT based intelligent collaborative urban traffic management system is proposed by them to solve the current traffic movement problems. Their architecture is integrated with perception layer, network layer, and application layer, where network layer contains data processing and intelligent computing was performed on cloud computing platform. The application layer is packaged into service-based solutions, such that intelligent collaboration of urban traffic is tackled by intelligent service request and response dispatching strategy. (Sun et al. 2014)

However, Mahdavinejad et al. (2017) argued that, an important aspect of urban planning is to draw long-term solutions for better living experiences for inhabitants rather than short-term solutions. Although, taking a decision on such massive population level is quite critical but IoT provide a platform through smart city data analysis, to predict local authority about the potential problems in order to find the right solution. As the data is being generated in a continuous manner from heterogeneous resources such as traffic, health, water system, energy footprints etc.., the improvement in infrastructure service would be highly benefited by combining IoT technologies with urban planning. Due to nature of heterogeneous sources, the collected data from devices play a crucial role to ensure its quality includes factor such as; data precision, noise, and discreet observation data sets.

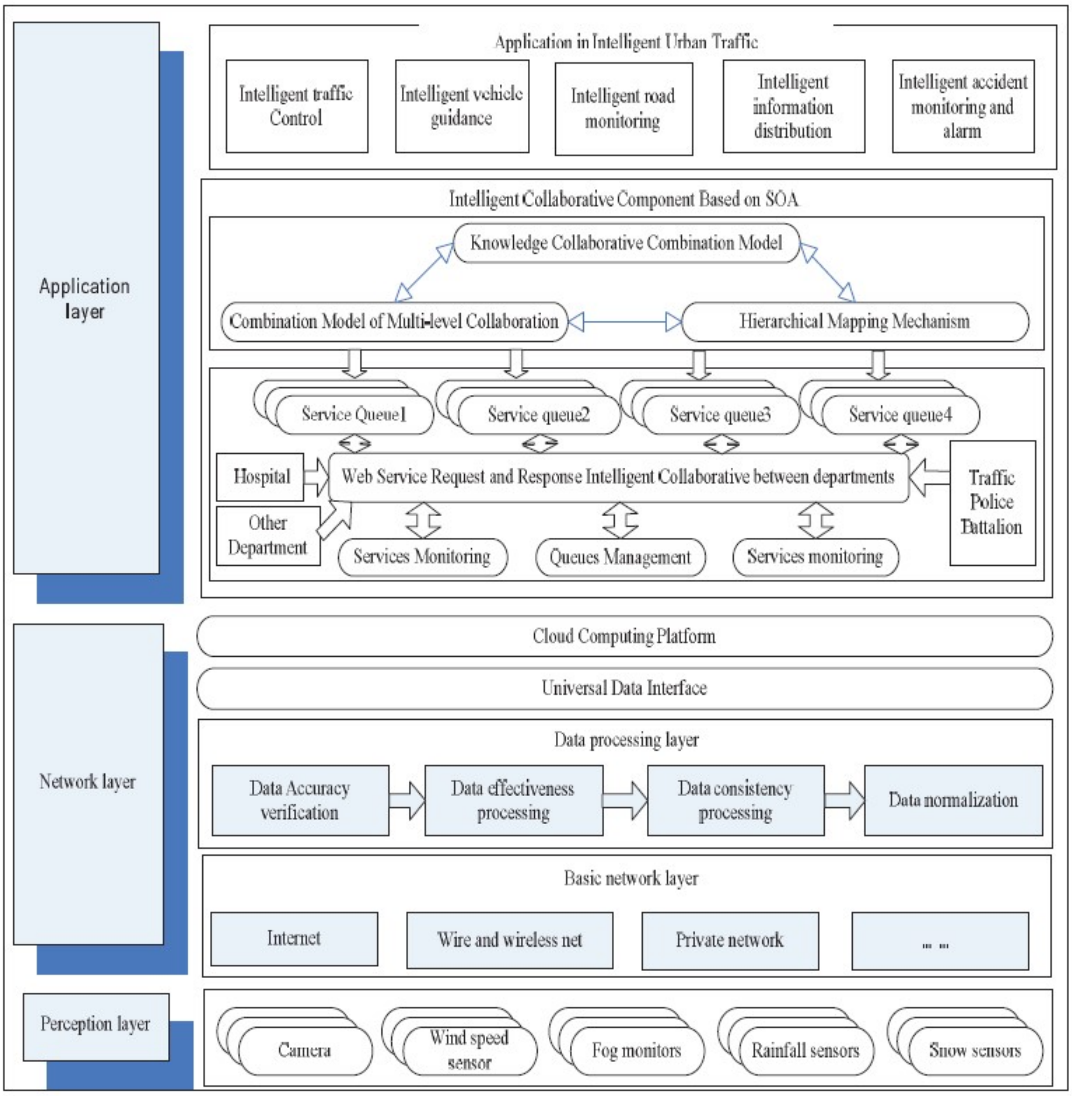


Fig. 2.14 Structure of the Intelligent Collaborative Urban Traffic Management System Sun et al. (2014)

In the approach of the smart cite mission, the object is to promote cities that provide core infrastructure and give a decent quality of life to its citizens. In order to achieve, a clean and sustainable environment and application of smart solutions. The core infrastructure element in a smart city would include, adequate water supply, assured electricity supply, sanitation, including solid waste management, efficient urban mobility and public transport, affordable housing, robust IT connectivity and digitalization with enhanced health care systems for inhabitants.

Fig. 2.15 Smart Solution Aspects for Smart City in India

As a result, among the all developing nations, India has been an active member into research and development of Urban Planning with modern IoT technologies. The prime minister, Shri Narendra Modi, strongly believes that for the first time in the country, people and urban leadership would play the pivotal role in deciding the future course of their cities. He was speaking at the launch of three major urban development initiatives: AMRUT (Atal Mission for Rejuvenation and Urban Transformation), Smart Cities Mission and Housing for All (Urban), at Vigyan Bhawan in the capital city of New Delhi. The Prime Minister explained that for the first time in India, a challenge was being floated, in which the citizens of urban India could contribute in the formulation of development visions of their cities. Those cities which were able to competitively meet the required parameters would be developed as smart cities. This competitive mechanism would end the top-down approach, and lead to people-centric urban development.
$Pmindia(2016)$

On the other hand, Makeinindia (2016) suggested that the Prime Minister Narendra Modis vision's Digital India, has set an ambitious plan to build 100 smart cities across the country. In particular, PM Modi in his speech quoted, Cities in the past were built on riverbanks and now are built along highways but in the future, it will be built based on the availability of optical fiber networks and next-generation infrastructure. The Government of India allocated INR 70.6 billion (US dollar 1.2 billion) for Smart Cities in Budget 201415. India has also been inviting foreign partnership in developing the smart cities and has signed deals to build eight smart cities as per three with Germany, three with the US, and one each with Spain and Singapore.

Fig. 2.16 Smart City Projects based on the State and Central Government Funding

2.7 State-of-Art IoT (Internet of Things) Projects

However, many smart home projects has been researched in the last decade, especially under context-aware assisted living research theme where heterogeneous sensors and devices generate a large amount of contextual information for remote monitoring and using rule-based reasoning, probabilistic model, data mining, sensor network based decision support system and video monitoring etc. In particular, utilizing long-term situational data has not been well embraced for modeling user activity state to predict future activity instances Forkan et al. (2015). Also in the work of (Nazerfard et al. 2010) and (Fahad et al.

2014), effective ambient intelligent systems are trained on observed activities sequence through embedded sensors which are capturing correct realization of activity state but missing the uniqueness of activity classification approach. Although, some significant work have been proposed for ambient intelligent space such as Care-lab, CASAS, Grator-Tech HIS, Aware home, iDorm and MavHome projects etc, where the entire process of activity classification is subdivided into four different subprocesses such as (i)sensing, (ii)data-pre-processing, (iii)data modeling for feature extraction and(iv) activity clustering by Liu (n.d.) . In addition, IBM Watson research center working towards, to develop a personalized IoT environment for inhabitants, identifying their preferences, activity patterns in order to provide them a customized digital assistant service to operate and control various appliance with certain amount of automated tasks (IBM 2017).

2.7.1 Panasonic Smart Home

The smart hub sits at the heart of every Panasonic smart home system, connected through wireless to all other smart home devices. It can be setup through one push pairing, and communicate with other device as far as 300m away. It also can record to microSD/MicroSDHC card any footage it receives from triggered smart home cameras. The camera, sensors and smart plugs will be able to communicate with each other via the smart home hub at the distance. The Panasonic smart home devices communicate with each other using a DECT Ultra low energy (ULE) wireless standard thats invisible to regular consumer products rather than Wi-Fi-based products.
(Panasonic, 2016)

However, Panasonic just only offer a home monitoring and control system. It gives the ability to monitor and control home from the convenience of smart phone or tablet. Therefore, system needs instruction from user in order to perform a task. As a result, the system is re active in nature and could not predict any activity accident in a pro-active manner. Despite being a simple installation and easy operation, panasonic smart home system missing the intelligence part in the system, which can perceive and predict the activity instance in the environment.

2.7.2 IBM Watson Internet of Things

In the IoT research, IBM Watson represents the leading edge of cognitive computing platform, using natural language to communicate with people. The Watson based smart home become intelligent by applying cognitive computing to connected home IoT data to uncover insights to enhance inhabitant convenience. The IBM Watson, expanded the cognitive IoT possibilities to infuse intelligence into the physical world to enhance the IoT based assisted living experiences. The traditional programmable computing systems are designed to handle specific scenarios and data sets while IoT data doesnt play by traditional rules, instead heterogeneous data sources such as images, videos, sounds and machine data of many types are combined with social media, weather and enterprise data in order to provide contextual information and relevance that sharpens insights.
(IBM 2017)

However, being a market leader for Cognitive IoT platform, IBM Watson provide intelligent solutions for AI based assistant such as connected home appliances and autonomous self driven vehicle. Our research also follow the same approach for improving smart home inhabitant experience through connected appliances but like IBM Watson we do not include the cloud computing paradigm into the system. On the other side, IBM Watson have their propitiatory(paid subscription) IoT cloud framework to communicate with other expensive home appliance brands such as Whirlpool, Nest etc.. Instead our research aim is propose and develop open source low cost IoT solutions which can enhance inhabitant experience with cost effectiveness.

2.7.3 i-Dorm (Intelligent Dormitory)

The IIE research group at the University of Essex researched on new ways for mankind to interface with the technology that surrounds them. The i-Dorm (intelligent-dormitory) represents an application of the grid architecture, whereby the actuators and sensors found inside represent different service within the room. The intelligent dormitory was an office in the computer science department at the University of Essex. It has been converted into a student dormitory, the design of which is based on the campus accommodation at Essex. iDorm uses three main communication protocols to allow its device to communicate with each other to show its networks independent such as LonTalk, 1-Wire, and IPv4. The LonTalk is a twisted pair network, similar to IP that comes in two versions-one that provides power to the device through the network and another that requires the device to have an external power supply. The network is laid out in a ring arrangement; there is no central server. While the 1-Wire is designed for small-scale applications where the distances between devices on the network are relatively small. Unlike LonTalk, this network required a central server that takes the form of a small piece of hardware called a Tiny Internet Interface board(TINI). The TINI board acts a gateway to the 1-Wire network. It is addressable on the IP as well as on the 1-Wire network. A Java Virtual Machine is embedded on the board and the research group has written a small server that accepts HTTP requests. The protocol that has been produced is an XML definition for the iDorm. All information requested form the iDorm must go through a central server. The server communicates with the iDorms LonTalk and 1-Wire network across IP using HTTP request to get information. The interfaces can request information about the iDorm in the form of an empty XML tag pair. This pair then gets returned to the interface with the blank part of the request filed with the appropriate data. The XML definition is stored on the central server and can be retried by any interface. (Holmes et al. 2002, Hagras et al. 2004)

However, iDorm gateway server work on Http server request to parse the sensor information into XML format. In the real time system, it could lead to security issues and late response time to the IoT controllers. As a result, the critical role of embedded-agent mechanisms are missing in the iDorm gateway server.

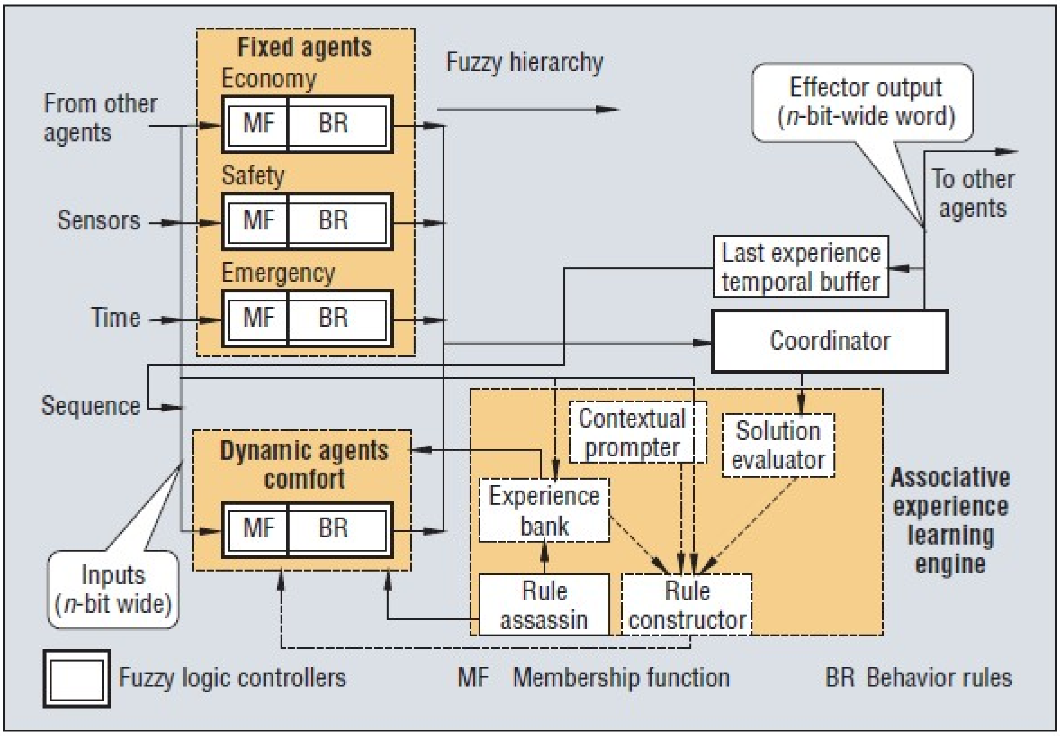


Fig. 2.17 Incremental Synchronous Learning architecture in the iDorm Embedded Agent

2.7.4 DOMUS

Subsequently, the DOMUS laboratory includes a standard apartment such as kitchen, living room, dining room, bedroom and bathroom located within the computer science department of the University of Sherbrooke and equipped with a set of infrared sensors, pressure detectors, smart light switches, electrical contacts on doors, audio and video systems, as well as smart tags (RFID) to obtain, in the apartment, the real-time position of the user and objects. The Autonomic Pervasive Computing can simplify the complexity of the configuration, maintenance, and management of pervasive environments such as smart spaces. Therefore Gouin-Vallerand et al. (2008) made an effort for the Autonomic Pervasive Computing approach and attempted to build an autonomic pervasive space at DOMUS Laboratory. (Gouin-Vallerand et al. 2010)

However, the implementation of a intelligent middle-ware architecture was missing for the autonomic pervasive computing system. Which could have been used for reducing the complexity in the pervasive spaces in pro active manner and also reduce the amount of needed human interventions.

2.7.5 CASAS

The CASAS2 Smart Home project is a multi-disciplinary research project at Washington State University. The CASAS project is focused on the creation of an intelligent home environment aiming at minimizing the cost of maintaining the home and maximizing the comfort of its inhabitants. The testbed smart apartment which is part of the CASAS smart home project is a three bedroom apartment that includes three bedrooms, one bathroom, a kitchen, and a living/dining room. The testbed apartment is equipped with different kinds of sensors and actuators to sense the environment and give back information to inhabitants accordingly Manuscript and Dysfunction (2015). However, the scope of the data collection to include a greater diversity of resident demographics and to perform longitudinal studies were missing in the architecture and subsequently home automation strategies that provide safe and energy-efficient support of resident daily activities could be investigated further. (Manuscript and Dysfunction 2015)

However, CASAS provide out of the box services with no customization or training mechanism. Although, the most important part was missing in the experiments is, population wide analysis of resident behavior using the sensor data. It could help to understand inhabitant's psychology about overall daily routines. The population wide analysis could provide the overall cognitive intentions of inhabitant in smart home environment.

2.7.6 Virtual Personal Assistants (Siri, Cortana, Alexa, G-home)

The virtual personal assistants or digital assitant, works as an cloud based application program that understand natural language(voice commands) to complete the task in reactive nature. Nowadays, spoken dialogue systems seem to be intelligent agents that are able to help users to finish tasks more efficiently via spoken interactions. There are many techniques used to design the VPAs, based on the application and its complexity.In particular, Google has improved the Google Assistant by using the Deep Neural Networks (DNN)

method which highlights the main components of dialogue systems and new deep learning architectures used for these components. Subsequently, Microsoft used the Microsoft Azure Machine Learning Studio with other Azure components to improve the Cortana dialogue system.
(Këpuska 2018)

However, the focus of cognitive IoT research is to produce a pro-active system whereas spoken dialogue is not appropriate due to re-active in nature. The user's direct interaction with IoT devices is less or minimally required in the ambient-intelligent system to ensure the un-obstructiveness and pro-active task executions.

2.7.7 MavHome

The agent based hierarchical architecture of MavHome smart home project focuses on the creation of an environment that acts as an intelligent agent, perceiving the state of home through sensors and acting upon the environment through device controllers. The agents goal is to maximize comfort and productivity of its inhabitants and minimizes operation cost. In order to achieve this goals, the house must be able to predict, reason about, and adapt to its inhabitants. The MavHome architecture has been implemented using a CORBA interface between software components and powerline control for most electric devices. In MavHome, prediction Using Sequence Matching work as the SHIP(Smart home inhabitant prediction) algorithm matches the most recent sequence of events with sequences in collected histories. (Das et al. 2002)

However, A key disadvantage is the fact that the entire action history must be stored and processed off line, which is not practical for large prediction tasks over a long period of time. A dynamic data handling approach is missing in the MavHome architecture. The periodic update in data pool for new datasets are very important to train system more effectively.

2.8 Summary

The CIoT (Cognitive Internet-of-Things) targets the next level of IoT systems to capture ambient parameters from the environment in such a way that it should not intrude inhabitant's consciousness unless it is required implicitly. The collected data from various ambient parameters in the environment provides the platform for the analytical models to analysis inhabitant's ADL patterns with help of machine learning algorithms to identify and forecast the most likely actions/ routines in their daily lives. However, in above mentioned research work applied the various method to produce state-of-art context-aware IoT smart home systems. But for those models applied data-driven machine learning approach have the larger impact and contribution to building modern ambient intelligent systems compared to the knowledge-driven model based who works on traditional ontologies framework.

Although, the role of ambient intelligence in IoT system is not only to observe the inhabitant activity states, but also take proactive actions and decisions to adapt the environment in order to enhance the independent living experience for young inhabitants in metro cities and elderly people in society. The ideal IoT system should anticipate the inhabitant desires and environmental factors without any direct interferences to a certain extent. The data-driven approach perform better compared to knowledge-driven approach in supervised machine learning applications. Although the major challenges to perform inhabitant activity identification form heterogeneous IoT sensor is a big problem. Due to the diversity in raw data format and impossible to use them directly in the machine learning models. The predictive analysis of the existing research work of ambient intelligent environment is not only limited to smart homes but also applies to smart cities, health care systems, urban planning, automated resource management systems. At present, the universal design approach to handle the gap between heterogeneous time series data from IoT sensors for predictive analysis is lacking in the existing research work. As a result, to fulfill the gap between IoT systems and ambient intelligence capabilities key areas such as unobtrusiveness, personalization, adaptiveness, and anticipation are much needed to be further explored and researched in order to come with new enhanced versions of modern ambient intelligent IoT systems for the generations.

Moreover, the state of art projects give an attempt to integrated HCI to AmI environment. The most important intelligence part is missing in their state of art projects . They merely addressed the generic issues stemming from the nature of ambient intelligence (AmI) research which relate to HCI practice. The current issue in AmI environment is to bridge the IoT and machine learning capabilities into the system. The current practice and paradigms must embrace the machine learning capabilities and awareness in the community of AmI solutions by providing a comprehensive IoT solution with ambient intelligence capabilities that can address the true meaning of Cognitive ambient intelligent systems.

In conclusion, the new generation ambient intelligent IoT systems would be focusing on, i)The collective activities performed in the environment, rather than individual tasks ii), focusing on the relation between sequentially performed activities, iii) No value would be given to starting and ending points of individual tasks rather than whole activity set, iv) emphasize that multiple activities are loosely connected in terms of achieving specific goals, v)The multiple perspectives could imply the reuse of information in terms of rules sets for different task executions.

CHAPTER 3

Ambient Cognition Model (ACM): a dynamic model for activity-intention identification

Overview

This chapter fulfills the first research objective to propose a novel Ambient Cognition Model (ACM) for the identification of inhabitant's intention and situation in the cognitive IoT (Internet of Things) environment. The definition of activity type at the micro and macro level provided with a clear methodology and vision for further data analytics. The IoT sensor's data streams provide real-time data readings of inhabitant activities in the environment. A data scrutiny approach with an appropriate statistical model facilitates to extract the contextual knowledge from the raw-data streams. The ACM provides a dynamic model, extracting contextual knowledge about the inhabitant's activity intentions and labeling them as discrete activity sets for further machine learning predictive analysis.

3.1 Ambient-Cognition Model (ACM)

The prime objective of Ambient Cognition Model (ACM) is to identify and label inhabitant's intentional activity states through numerous embedded IoT(Internet of Things) sensors in the Cognitive IoT environment. The architecture of ACM as shown in figure 3.1 provides a novel activity identification technique for ambient-intelligent smart space. The ACM collects time-series data log from various embedded IoT sensors in raw format and preprocess them for noise filtering to smoothen data consistency and the redundancy. Afterward, setting up threshold (th_i) range and weight (W_i)scheme on the individual sensor for SA (Spot Activity) observations facilitates the activity identification process. As a result, the SA states emerge as an absolute state in 0 or 1 (ON/OFF) format. The most challenging task in cognitive IoT systems is to extract the contextual information about the surrounding situation from individual sensors to infer the activity intentions of inhabitants. Especially, multiple embedded sensors (Wearable/Non-Wearable) generate events data log in analog and digital formats such as luminosity, motion, pressure, noise, the ultra-sonic distance of object etc. As a result, the accelerated sensors data from heterogeneous sources creates a problem of data normalizations to infer activity state accurately. The historical knowledge base and understanding about the surrounding spaces are utilized to set threshold(th_i) values for various sensor's activity datasets. The threshold (th_i), provides the filtration technique to normalize the accelerated raw data from sensors into an absolute state of activities. The diversity in sensor's accelerated data results in a key problem to estimate the activity states, therefore (th_i) provides a solution to the existing problem. As a result, the collective information from individual sensor forms a contextual outcome of an intentional activity state. We applied the weight scheme to individual sensors, based on their placement and impact factor in the cognitive IoT environment. The current states on inhabitant situation in the environment extracted through the ACM equation 3.8 along with their defined weights. As a result ACM, provides a final outcome of inhabitant current situation as $sleeping(CS_s)$, $working(CS_w)$, $cooking(CS_c)$ etc.

$$\Delta IA_i = \sum_{i=1}^{n}(SA_i \times W_j) + LW_k \tag{3.1}$$

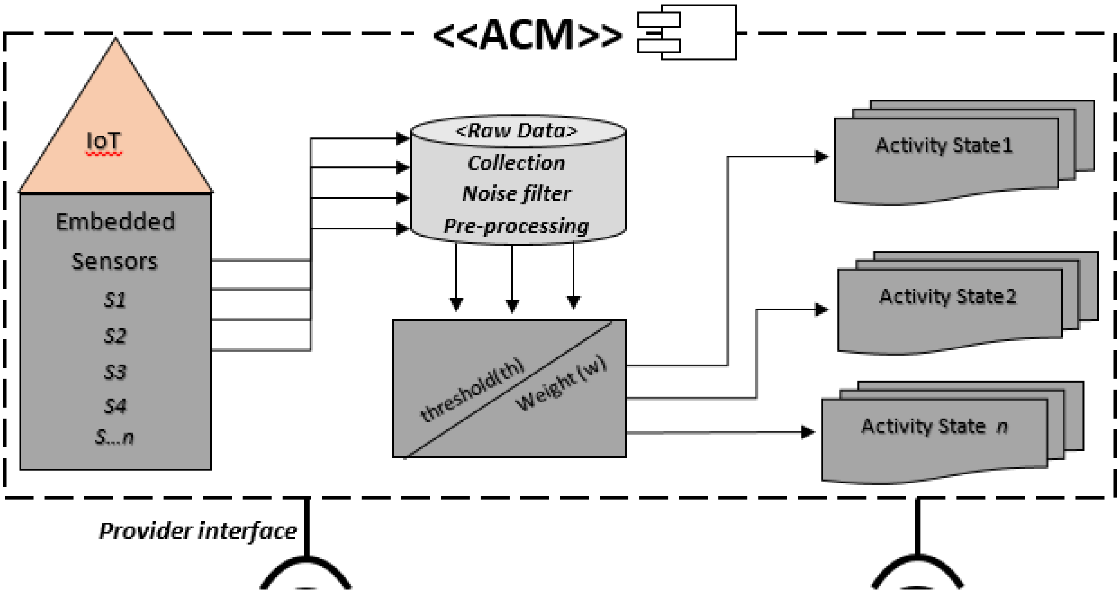


Fig. 3.1 ACM Architecture for Activity Identification

In other words, ambient intelligent space is a spider net where all visible and invisible sensors are embedded in the environment where each sensor represents micro unit-level activity log of inhabitant's daily work patterns. The Horizon 2020 European Commission predicted that about a billion sensor will be interconnected in the embedded environment by 2020, so it can be imagined that enormous amount of data-log would be generated by those IoT sensors which would create new a research opportunity and platform for ambient intelligent smart space research to enhance the assisted living experience around the world. Under such circumstances, extracting the contextual information from such massive data log will require a dynamic framework and statistical model to embrace the diversity to normalize the datasets with appropriate labeling. Therefore, data preprocessing is achieved with threhold(th) and weight(w) schemes. As a primary factor for building an ambient intelligent system, the figure 3.1 propose a novel architecture of Ambient Cognition Model(ACM) to label the activity states and identify inhabitant activity intentions in the Cognitive IoT environment. The fusion of micro-level information provides overall contextual knowledge of the inhabitant's intention and a situation in the environment.

3.2 Noise in IoT data

The heterogeneous sensors generate multi-dimension data in the IoT systems. The multivariate data logs are complex in nature and require preprocessing prior to analytical tasks. The jitter or noise in IoT sensors generated dataset mainly have two common attributes such as i), missing data and ii), false sensor reading. There could be various other routing protocols reasons for data noise but these are the most common in IoT systems. Form the signal processing point of view, few noise filters are effective to remove the unwanted noise form IoT sensors reading such as i), Wiener filter ii), Kalman filter and iii), Savitzky-Golay smoothing filter. These filters are applied to a set of digital data points for the purpose of smoothening the data, to increase the signal-to-noise ratio without greatly distorting the sensor readings. Due to the lower complexity, a lightweight low pass moving average (LMPA) filter is applied to remove the statistical noise including missing values and inaccuracies from the IoT data sets. The LPMA has a simple structure that comes handy to filter unwanted noise component from the intended data in IoT environment

3.3 De-Noising IoT Sensor Data : Low-pass Moving Average (LPMA) filter

In the real-time IoT systems, sensors generate the massive amount of data logs with significant jitter and high-frequency noise included in the datasets. Therefore, smoothing process for noise reduction from the datasets becomes very important to proceed with further data analytics operations. In ACM, a Loss Pass Moving Average (LPMA) filter applied for data preprocessing to ensure unwanted jitter and noise removal. The LPMA filter excludes the first number of data sets and takes the next number following the original subset in the series to find its updated mean value. The process is recursively repeated over the entire series of data to obtain the final updated outcome as moving average versions of the datasets. Also, the implementation of low pass moving average filter ensures the reduction in the time lag between the input and output sensor readings.

$$\bar{\mu} = \frac{1}{n}\left(\sum_{i=0}^{n-1} p_{M-i}\right) = \frac{p_M + P_{M-1} + \cdots + p_{M-(n-1)}}{n} \tag{3.2}$$

$$\therefore M = WindowSize, n = no.of DataSets$$

Furthermore, being the simplest low pass FIR(Finite Impulse Response) filter, it is commonly used for smoothing an array of sampled data/signals. In particular, it takes M samples of inputs at a time and takes the average of those M-samples and produce a signal output point. Whereas, the filter length increases (the parameter M) the unwanted noise removed with a higher smoothness of the output increases. On the other side, the sharp transition in the data is considered increasingly blunt. However, the LPMA filter has excellent time domain response but also lacks in poor frequency response. Therefore, a careful consideration of parameter M has to be taken for removing jitter/noise from the time series data sets (*Moving average filter* 2018). Furthermore, the problem of the missing data point is tackled by assigning a unique ID for each time interval as a counter. Each vector contains sensor readings with a unique ID in continuous time-series (t_n) manners ;

$$< t_n : s_1, s_2, s_3, s_4, s_5, ...s_n >$$

3.4 Activity Allocation

The prime objective of ACM is to establish a relationship among several individual activities together in order to provide a contextual knowledge about inhabitants intentional activities. Particularly, collecting data logs from several individual IoT sensors provide the contextual information about inhabitant's intentional activity and the situation in surrounding environment. In cognitive IoT environment, data log generated from individual sensors leads to extract complex activity information as a collective effort of each small activity they observed. The clear identification of IA (Intention Activity) derived from the SA (Spot Activity) in ACM model. Therefore, a statistical model approach with threshold and weight scheme provides important features to the system as a solution. The table 3.1 provides details about the IoT sensors types and their attributes used for data collection process.

Table 3.1 IoT Sensors Attributes for Data Collection

Attributes	IoT sensors
Observation	luminosity,Motion,Noise,Pressure,Temperature
Data-type	Analogue, Digital
Microcontroller	Arduino Uno Wifi
Sensor Type	Sunfounder IoT sensors
Data Nature	Environmental Data
Datasets	training-testing(1:1)
Number of Sensors	9
Data sampling Interval	per 30 seconds
Start Date	28 September, 2017
End Date	5th, October , 2017

3.4.1 Spot-Activity (SA) : micro-level situation learning

The Spot-Activity(SA) represents the micro level information as the momentary observation or a spot view in the IoT environment. The SA indicates the precise detail about each sensor's observation and no uncertainty is included with it. More specifically, it can be said that observations from the environment are indicated in an absolute manner and all information is highly depended on the sensors positions in the environment.

In case of Activity Daily Living (ADL), many embedded sensors accelerate data at a predefined interval in a time-series manner, which refers to Spot-Activity (sa_1) at time (t_1) interval. In addition, being the first layer to the physical environment, sensor's accelerated data is captured at the micro-unit level, which itself is capable to understand sensors state as $ON(1)/Off(0)$ at a particular time(t_i). As a result, each sensor observation (O_a) provides micro-level information which can not be further broken-down.

3.4.1.1 Statistical model for threshold value

The selection of the threshold value (th_i) is quite critical and often requires a human expert knowledge to make the choices through either experiences or trial-and-error techniques. This is particularly a tedious task in IoT sensors implementation. The environment is embedded with heterogeneous sensors and each sensor has its own capability and frequency to generate raw data in digital and analog format. As per the matrix 3.3, the raw values of sensors have wide range such as :

$$Sensors(S_i) :\in 345 \rightarrow 899 \rightarrow 0 \rightarrow 93$$

$$SensorsRawValue(SRV_{i,t}) = \begin{bmatrix} t_n & S_1 & S_2 & S_3 & S_4 \\ 1 & 266 & 689 & 543 & 200 \\ 2 & 100 & 350 & 450 & 199 \\ 3 & 123 & 431 & 351 & 180 \\ n & 99 & 110 & 140 & 108 \end{bmatrix} \tag{3.3}$$

To formalize the diverse raw values, a statistical approach of feature scaling with mean value and expert knowledge about the IoT environment is applied. The purpose of the mean value is to define a threshold (th_i) for individual sensors and expert knowledge base is to adjust the (th_i) values if required to the context. As a result, threshold values work effectively to measure the imperative change in the environment, by concluding $ON(1)$ or $OFF(0)$ state of sensors. The selection of optimal threshold values of each sensor, (th_i) provides the integral measure of observed variables, which extract the vital information from the raw data values.

Indeed, it is almost impossible to apply machine learning algorithms on raw and cluttered data sets for further data analysis tasks. Therefore, the equation 3.4 is applied for feature scaling with mean value to normalize the scattered data sets. In particular, threshold demarcation (th_i) process is the important factor for data preprocessing in ACM model. Since the number of the data sets in the IoT embedded environment is in the thousands, therefore normalizing each attributes using (th_i) values to label them into absolute states $(0/1)$ is effective and appropriate for the research. The table 3.2 represents the outcome as unified threshold values.

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{3.4}$$

The obtained threshold values (th_i) mold the sensors raw value into to absolute state of 0 and 1. In ACM model, each sensor has the capability to sense at the micro levels

Table 3.2 threshold (th_i) values for individual sensors

th(luminosity)	th(pressure)	th(noise)	th(temp)
783	130	100	100

information such as pressure, motion, light and temperature etc . Therefore, in ACM model ensure if the sensor's raw data value (SRV_i) is less than the threshold value (th_i), then $SpotActivity(SA)$ state would be considered as $OFF(0)$, else if the sensor's raw input value $(SRV_i)$is greater than the threshold value (th_i), $SpotActivity(SA)$ will be identified as $ON(1)$ state of the activity.

$$\begin{cases} 0_{(SA)} & If(SRV_i \leq th_i), SpotActivity = 0, OFF \\ 1_{(SA)} & ElseIf(SRV_i \geq th_i), SpotActivity = 1, ON \end{cases} \tag{3.5}$$

$$SpotActivity(SA) = \begin{bmatrix} t_n & S_1 & S_2 & S_3 & S_4 \\ 1 & 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 & 1 \\ 3 & 1 & 1 & 1 & 1 \\ n & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

Furthermore, IoT environment creates a complex problem to work with numerous sensors at the same time-interval. As a solution, ACM's focal point in figure 3.2 target the spaces which are more active than other . However, in single inhabitant space, a person can be reachable at one place at a time. Therefore, data collection in 24X7 environment, ACM embrace the difference between (0/1) SA states. As a result, complexity to next level contextual knowledge extraction for inhabitant's Intention-Activity (IA) is reduced.

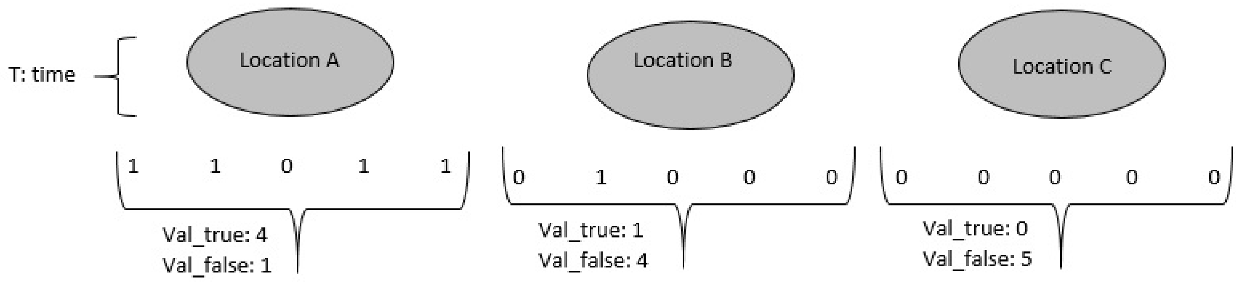


Fig. 3.2 Spot Activity (SA) states represent the inhabitant's presence at a place

3.4.2 Intention-Activity(IA) : contextual situation interpretation

Intention-Activity (IA) represents the contextual situation, where a collection of various Spot-Activities(SA) are merged together to infer inhabitants current intention/situation in the environment. In addition, based on the sensor's influence factor in the environment the appropriate weight (W_i) scheme applied to give them a unique activity label and aggregate individual sensor state as a collective effort of user's activity intention at t time interval. The objective of labeling and inferring inhabitant's activities states is to identify their ADL patterns such as working in living room, cooking in kitchen, sleeping in the bedroom, relaxing in living room, working in kitchen and reading in the bedroom etc..

Therefore, every sensor is vital in the environment to observe inhabitant activities states. During, Spot-Activity (SA) identification each sensor has been defined with a threshold value, which concludes its ON-OFF(0/1) states. Afterward, the technique of fixed weight scheme assignment for each sensor applied to rank individual sensor influence factor in the environment. In the later chapter of the case study implementation, Cognitive Ambient Intelligent Smart Home (CAiSH) represents the smart lab environment, where every sensor plays an important role to identify Spot-Activity (SA) and Intention-Activity (IA) at a specific place and time interval. For instance, force-resistor(pressure) sensor placed under mattress in the bed-room, signify the activity state of inhabitant if he is sleeping or not, while PIR (motion) and photo-resistor (light) sensors observe the activity states such as working/studying /sleeping in the bedroom. Similarly, information of other activity states in the living room and kitchen area is captured respectively.

3.4.2.1 Sensor Weight (S_W) scheme for influential factor

The weight assignment process represents the sensor's impact factor to deduce inhabitants IA state in the IoT environment. Each sensor observes environmental changes in 24x7 fashion and influence of an individual sensor is different from each other to estimate the overall intentions of the inhabitant in the context. In particular, some sensors are more influential than others in order to recognize intention activity context in the IoT environment.

With the prior expert knowledge base about the IoT environment, the weights are defined between a range of [-a, +a]. In ACM model, the range is defined between$[2 \rightarrow 6]$ [2 - 6] based on sensors influential factor in the environment. In other words, weight scheme works as priority ranking of the individual sensor. Therefore, the IA state is a collective outcome of individual sensor SA state observation $< sa_1, sa_2, sa_3, sa_n \rightarrow IA_j >$. Particularly in IoT environment, location and the placement of sensors play an important role to define their influential factor compared to other sensors. The sensors with a higher influential factor, assigned with more higher weight as compared to other. For instance, pressure sensor (p_1) represents the inhabitant's occupancy in the bedroom area to identify if he is sleeping or not, might have been assigned more weight(w_i) as compared to the motion (m_1) and light (l_1) sensors at the same location and time interval. In addition, a location weight (L_W) is plugged-in as for $< BedRoom \rightarrow 0, LivingRoom \rightarrow 10, Kitchen \rightarrow 20 >$ to segregate several IA states from each other such as cooking, cleaning, sleeping, and working etc. The table 3.3, table 3.5 and table 3.4 represents the weight schemes for the bedroom, kitchen and living room areas respectively.

Table 3.3 Sensor Weight Scheme for Bedroom Area

Weight-range	$W_p(pressure)$	$W_l(luminosity)$	$W_m(motion)$	(L_W)
1-10	6	2	2	0

As a result, the weight assignment process is a collective effort of expert knowledge

Table 3.4 Sensor Weight Scheme for Livingroom Area

Weight-range	$W_p(pressure)$	$W_l(luminosity)$	$W_m(motion)$	(L_W)
1-10	6	2	2	10

Table 3.5 Sensor Weight Scheme for Kitchen Area

Weight range	$W_p(pressure)$	$W_t(temprature)$	$W_l(luminosity)$	(L_W)
1-10	2	6	2	20

base, considering the environmental factors to identify and the ranking sensors as per their criticality in the process. For the data-consistency , the sum of sensor weight $\sum(SW)$ vector is equal to 10, such as $W_p = 6, W_l = 2, W_m = 2$;

$$\sum_{j=1}^{n}(W_n) = 10 \therefore W \in (0, 10), n : number of sensors \tag{3.7}$$

The inhabitant's(IA_i) states identification is achieved with equation 3.8, where (SA_i) states are multiply with weight(W_j) scheme and added the location weight (L_W) such that ;

$$\Delta IA_i = \sum_{i=1}^{n}(SA_i \times W_j) + LW_k \tag{3.8}$$

$$\therefore SA_i = (sa_1, sa_2...sa_i), W_j = (w_1, w_2...w_j), LW_k = (lw_1, lw_2...lw_k)$$

As a result of equation 3.9, $(IA)_i$states would be extracted to identify inhabitant's activity intention and situation in the surrounding environment. Furthermore, IA states are labeled appropriately in IF-ELSE condtion such that;

$$|C_S(CurrentStates)| = \begin{cases} CS_A & If(IA) \geq Score \\ CS_B & If(IA) \leq Score \\ CS_C & If(IA) = Score \\ CS_D & Else(IA) = 0 \end{cases} \tag{3.9}$$

Hence, in the Ambient Cognition Model (ACM) heterogeneous IoT sensor's accelerated

Table 3.6 Intention -Activity(IA) State labels

$\Delta C_S(CurrentState)$	Name	IA	Location
1	Sleeping	$IA = 6$	Bedroom
2	StudyonBed	$IA > 6$	Bedroom
3	Working	$IA < 6$	Bedroom
0	Absent	$IA = 0$	Bedroom
4	Relaxing	$IA = 16$	Livingroom
5	Excercising/houshold work	$IA < 16$	Livingroom
6	Study	$IA > 16$	Livingroom
0	Absent	$IA = 0$	LivingRoom
7	Cooking	$IA > 26$	Kitchen
8	Working	$IA < 26$	Kitchen
0	Absent	$IA = 0$	Kitchen

time-series data have been preprocessed with appropriate statistical mode approach. Each sensor accelerated data point represent the sequence of Spot-Activity (SA_i) at specific time (t_i) interval. The weight scheme(W_i) facilitate the identification process of inhabitant Activity Intention (IA_i) in the surrounding environment. Therefore, the prime objective of ACM to identify inhabitant's IA(intention activity) states is successfully achieved for the stage one experiments. Afterward, in the stage, two experiments, the identified(IA_i) states would be used as input data sets for Ambient Expert system(AEM) to run machine learning experiments in order to build a predictive model to forecast inhabitant activity state pattern for performing pro-active tasks in the ambient-intelligent environment. The whole eco-system of Cognitive IoT (Internet of Things) system is divided into two main parts known as ACM and AEM(Ambient Expert Model).

3.5 Conclusion

The cognitive IoT sensor data preprocessing for the inhabitant's activity state identification with unique and appropriate labeling is the primary objective of Ambient Cognition Model. In contrast, the heterogeneous IoT sensors generate the enormous amount of data which can not be used for further machine learning analysis until the appropriate preprocessing and labeling of identified activity states have been performed. The activities identified at the micro level represented as SA(Spot-Activity) and identification at the contextual level represented as IA(Intention-Activity) states. Therefore, the ACM model efficiently applies the threshold and weighting scheme approach for inhabitant ADL identification and extracts contextual information for further machine learning models predictive analysis.

The contribution to the knowledge from ACM has been published in IoTBDS, 2017 conference with the title of ’ A cognitive IoE approach to ambient intelligent smart home’. The peer review assessment ensures the novelty of the proposed work and embraces the critical reviews efficiently in the further research work.

CHAPTER 4

Ambient-Expert Model : rule based time-series ADL forecasting model

Overview

This chapter describes the architecture of Ambient-Expert Model (AEM), based on the supervised machine learning approach of HMM (Hidden Markov Model) to forecast ADL(Activity Daily Living) states-patterns of the inhabitant and rule-based system to execute tasks in a proactive manner. The prime objective of AEM is to recognize the inhabitant's activity state patterns and make predictions for pro-active task execution in the IoT enabled smart home scenario. The identified SA (Spot-Activity) and IA (Intention-Activity) from the ACM (Ambient Cognition Model) is employed as the input observation sequences into the AEM system in order to train system for the ADL pattern forecasting in time series manner. The rule-sets are applied on predicted activity states to perform pro-active task execution in the IoT enabled smart home environment. The architecture of AEM, follow a data-driven probabilistic model approach which makes system ambient intelligent to understand the surrounding situations to apply appropriate rule sets for pro-active task execution. The Ambient Expert Model (AES) is trained over time series activity states datasets and consequently tested for accuracy measures for the use case study.

4.1 AEM (Ambient Expert Model)

The AEM (Ambient Expert Model) is an important element for the design and development of modern IoT smart ambient intelligent systems. In the AEM, information about inhabitant's activity states is vital to discover the hidden ADL (Activity Daily Living) patterns in the environment. The task of data preprocessing and feature extraction is achieved by ACM(Ambient Cognition Model). More specifically, ACM provides the initial platform for activity identification and labeling them into various SA(Spot-Activity) and IA (Intention Activity) states. Later, these activity states are used as the input datasets in AEM to perform predictive analysis on time-series data for activity pattern recognition. The application of supervised machine learning approach of HMM (Hidden Markov model) is effectively embraced in the AEM architecture. The whole ecosystem of AEM follows the principle of discrete probabilistic model, where an individual spot observation$(SA_i)$is inferred to a specific activity state(IA_i) with a higher posterior probability$< (PS_h)$, h$\rightarrow high >$ value as compared to low posterior probability$< (PS_l)$ l $\rightarrow low >$ states. In AEM, the Viterbi and Baum-Welch algorithms are applied for parameter estimation in order to identify most likelihood activity Intention Activity (IA) states(IA_i^h) with higher probability. The identified likelihood states(IA_i^h), provide the recognized ADL pattern for proactive task execution. The rule-based system adapt the ADL patterns and apply certain rule-sets for proactive task execution in the environment. As Mavropoulos and Chung (2014) suggested, a decision-making system can only be successfully achieved with the expert knowledge that emulates the decision-making ability of a human expert, such that expert systems intend to emulate in all aspect of the human knowledge base to perform specific tasks. (Tran and Wagner 1999) In figure 4.1, the proposed AEM (Ambient-Expert Model) follows the discrete probabilistic approach of Hidden Markov Model(HMM) using Expectation Maximization (EM) methods to train model parameters with the Baum-Welch and Viterbi algorithms. The purpose of applying EM is to maximize the most likelihood Activity Daily living (ADL) patterns. Furthermore, the predictive regression modeling starts with historical time-series data where supervised machine learning algorithms examine the historical data and check for patterns of time decomposition, such as trends, seasonal

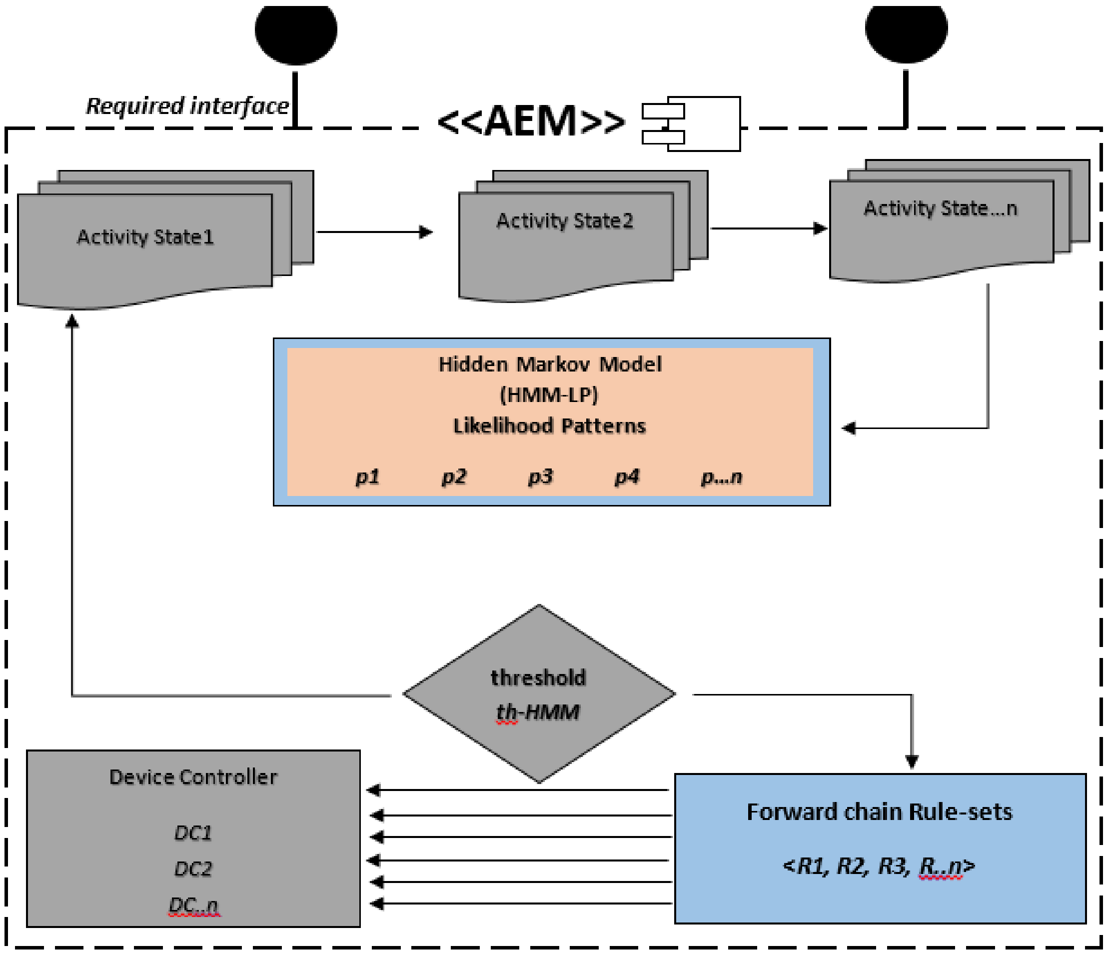


Fig. 4.1 The Architecture of AEM to Forecasting Time Series ADL Patterns

patterns, cyclic patterns, and regularities. Therefore, in many use case scenario organizations including product lines, financial market analysis, and product-sales patterns, uses the same approach of time series forecasting to evaluate the probable technical costs and consumers demand based on the historical datasets analysis (*TechTarget* n.d.).

4.1.1 Activity Cycle in Daily Routine

The individual set of activities is performed together in order to infer a specific task situation in the smart home scenarios. The figure 4.2 shown daily activity routines and related task for proactive execution through predefined rules sets for proactive and enhanced assisted living IoT environment. The behavioral changes in ADL routines could be caused by the inhabitant's lifestyle or the other environmental factors such as sensors placement in the smart home. For instance, weather conditions could be the most affecting constraint to influence inhabitant's daily activity sequences and the placement of motion sensors in inappropriate space to capture inhabitant's movement presence in a time-series manner. Therefore, in IoT sensors, it is very important to place them at appropriate place and set threshold range to capture action sequences data log in a more logical way with consideration of geographic configuration. In the table 4.1 individual activity states are labeled in a discrete manner with their appropriate aggregated weight scheme for IA(intention-activity) identification.

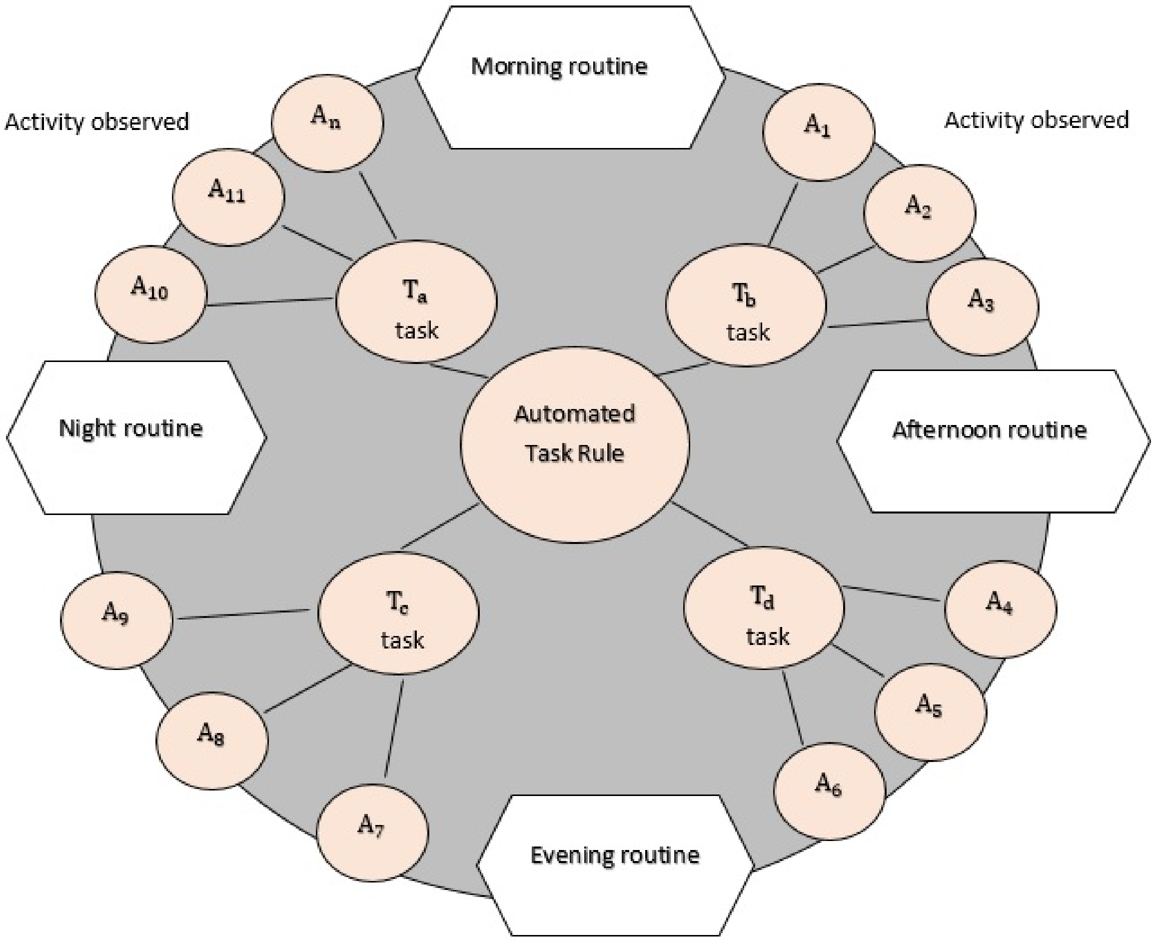


Fig. 4.2 Activity Cycle in Daily Routine

Table 4.1 IA(intentional-activity) state label and names based on weight aggregation value

Aggregated Weight $\sum(sa_i)$	IA (Intention Activity)	Activity-Name	Location
=6	1	Sleeping	Bedroom
$\geq$ 8	2	Reading on Bed	Bedroom
$\geq$ 4	3	Working	Bedroom
0	0	not available	Bedroom
=14	4	Working	Livingroom
=16	5	Relaxing/Sitting	Livingroom
$\geq$ 18	6	Reading on sofa	Livingroom
0	0	not available	Livingroom
=26	7	Cooking	KitchenArea
$\geq$ 28	8	Cooking-cleaning	KitchenArea
=24	9	Cleaning	KitchenArea
0	0	not available	Livingroom

4.2 Appropriate machine learning method

In the context of current research challenge, various pattern recognition algorithms fit into the system. More specifically, the sequential time series pattern recognition algorithms with low computing resources would be suitable to embrace the research challenge. Such as the Navies Bayesian is very effective and requires less computing resource and processing time to train the model. But in terms of results, the output of Naive Bayesian is less accurate to predict the sequential pattern with low error rate. The reason behind that is, NB only considers previous state $n, (n-1)$ in order to predict $(n+1)$ activity state. While in current research a chain of activity states are related to each other in a sequential manner $n, (n+1.....m)$, which is not possible to cover through the NB model. On the other hand, RNN is very effective for time series prediction with less error rate but require more computing resource and processing time to train the model. In addition, various input parameters are needed including hidden layers size, input delay, divide parameters etc.. As a result, the frequent effort required to fine-tune the model for better result outcomes. Where in the current research challenge, we only know the type of activity state as the input parameters. In addition, IoT devices have limited processing capacity to run RNN models. on the other side, HMM is a well-known method for sequential pattern recognition task. In HMM the input parameters are required once and change internal automatically based on the current dataset, which is well suited for our research context. The type of activity states are known through the AEM model and Datasets are periodically updated through Data-pool, which makes HMM to integrate easily into the system. In addition compared to the RNN, HMM require less computing resources and processing time to train the model with low error rate. As a result HMM turn to be the most effective method for forecasting ADL patterns in the IoT systems

4.3 Forecasting ADL patterns with Hidden Markov Model

The Hidden Markov Model (HMM) is one of the most competent discrete probabilistic models for time series sequential events that can observe in the scope of pattern recognition to provide a universal solution for many different problems (Jurafsky and Martin 2017). The Hidden Markov models and their generalizations provide a universal solutions for wide range of problems areas such as Automatic Speech Recognition (ASR) systems, health care diagnosis and prognosis, forecasting financial economics, image analysis, anomalies event detection, fault diagnosis in production lines, weather forecasting, protein sequence detection and automated assisted living experiments. In contrast, the input factor for all mentioned problems cases requires a discrete labeled datasets in time-series training manner. Therefore, HMM is suitable and provides a robust platform for discrete-time series data analysis to forecast patterns with maximum likelihood probabilities. The training method of HMM is initiated with expectation maximization (EM or Baum-Welch) algorithm, which is attractive and used for many other probabilistic models, such as the finite mixture models (FMM) and the Gaussian mixture models (GMM) (Huda et al. 2014).

In Hidden Markov Model, the input variables include initial probability, transmission probability and emission probability matrix of activity state observations. As shown in figure 4.3 , HMM models the joint probability distribution of those variables and naively assumes that hidden state of (IA_t) at the time step (t) only depends on hidden state at previous time step, (IA_{t-1}), while the observation (SA) at time(t) only depends on the hidden state at the same time slice. Therefore, HMM can be mathematically described by three parameters: the initial state y_1, transition distribution $p(y_k|y_{k-1})$, and emission probability $p(x_k|y_k)$, the joint probability distribution of the variables can be formulated as follows :

$$p(x, y) = p(y_1)p(x_1|y_1)\pi_{k=2}^{K}p(y_k|y_{k-1})p(x_k|y_k) \tag{4.1}$$

In the AEM model, ADL states are considered as input data points for HMM parameters such that all IA (Intention Activity) states $< IA_{(all)} \rightarrow Y >$ represent the states as , $< IA = ia_t^1, ia_{t+1}^2, ia_{t+3}^3, ...ia_N^n) >$ states at discrete time (t) interval. Likewise, all micro level SA(Spot-Activity) $< SA_{(all)} \rightarrow X >$, denoted as observations such that $< SA = (sa_t^1, sa_{t+1}^2, sa_{t+2}^3, ...sa_M^m) >$. The HMM model parameter notation would be as $\lambda = (A, B, \pi)$, where ;

$A = (IA_{ij}), 1 \leq i, j \leq N$ is the state transition probability distribution, with $a_{ij} = P(IA_t|IA_{t-1})$.

$B = (b_j(IA_k)$ for $1 \leq i \leq N, 1 \leq k \leq M$ is the emission probability distribution in state j, with $b_j(SA_k) = P(SA_k$ at $t = IA_j|q_t = IA_j)$.

$\pi = P(q_i = IA_i), 1 \leq N$ is the initial state distribution probability.

Furthermore, figure 4.3 represents the transition probability of IA state (IA_t) at time step t only depends on the previous Intentional-activity state(IA_{t-1}). While Spot-Activity(SP_t)observation at time t only depends on the Intentional-Activity(IA_t)state at the same time slice(Wen and Wang 2016). The whole process of HMM is based on transition $P(y_k|y_{k-1})$ and emission$P(x_k|y_k)$ probability distribution between the observed sequences. Each observed sequence is univariate, represent along with time series $(y1, y_2, y_3, ...y_n)$ are discreet in nature due its time series representation such that the observed activity $< y_1, t_1 >$represents the discreet data points in HMM(Trabelsi et al. 2013).

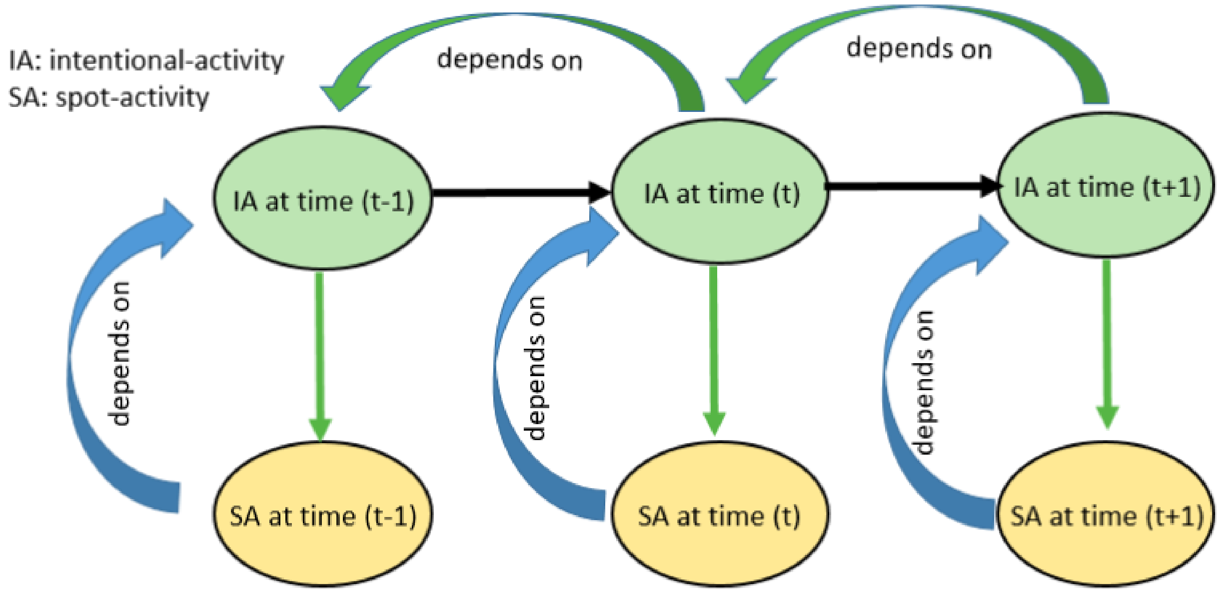


Fig. 4.3 Graphical model of HMM in AEM Model

4.3.1 ADL Parameters Estimation with Baum-Welch Algorithm

The parameter estimation and standard training of HMM is obtained by the EM (expectation maximization) algorithm known as Baum-Welch that is attractive and used for many other probabilistic models, such as the finite mixture models (FMM)and the Gaussian mixture models (GMM). An HMM has two stochastic processes where observations are maintained by multiple distributions with switching between the states. EM computes HMM parameters by maximizing the log-likelihood $< L = logP(O|\lambda) >$ of the observed data $< O_i >$ where the maximum-likelihood (ML) estimate of HMM parameters are ;

$$\lambda^{ML} = argmax\lambda logP(O|\lambda) \tag{4.2}$$

$$L = log[P(O|)] = log[P(O,q|)]log[P(q|O,)] \tag{4.3}$$

The EM estimates the HMM parameters in two steps. Firstly, in the E-step (expectation), EM computes log-likelihood of complete data with respect to $< \ P(q|O,\lambda^k) \ >$ given the initial value, $< \lambda^k >$. Secondly, in the M-step (maximization), EM maximizes log-likelihood of complete data of E-step and obtains a new set of values for model parameters $< \lambda^{k+1} >$ (Huda et al. 2014).

As known from its name, EM algorithm's prime task is to increase the log-likelihood patterns from the observed datasets for making a better probabilistic model. In the Matlab, Hidden Markov Model parameter estimation from the known sequences and states are achieved with hmmestimate method to achieve optimum transition and emission probability matrix such that ;

$$[TRANS, EMIS] = hmmestimate(seq, states) \tag{4.4}$$

The hmmestimate calculates the maximum likelihood transition(TRANS) and emission(EMIS) probabilities of a hidden Markov model for the Observation Sequence(SA)with Intention-Activity (IA)states.

4.3.1.1 Decoding Likelihood ADL Pattern

The process of discovering hidden ADL (Activity Daily Living) patterns with higher probabilities from given observation sequences in HMM, is known as decoding process. The Viterbi algorithm is commonly used in order to identify likelihood sequences within the scope. The algorithm is named after "Andrew Viterbi", the application of algorithm is to maximize problem involving probability. In other words, Viterbi algorithms can be well understood as a max-product algorithm, whose primary function is to find the most likely subset of latent variable among a large number of datasets. The application of Viterbi algorithms is well embraced in all graphical model such as Bayesian Network, Hidden Markov Model, and Conditional Random Fields to find the best match with higher maximum probability sub-sequences of observations.

$$q = argmaxlog(P(q|O, \lambda)) \tag{4.5}$$

To simplify this probability function, we can write the equation as ;

$$q* = argmaxlogp(q_1)\pi_{l=2}^{L}(P(q_1|q_{l-1})P(O_l|q_l)) \tag{4.6}$$

The objective function of the Viterbi algorithm to find the most likelihood sequence in ADL with higher probability can be written as ;

$$maxlog(p(L, q_L)) \tag{4.7}$$

The most probable state path using Viterbi algorithm in MATLAB, defined as $hmmviterbi$ function, where $[STATES(q)] = hmmviterbi(observation_{seq}, prob_{transMat}, prob_{emisMat})$ given an set of observations sequences, transition probability matrix and emission probability matrix to calculate the most likelihood ADL states. Particularly, because hmmviterbi computes the most likely path based on the fact that the model begins in state 1, the function hmmviterbi begins with the model in state 1 at step 0, prior to the first emission. The Maximum likelihood estimation chooses the state with the highest probability at the best estimate at each time (t_i)step.

4.3.2 Posterior Probability of ADL patterns

Once the inhabitant hidden ADL patterns are recognized, the next step required is to calculate the posterior probability. The conditional probability distribution is based on the evidence obtained from the experiments, which means all the evidence is taken into account for examining each probability distribution. The relationship between posterior probability and likelihood function represents the inhabitant's activity state patterns in probability distribution manner. In contrast, the higher probability of activity states will be obtained as the most likelihood activity state of hidden pattern, and can be formulate as ;

$$p(y_k|x_k) \tag{4.8}$$

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \tag{4.9}$$

In particular, $posterior = (likelihood * prior)/evidence$ function represents the posterior probability of activity states, which are the conditional probabilities of being at activity-state $< St_{(k,i)} > k$ at step i interval, given the observation-sequences $< O_i >$. The calculation of posterior probability state is well presented in ”$< hmmdecode >$” function in the MATLAB through probability distribution matrix.

$$P = posterior(obj, X) \tag{4.10}$$

$$[P, nlogl] = posterior(obj, X) \tag{4.11}$$

The $P = posterior(obj, X)$ returns the posterior probabilities of each of the ”k” components in the Gaussian mixture distribution defined by ”obj” for each observation in the data matrix X. In particular, ”X” is [n X d] dimension matrix, where ”n” is the number of observations and ”d” is the dimension of the data, whereas ”obj” is an object created by ”gmdistribution ”or ”fitgmdist” . The ”P” is ”n-by-k”, with P(I,J) the probability of component ”J ” given observation ”I”. In contrast, posterior treats NaN values as missing data. Rows of X with NaN values are excluded from the computation. In addition, ”[P,nlogl] = posterior(obj,X)” also returns ”nlogl”, the negative log-likelihood of the data. In the MATLAB, the Hidden Markov model posterior state probabilities can be obtained by ”$hmmdecode$ ” function, such that ;

$$PSTATES = hmmdecode(seq, TRANS, EMIS) \tag{4.12}$$

Furthermore, the posterior state probabilities are the conditional probabilities of being at state ”k” at step ”i”, given the observed sequences. The HMM parameters is given a transition probability matrix (TRANS), and an emissions probability matrix (EMIS). Where as, $< TRANS_{(i,j)} >$ is the probability of transition from state ”i” to state ”j” and $< EMIS_{(k,seq)} >$ is the probability that symbol ”seq” is emitted from state ”k”. The

PSTATES is an array with the same length as seq and one row for each state in the model. The function "hmmdecode" begins with the model in state 1 at step 0, prior to the first emission. The probabilities in PSTATES is based on the fact that the model begins in state 1.

4.4 Rule Based System for Proactive Task Execution

The prime objective of AEM is to forecast and make predictions about inhabitant activities states in order to provide proactive services via implying expert rules for task execution. Therefore, the rule-based system enables the proactive task execution as an expert does in the problem scenario to understand the situation and intentions in the surrounding environment with its domain-specific knowledge base. More specifically, the rule-based system processes the environmental information in the human-crafted rule sets in $< if - else >$ conditions. In the rule-based system, a domain-specific knowledge base is used to make detections and choices for executing instructions to accomplish a specific set of tasks. Therefore, the rule-based system emulates the decision-making ability of a human expert, which means a system intended to act as a human with expert knowledge for performing the set of tasks. In the rule-based system, complex problems are solved by reasoning knowledge expressed in rule sets to focus on the specific set of problems inputs and perform complex operations as the outcome. The application of rule-based system has become universal and benefiting several industries for troubleshooting and complex large data analysis problems including financial credit rating, loan authorization, health informatics and product-assemblies etc.

Table 4.2 Expert Rule Sets for the CAiSH Environment

Label	$IA_i(IntentioalActivity)$	$L_i(Location)$	$RBS_i(RuleBasedSystem)$
1	IA_{sleep}	$L_{bedroom}$	$SwitchLightOFF \cap LockMainDoorON \cap SetTempON$
2	IA_{work}	$L_{bedroom}$	$SwitchStudyLightON \cap SwitchCoffeeMakerON \cap SetTempON$
3	IA_{relax}	$L_{livingroom}$	$SwitchLightOFF \cap SwitchTVON \cap SetTempON$
4	IA_{work}	$L_{livingroom}$	$SwitchStudyLightON \cap SwitchTVOFF \cap SetTempON \cap SwitchRadioON$
5	$IA_{cooking}$	$L_{kitchen}$	$SwitchWashingMachineON \cap SwitchDishWasherON \cap SwitchExhaustFanON$
6	IA_{eating}	$L_{kitchen}$	$SwtichRadioON \cap SwitchCoffeeMakerON$
7	$IA_{reading}$	$L_{bedroom}$	$SwitichStudyLightON \cap SetTempON \cap LockMainDoorON$
8	$IA_{reading}$	$L_{livingroom}$	$SwitchStudyLightON \cap SetCoffeeMakerON$
9	$IA_{notpresent}$	L_{empty}	$SwitchLightOFF \cap SetVoiceMessageON \cap LockMainDoorON$

In the RBS (Rule Based System), rules represent the expert knowledge to perform a specific set of tasks based on current ambient states information and situation. In the RBS rule consist of two parts, the IF part called the antecedent(premise or condition) and the THEN part called the consequent(conclusion or action).

IF $< antecedent >$

THEN $< consequent >$

However, a rule can have multiple antecedents joined by the keywords AND (conjunction), OR (disjunction) or a combination of both.

IF $< antecedent1 >$ IF $< antecedent1 >$

AND $< antecedent2 >$ OR $< antecedent2 >$

$*$

$*$

$*$

AND $< antecedentn >$ OR $< antecedentn >$

THEN $< consequent >$ THEN $< consequent >$

According to table 4.2, inhabitant's IA (Intentional Activity) state (IA_i) and location (L_i) play as antecedent input parameters in order to perform some task in the proactive manner using (RBS_i) expert-rules as the consequent output. For instance if the inhabitant's current state is sleeping in the bedroom location then rule executes to switching off lights, locking main door and setting temperatures of the house in proactive and automated manner. Same as if the inhabitant's IA (Intentional Activity) state is eating in the kitchen location then a rule set would be activated to switch on radio and later to start coffee machine to ensure an enhanced assisted living experience in the smart home.

As shown in the figure 4.4 , the process of task execution in AEM (Ambient Expert Model) is divided into four parts, including : i) working state patterns, ii) forward chain rule inferencing, iii) rule base , and iv) device-controllers. A detailed explanation of each part is provided for better understanding of the system.

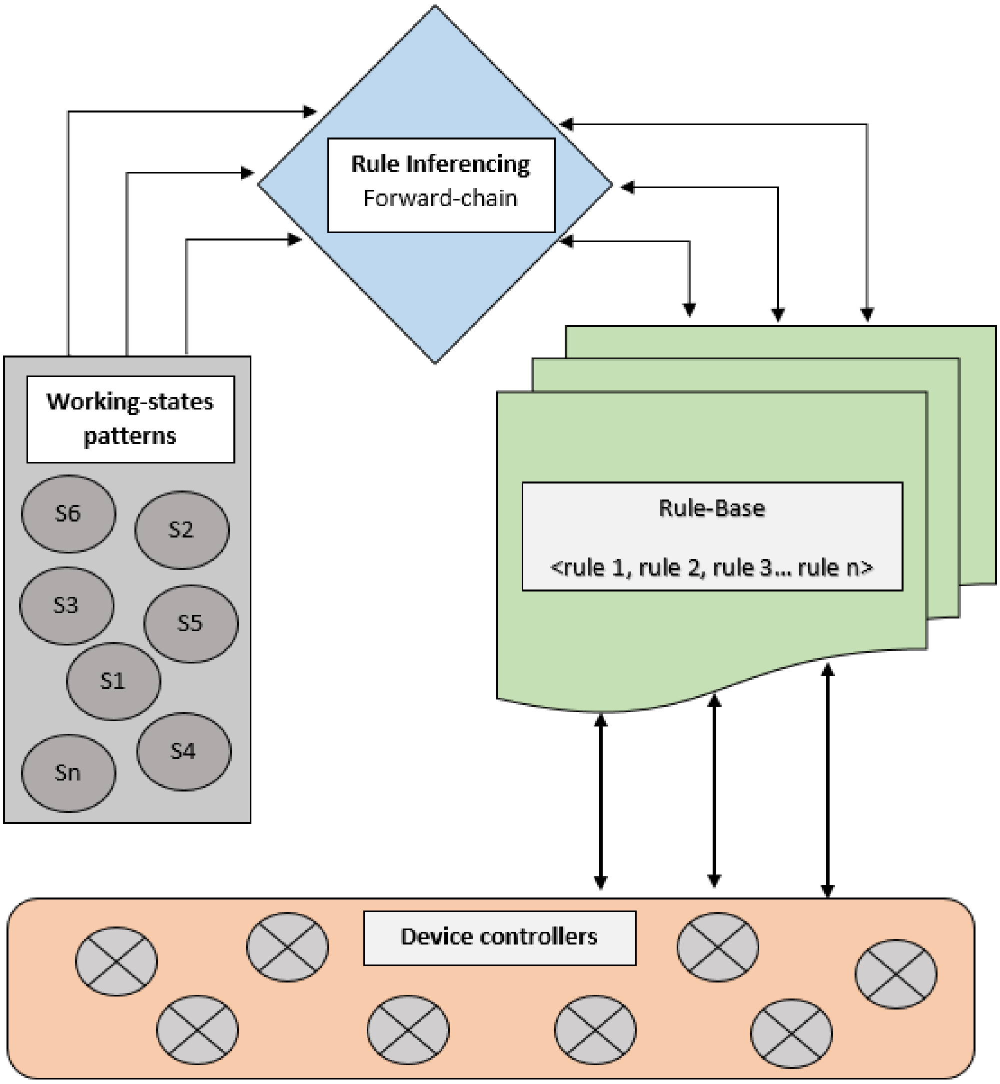


Fig. 4.4 The Rule Based System for Ambient Expert Model

4.4.1 Working State Pattern

The input interface of the rule based system , starts with working state pattern. The sensor data sets are preprocessed in AEM through Hidden Markov Model to predict an inhabitant's ADL patterns in a time series manner. The type of inhabitant's ADL patterns are depended on the number of a identified activity states such as sleeping, reading, working, cooking and watching TV. Once the rule based system receives the activity-states inputs, the next step starts to compare them with the rule base system's predefined activity sets.

4.4.2 Association Rule Inferencing with Forward Chain

Association rule inferencing is a rule-based machine learning method for discovering potential relations between working memories variables and the rule base repository. The rule inferencing performs the 'IF' part as an antecedent condition of RBS comparing to the consequent part for condition fulfillment. Every rule is composed by two different sets of items, also known as item sets $X < antecedent >$ and $Y < consequent >$. The two kind of chaining methods exists for rule inferencing, know as data-oriented forward chaining and goal-oriented backward chaining.

Forward Chaining : Data-Oriented

Existing facts matched to rule antecedents

Matching rules result in consequent

Backward Chaining : Goal-Oriented

Select goal to match rule-consequents.

Checking for match between rule-antecedents and facts

Repeat until goal matches fact

The AEM follows the forward Chaining method which is a data-driven approach which drives to a particular goal from the given knowledge base and set of inference rules. It also known a production system as each of the rule represents miniature procedures where the antecedents facts match the working storage elements, afterward action sets consequents are activated accordingly.

4.4.3 Rule Base

Once, the working memory is compared to the rule inferencing logic, a specific rule or rule sets will be activated. The two-way communication between the rule-inference and the rule base makes system to obtain appropriate rule for the task execution. Particularly, rule base performs the 'Then' part to achieve the task, based on the rule inferencing in

Algorithm 2 Forward Chain Rule Inference

$IF(IAState = Sleep)$
AND
$IF(Location = Bedroom)$
THEN
$SwitchLightOFF$ ▷ Command to Adruino D.C
AND
$LockMainDoor$ ▷ Command to Adruino D.C
AND
$SetTemparature$ ▷ Command to Adruino D.C

'IF' part of working memory space.

4.4.4 Device Controller

As the last layer of RBS, device controller works as the physical layer of the cognitive ambient intelligent smart home environment. On the basis of the activated rule set, a specific device controller or controllers initiate the activation and deactivation of multiple devices and appliances in the smart home. Device controllers work on the absolute logic such as ON or OFF (1/0) state received from above decision-making layer. Every device controller would be associated with RBS to enhance the assisted living experience in the cognitive ambient intelligent smart home environment.

4.5 Conclusion

The AEM follows the Markovian process to identify the hidden ADL patterns of inhabitants in the smart home scenario through the training procedure of Hidden Markov Model. Furthermore, ACM enables the labeling task of the discrete set of identified activity states (SA and IA) to provide the ground truth as training datasets in time-series order for AEM model training and evaluation process. The predictive regression modeling approach for the ambient intelligent smart home environment ensure the time-series forecasting of the inhabitant's situation and activity intentions for proactive task execution to enhance the assisted living experience. The rule-based system enables the AEM to achieve automated task accomplishment in a proactive manner and fulfill the AEM (Ambient Expert Model) viability in the cognitive IoT environment.

The contribution to the knowledge has been published in well known SEKE, 2017 conference title as ”Home Automation: HMM-based fuzzy rule engine for Ambient Intelligent Smart space”. The research paper received valuable feedbacks from critics which ensure the credibility and novelty of proposed work. The feedback has been addressed in the research work with further modification.

CHAPTER 5

CAiSH (Cognitive Ambient-Intelligent Smart home) Framework

5.1 Overview

The CAiSH framework provides a complete architecture of the cognitive ambient intelligent smart home through the integration of two intelligent models namely ACM (Ambient Cognition Model) and AEM (Ambient Expert Model). The ACM provides identification of inhabitants activity intentions in the cognitive IoT environment and assigns discrete activity state labels with a statistical model approach. The data pre-processing, activity identification and activity labeling are the primary objectives of ACM model. On the other hand, the predictive regression modeling approach of AEM model pursue the Hidden Markov Model principles to forecast inhabitant's hidden ADL(Activity Daily Livings) patterns. The two interdependent model provides a unified framework of the CAiSH (Cognitive Ambient Intelligent Smart Home). In Cognitive IoT research, the modern IoT technologies are integrating with machine learning models to develop a proactive IoT system which understands inhabitants situations and intentions in the surrounding environment. The Figure 5.1 represents the proposed CAiSH framework as a novel contribution to intelligent IoT systems. However, the concept of CAiSH framework could be successfully applied in smart spaces for Elderly Assisted Living, Health Care Systems, Smart City, Urban Planning and various IoT automation projects and research fields.

The provider interface of ACM provide the training data sets for the machine learning models. While, the required interface of AEM takes the training data sets from ACM in order to perform the ADL pattern forecasting of inhabitant. The knowledge base form

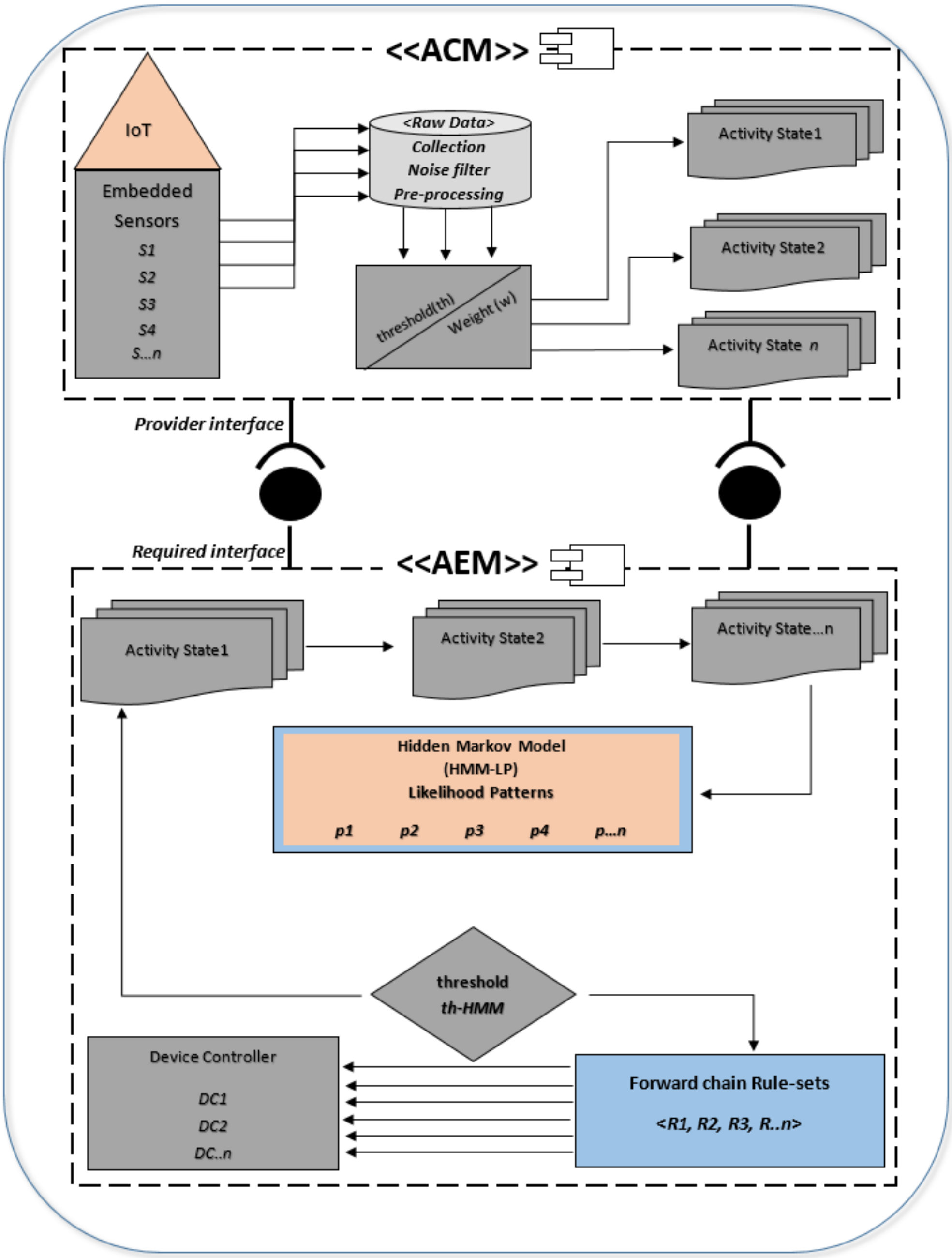


Fig. 5.1 The framework of CAiSH (Cognitive Ambient-Intelligent Smart Home)

CBSE (Component Based Software Engineering) applied to connect two architectures together with provider and required interfaces. Such CBSE approach modularized the framework to ensure the re-usability and loose coupling techniques.

5.2 ACM (Ambient Cognition Model) for Activity State Identification

The ACM (Ambient Cognition Model) is one of two most important component of the CAiSH framework. It provides the primary building block to the CAiSH framework through focusing on the capability to understand the situations or intentions of the inhabitant in surrounding environment. In IoT smart spaces, multiple sensors are embedded in the environment to capture the data of every single activity in the environment. Although, the data generated from heterogeneous sensors in a time-series manner, which creates a data preprocessing problem. However, the data log provides micro-level activity state information but to identify overall intentions or situation in the environment is the very complicated task in cognitive IoT environment. Therefore, the ACM apply threshold (th_i) and weight schemes(W_i) in statistical model approach for the data normalization and activity state identification task. The ACM extracts the information from SA(spot activity) $< sa_1, sa_2, sa_3, ..sa_n >$ to identify inhabitant's IA(Intentional Activity) states $< ia_1, ia_2, ia_3, ...ia_m >$ in the environment with discrete labels.

5.3 AEM (Ambient Expert Model) for ADL Pattern Prediction

The AEM (Ambient Expert Model) is the second most important component of the CAiSH framework, it forecast inhabitants ADL (Activity Daily Living) patterns and perform proactive task execution through the RBS(Rule-Based System). The identified activity states of ACM model are used in the AEM model for the HMM training process to forecast hidden ADL patterns of the inhabitant. The supervised machine learning model of HMM (Hidden Markov model) applied to predict inhabitant's ADL patterns for the pro-active task execution. The proactive task execution is performed by the RBS (Rule-Based System) to enhance the assisted living experience in the smart home scenario. The RBS represents the expert knowledge base in a smart home environment with their rule sets to activate the appropriate rules for the proactive task executions. In the AEM model, regressive analysis of machine learning methods enables the system to become ambient intelligent and understand the inhabitant intentions and situation in the surrounding cognitive

IoT environment.(Technologies and Change 2014)

5.4 CAiSH (Cognitive Ambient intelligent Smart Home) Framework

The cognitive IoT (Internet of Things) provides the fundamental ICT infrastructure to enable independent assisted living IoT environment for solving upcoming societal challenges of the smart homes. For the growing urban population and elderly population around the globe, the AAL (Ambient Assisted Living) solutions aim to increase the independent assisted quality life, which makes the increased demands of intelligent IoT systems in the future.(Technologies and Change 2014)

Figure 5.1 describes the CAiSH framework as a modular architecture, where the ACM and AEM models are integrated into the framework. In contrast, component-based software engineering approach has been applied to divide the complete framework into two main components. The CAiSH intends to facilitate the ideal smart home framework by offering the ambient intelligent IoT services in a proactive manner. Therefore, the CAiSH framework reinforces the amalgamation between time-series data preprocessing, activity identification-labeling, forecasting inhabitant ADL patterns and the rule-based proactive task executions. As mentioned in the ACM model, inhabitants activity state information at micro level helps to extract the contextual knowledge about the surrounding environment in order to build an ambient intelligent smart space. The prime objective of the CAiSH framework is to improve the usability of applications by adapting ACM and AEM model functionality. In particular, the RBS (Rule-Based System) works as an expert system to apply appropriate rules with forward chaining method to perform certain tasks in a proactive manner. The expert system is derived from the expert's knowledge base, depicts the situation in which an expert would respond and perform set of specific tasks. To achieve the enhanced sustainability and efficiency, the CAiSH framework trained and evaluated on the updated datasets. In addition, CAiSH introduced a new concept of training process called the "DataWell", where the latest data sets are updated dynamically for the training and evaluation process. The CAiSH discards old historical data from the repository, which are more than seven days old. However, DataWell is flexible in nature and could be changed for the latest data retention duration based on use case scenario. This way the CAiSH framework always trained and evaluated on the new/fresh data sets and furthermore divided into the equal ratio of 1:1 for the target and training samples.

5.5 Comparision to existing IoT framework

In comparison to the existing IoT framework, such as MavHome, CASAS, and iDorm, CAiSH provide a component-based architecture to provide a dynamic ambient intelligent IoT framework. The existing framework including iDrom, Domus, CASAS and Siri are designed for static data mining task where a system is trained on historical datasets without any periodic change in the data pool. In contrast, every framework considers no change on inhabitant activity pattern over the period. Where CAiSH integrated the ”DataWell” to refresh the dataset over a period and provide the new data point for model training and testing. However, IBM Watson is follow the slightly different approach to embrace the change in inhabitant activities in Cognitive IoT environment. It covers a wide range of cognitive computing research while we are more focused on ambient intelligent smart home for a single inhabitant. On the other hand, few states of art product such as Panasonic, Samsung, Siri are ready to use IoT devices but rather being proactive they are reactive in nature. In fact, the intelligence part is missing in ready to use IoT devices. In to context of current research challenges, a low cost dynamic intelligent IoT system is required where IoT sensor can be embedded in the environment and intelligent machine learning models can train IoT system frequently to adapt the change in inhabitant activity patterns.

5.6 Conclusion

The approach of the CAiSH framework is to provide a unified architecture by integrating ACM and AEM model components as a complete framework for the cognitive ambient intelligent smart home environment. The identification of Inhabitants activity states and adaptation of the regression models to forecast ADL patterns provides fundamental building blocks to the CAiSH framework. Therefore, the CAiSH framework understands the situation in surrounding space and adapt them into a regression model to perform task execution in a proactive manner. Hence, CAiSH proposes a novel architecture and contribution to modern intelligent IoT systems. However, the viability of the CAiSH framework is not limited to the smart homes and it can also be successfully applied in other use case scenarios of smart spaces including the health informatics, smart city, smart transport, production units and the other IoT projects.

The contribution to the existing IoT research has been delivered with the submission of journal paper in IEEE consumer electronic, title as " A Cognitive Ambient-Intelligent Smart Home". The review process has been benefited with the critical suggestion and further modification in the journal paper in an iterative manner. The suggested reviews on paper ensure the credibility of research work and ensure the better quality of research paper.

CHAPTER 6

Prototype Implementation

6.1 Background

This chapter meets the fourth research objective of implementing IoT prototype system for the CAiSH use-case scenario. The proposed approach is implemented on the time-series data sets. The data sets are collected from the three Arduino microcontrollers programmed with nine IoT sensors and deployed inside the one bedroom house. The experiment is the extension of the paper Jamnal et al. (2017) and Jamnal and Liu, (2017) research work, with the detailed implementations, are described in the chapter. The deployed IoT sensors with micro-controllers create a smart home environment for collecting real-time activity observations of the inhabitant. The deployment of CAiSH (Cognitive Ambient Intelligent Smart Home), provides the investigation platform for predictive analysis of real-time series data sets in the smart home scenario. At present, in the other research work, datasets are not captured in a cognitive manner where collective effort and contextual knowledge is almost impossible to extract and inappropriate for the further predictive analysis. Therefore, CAiSH prototype ensures the cognitive IoT system to combine the atomic level activity information in a collective manner to extract the higher level contextual knowledge of inhabitants activity intentions and situation in the IoT environment. The system trained on the regression models of supervised machine learning algorithms including the HMM (Hidden Markov Model), NB (NaiveBayes) and the RNN (Recurrent Neural Network) and perform a comparative analysis of prediction accuracy of each training models. The training and evaluation process are successfully accomplished in the MATLAB environment.

6.2 CAiSH (Cognitive Ambient-intelligent Smart Home): IoT smart lab deployment

The CAiSH framework embraces the core functionality of ACM and AEM as a building block to construct an ambient intelligent cognitive IoE (Internet of Everything) environment. Inhabitant activities at the spot and intentional level are systematically captured with calibrated IoT sensors in well-programmed microcontrollers to ensure data consistency and quality standard. The proposed CAiSH framework is applied to a one bedroom apartment located in Edinburgh, UK. The activities are oriented on a single individual in single occupancy flat. The real time-series data about the inhabitant's daily activities routines observed over a 3 days(12/01/18-14/01/18) period of time. The inhabitant agreed to have his personal behavioral data to be collected and evaluated for the purpose of training and validating machine learning algorithms to forecast the ADL patterns. The data from all sensors are collected in a central repository system developed on Raspberry Pi3, 1.2 GHZ quad-core ARM Cortex A53 1 GB LPDDR2-900 SDRAM, 802.11n Wireless LAN accessed over the Internet using MQTT(Message Queuing Telemetry Transport) tunnel. The collected data is then used for supervised machine learning experiment to identify inhabitants situation and intention in the surrounding environment and forecast maximum likelihood probabilities of ADL routines.

Prior to deployment, the privacy concerns have been considered to avoid ethical constraints. The individual is a healthy adult with no disability. In addition, the beneficial features of the Arduino sensors do not reveal the inhabitant's identity. Particularly, as compared to the camera-based observations, Arduino sensors are more useful to protect privacy and confidentiality of the inhabitants. As a result, data collection method based on micro-controller and IoT sensors, experience a higher acceptance rate by the people, compared to the camera surveillance sensors. The surface area is covered in CAiSH test bed is 500 sq.ft including the bedroom, living room and the kitchen area. As per figure 6.1, a comprehensive inspection of the flat has been considered to have a complete understanding of areas obstacles, walls, ceilings, and household objects to understand the environment for IoT sensors deployment purpose.

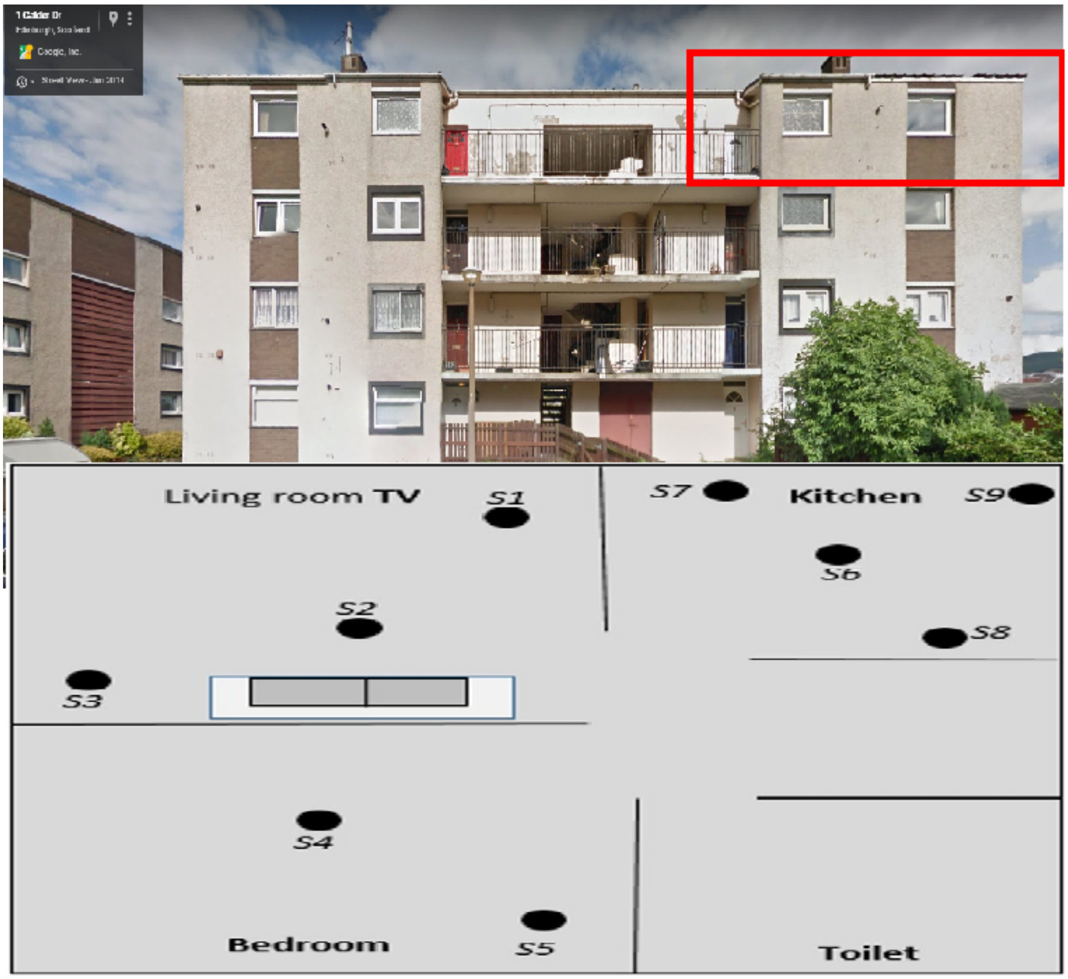


Fig. 6.1 Floor Plan of the CAiSH Deployment

6.3 Behavioral Routines Observation

Raw data from the multiple sensors in the prototype deployment is interpreted and described as states of semantic meaning, that serve as input to the learning system. As per table 6.1, the individual sensor capture data on the default embedded frequency and generates data log in analog/digital formats. The micro level sensor's Spot Activity (SA) observations and the complex level Intention Activity (IA) states information provide a cognitive meaning of inhabitant daily living activities routines. As per figure 6.2, the time interval have been set to per 30 seconds to capture activity observation datasets from the four type of ambient sensors including the force resistor -pressure, photo resistor-light, PIR-motion and the thermostat-temperature. The collected activities observation data sets represent the inhabitants ADL routines inside the home. Examples of such activities of daily routine are the following :

Table 6.1 Specifications of Sensors and Data Type

Sensor	Frequency (GHz)	Data Type	Measuring Range
Photo-resistor	2.4	Analog	1-500
Force-sensitive Resistor (FSR)	1.4	Analog	100-1024
PIR motion	1.8	Digital	0 or 1
Temperature	2.0	Analog	100-900

- Sleeping in the bedroom
- Working in the bedroom
- Laying on the bed and studying book
- Relaxing in the living room
- Doing household work/ exercise in the living room.
- Studying-working in the living room.
- Cooking in the kitchen.
- Doing cleaning or meal preparation work in the kitchen.
- Reading in the bedroom
- Reading in the living room
- Not present at the house

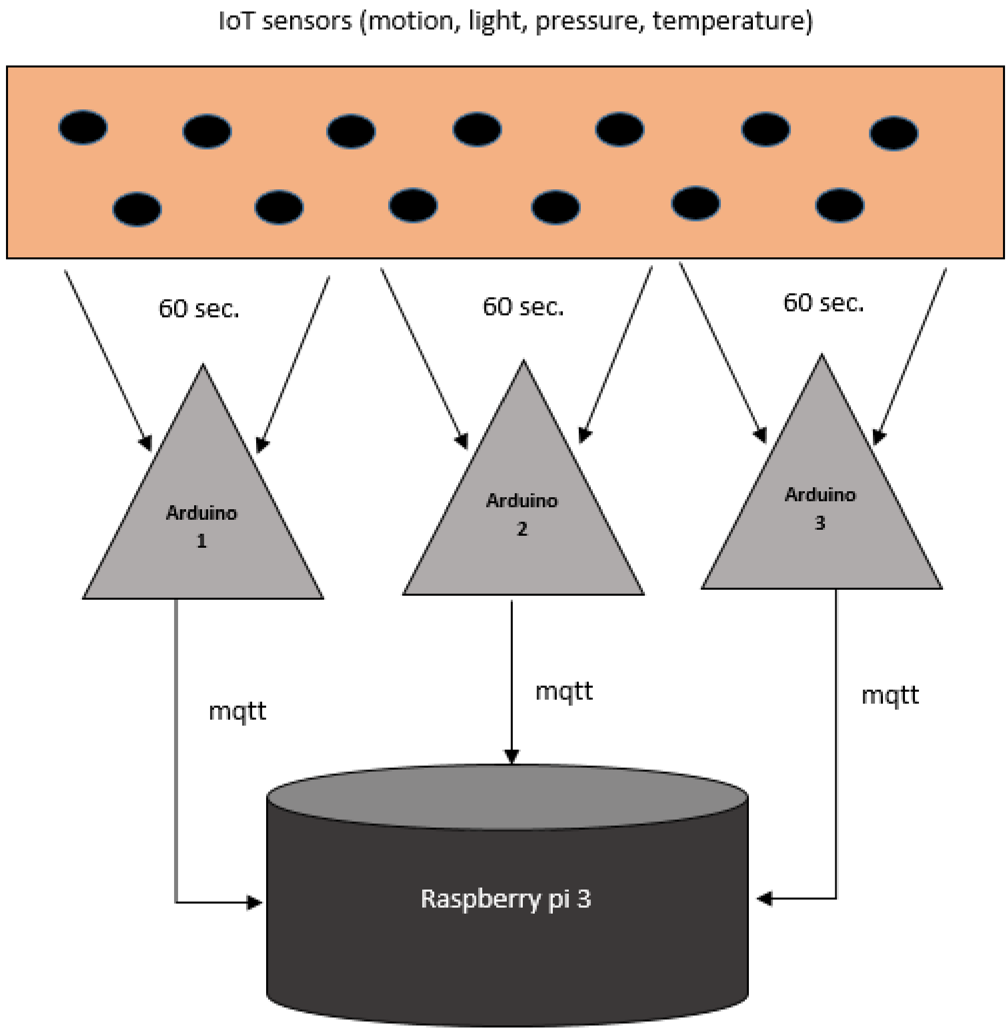


Fig. 6.2 Data Collection Interval (Per 30 Seconds)

6.4 IoT micro-controllers and sensors

In nowadays IoT technologies, micro-controllers are self-contained systems with peripherals, memory and a processor that are used for embedded IoT systems. The programmable micro-controllers are designed to be used for embedded applications, unlike microprocessors that can be found in PCs. As a result, micro-controllers are being used in automated controlled devices including power tools, toys, implantable medical devices, office machines, engine control systems, appliances, remote controls and other types of embedded systems. Most programmable micro-controllers that are used nowadays are embedded in other consumer products or machinery including phones, peripherals, automobiles and household appliances for computer systems. Due to that, another name for a micro-controller is "embedded controller." Some embedded systems are more sophisticated, while others have minimal requirements for memory and programming length and a low software complexity.(*futureelectronics2017* n.d.)

6.4.1 Arduino Uno Wi-Fi microcontroller

Arduino is an open-source programmable micro-controller platform based on easy-to-use hardware and software. As per figure 6.3, Arduino boards are able to read inputs such as the light, humidity, motion, air quality, pressure, object distance and the temperature in surrounding environment and respond them to activate or deactivate electrical appliances as the output. Arduino can be easily programmed by writing a set of coding-functions on the SDK (Software Development kit) interface through the object-oriented programming language (based on Wiring), and then the processing logic. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping for IoT embedded system research. As soon as it reached to a wider community, the Arduino board started to adapt to new needs and challenges. Arduino differentiates its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and other smart embedded peripherals. Particularly, Arduino boards are completely open-sourced platform, empowering researcher community to build independently and eventually adopt new development packages for the particular needs. The software IDE is open-source, and it is growing rapidly through the contributions of worldwide users. (arduino, 2017) and (*futureelectronics2017* n.d.)

6.4.2 Characteristics of Arduino Uno Wi-FI

- **Economical in Nature** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the assembled Arduino modules cost less than 50 dollars.

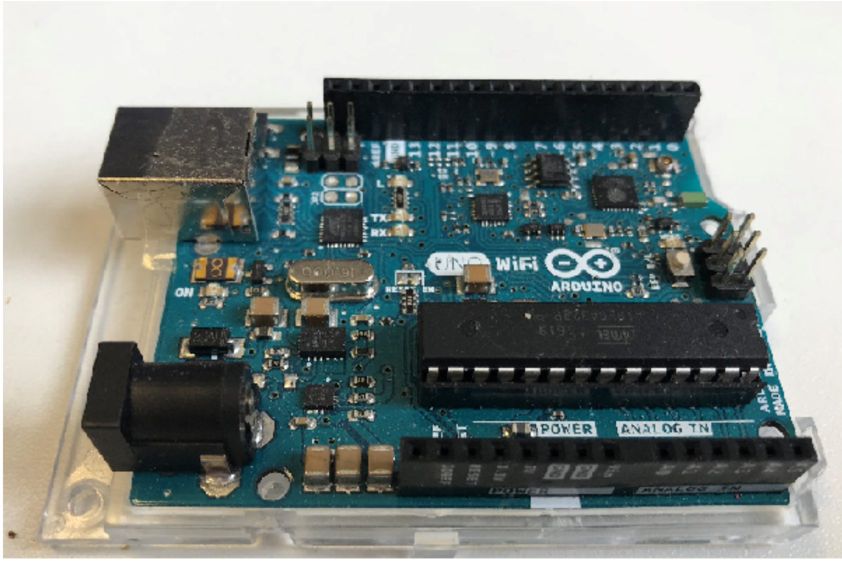


Fig. 6.3 Arduino Uno WiFI Board with Analog and Digital Pins

- **Cross-Platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems, whereas other micro-controller systems IDE are limited to Windows or Linux.

- **Easy Interface** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For researchers, ,it conveniently fulfills the research requirement and processing programming environment.

- **Open Source** - The Arduino software is published as open source tools, available for extension by the experienced programmers. The language can be expanded through C++ libraries, and help researchers to understand the technical details can make the leap from Arduino to the AVR C programming languages.

- **Extensible Hardware** - The plans of the Arduino boards are published under a creative commons license, so experienced circuit designers can make their own version of the module and extend and improve the current module. Even the relatively inexperienced researchers can build the breadboard version of the module in order to understand how it works and could save research budgetary funds.

6.4.3 MQTT : Data transmission protocol

MQTT (Message Queuing Telemetry Transport) is a lightweight, fast communications protocol designed for the Internet of Things research. It has its origins at IBM (where it was originally developed by Andy Stanford-Clark), and it has since been submitted to the organization for the advancement of structured information standards (OASIS). The current version of the protocol standard is v3.1. The MQTT V3.1 protocol specification states that it is a lightweight broker-based publish/subscribe messaging protocol, and ensure the simple, lightweight and easy implementation for the research and development projects. The "easy to implement" part has certainly proven to be true, as several different libraries implementing MQTT clients have been developed. The MQTT is perfect for use in Arduino Uno WiFi due to its asynchronous, with multiple different levels of quality of service, which is important in cases where Internet connections are unreliable to sends short, tight messages that make it handy for low-bandwidth situations.
(?)

6.5 Real Time-Series Data Collection from Heterogeneous IoT Sensors

In this study, inhabitant activities are classified into various states based on time-series data log observations from the three CAiSH deployments. With our knowledge base and experience about the inhabitant's life style, the placement of each CAiSH has been carefully considered to capture/ observe inhabitant's activity patterns obstructively in the living room, bedroom, and the kitchen areas. Therefore, the CAiSH deployment emphasizes to ensure the less constraint and better comfortableness in the inhabitants ADL routines. The data sampling rate is currently set to per 30 seconds intervals to generate and capture ADL data log for the predictive analysis. As per figure 6.4, the total number of 5499 Spot-Activity (SA) and Intention-Activity (IA) are captured over 3 days of the period from the cognitive IoT environment. The collected time-series SA and IA data sets are divided into equal (50:50) ratio such that a total number of data set 5499 are equally divided into 2390 training sets and 2392 testing sets for accuracy evaluation purpose in the MATLAB.

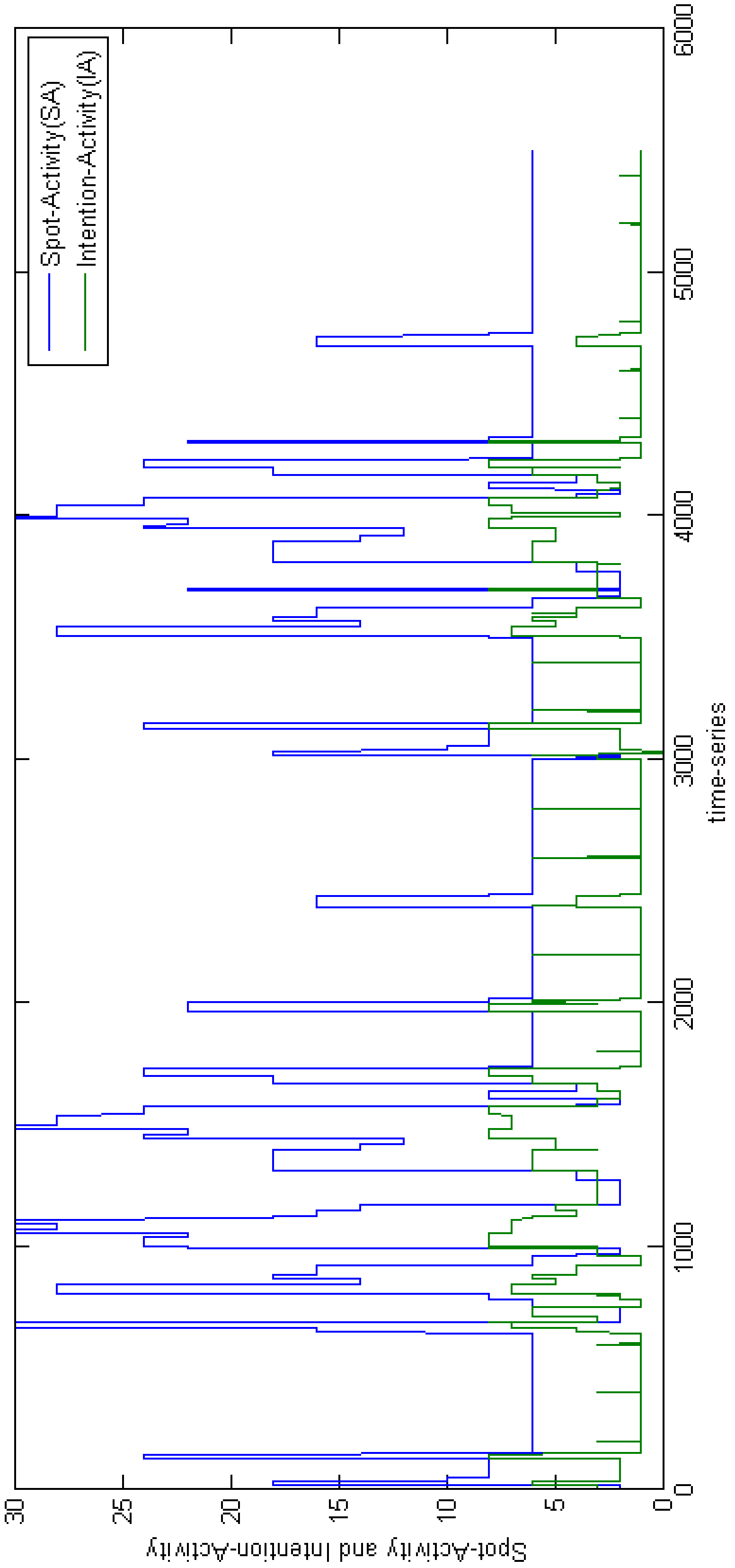


Fig. 6.4 Total Data Sets Collection for SA and IA States

6.5.1 CAiSH (1) Deployment in the Bedroom Area for Activity Observations

To start with the deployment of CAiSH 1 in the bedroom area has been presented in figure 6.5 with the capability to capture ambient parameters of pressure, motion, luminosity level via using force-resistor, PIR and photo-resistor sensors respectively connected through MQTT protocol. The ACMs statistical approach has been applied on the time-series data log to identify and label activity states . The series of Spot-Activity (SA) vector$< SA = (sa_t, sa_{t+2}, sa_{t+3}, ...sa_n) >$ and Intention Activities (IA) vector $< IA = (ia_t, ia_{t+1}, ia_{t+2}, ia_{t+3}, ia_t) >$ represents time-series activity log of per 30 second intervals(t_i). The analog and digital values of individual sensors have preprocessed through the ACM threshold(th_i) and weight schemes (W_j).

6.5.2 CAiSH (2) Deployment in the Livingroom Area for Activity Observations

In the second deployment of CAiSH 2 in the living room area, figure 6.6 shown the placed prototype in appropriate location of the room. The ambiance parameters captured on per 30-second interval including motion, luminosity level and pressure through PIR, Photo-resistor and force-resistor sensors respectively. As mentioned in the previous deployment, every spot level activities are captured and further preprocessed to infer inhabitant's intentional activity and situation through ACM statistical model approach. The sensor readings are captured in Vector(V_n^t) , where $V_n(SensorsReading)$ and $t(timeinterval)$.

6.5.3 CAiSH (3) Deployment in the Kitchen Area for Activity Observations

Furthermore, activities performed in the kitchen area are captured by the CAiSH 3 deployment in figure 6.7, with ambient parameters such as temperature, luminosity level and motions using a thermostat, PIR and photo-resistor sensors respectively. We used thermostat

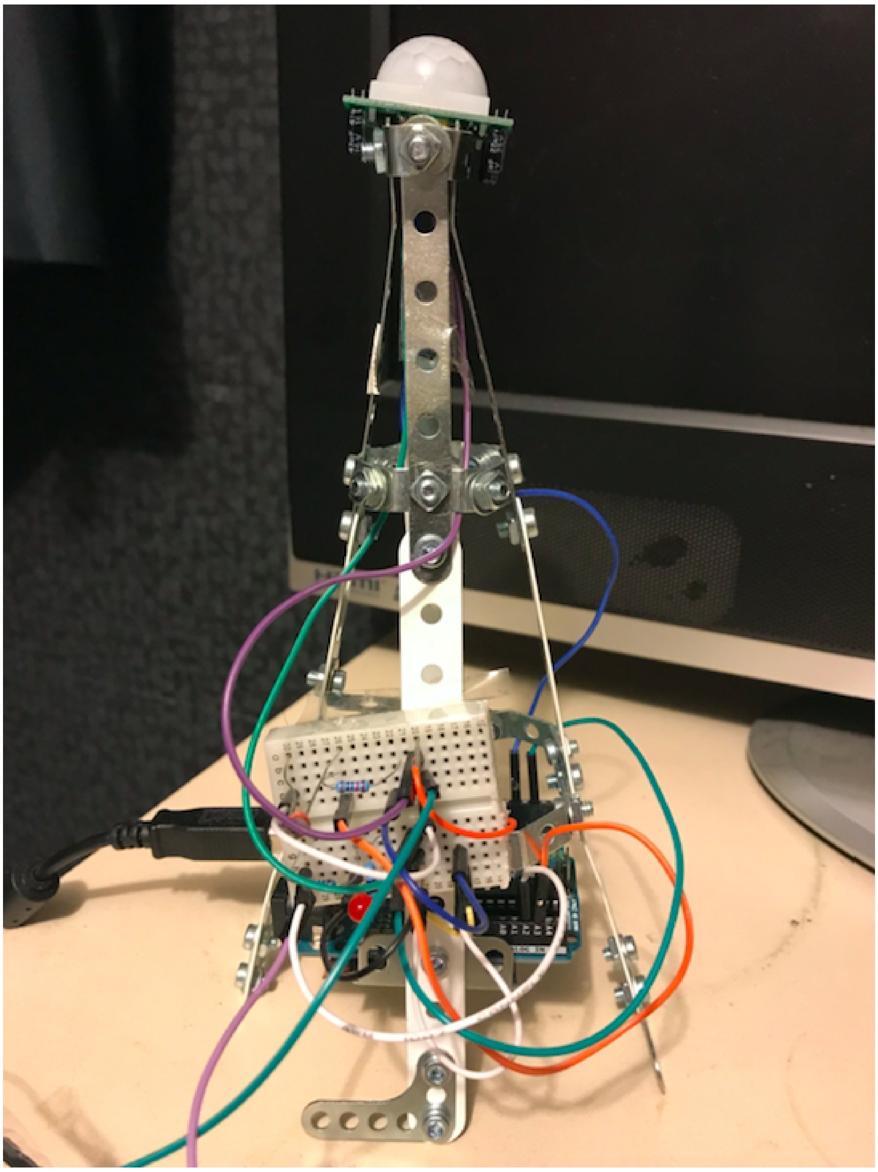


Fig. 6.5 CAiSH 1 Deployment in the Bedroom Area

sensor to observe if inhabitant performing some cooking related activity nearby hop or gas cookers by capturing surrounding temperature level. While PIR and photo-resistor sensor observe momentary activity and luminosity level in the kitchen area respectively.

In total, 30 Spot-Activity (SA) states and 8 Intention-Activity (IA) states have been discovered and investigated in the research according to the ACM model definitions of (SA_i) Spot-Activity and (IA_i) Intention Activity state definition.

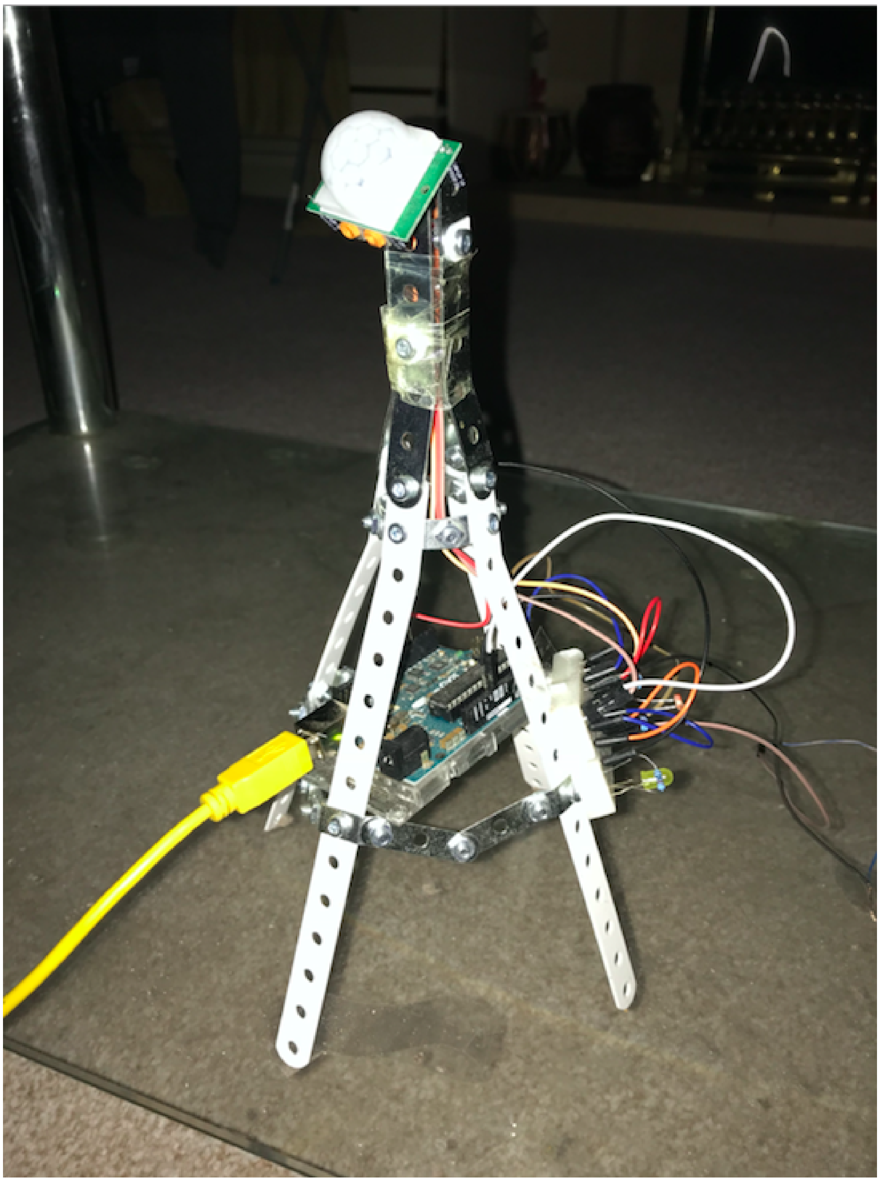


Fig. 6.6 CAiSH 2 Deployment in the Livingroom Area

6.6 Conclusion

This section outline the implementation of CAiSH prototype to observer the inhabitant ADL (Activity Daily Living) patterns in a time-series manner. Also, the development process of the prototype has been comprehensively inspected with its technical details and viability to capture ambient parameters in the smart home environment. The various aspect of micro-controllers and IoT sensors have been considered including their appropriate use for privacy and confidentially concerns. The collected time-series datasets provide

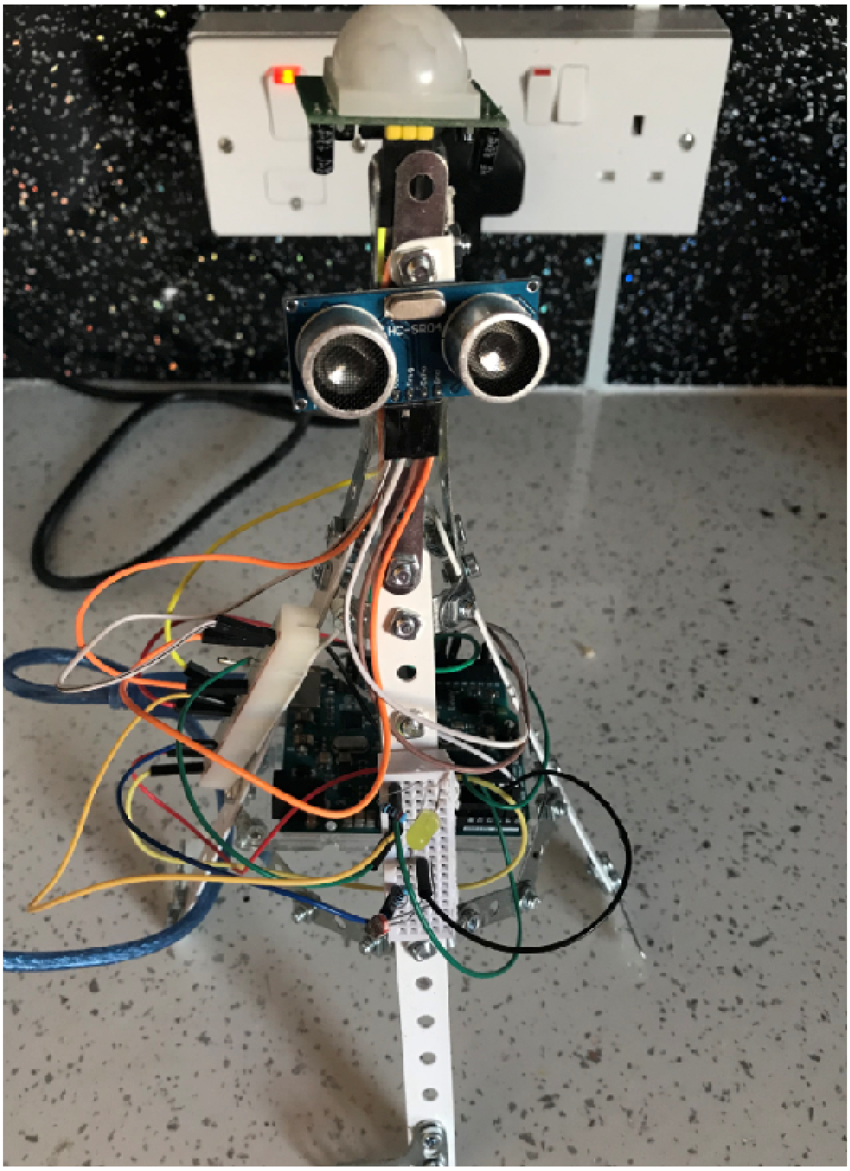


Fig. 6.7 CAiSH 3 Deployment in the Kitchen Area

research ground for the further machine learning predictive analysis experiments and investigations to enhance living experiences for inhabitants in the assisted and independent living environments.

CHAPTER 7

Training and Evaluation

7.1 Time-Series Forecasting in the CAiSH Framework

This chapter presents the summary of the training process and standard quantitative measures taken against predicted outputs of the individual machine learning training models. The activity identification and predictions are evaluated based on the dataset collected from CAiSH (Cognitive Ambient Intelligent Smart Home) prototype deployment. The regression models of supervised machine learning methods are applied to forecast inhabitant ADL patterns. In addition, accuracy measures are considered to evaluate predicted outcomes of the system. Particularly, the measuring methods of MSE (Mean Square Error), MAPE (Mean Absolute Percentage Error) and the MAE (Mean Absolute Error) are used for the CAiSH framework evolutions. Furthermore, the supervised machine learning methods of HMM (Hidden Markov Model), ND (Nave Bayes) and the RNN(Recurrent Neural Network) have been described for CAiSH training process and their predicted outputs. The split for training and testing process has been performed by divided dataset set into equal (50:50) ratio. The evaluation of each machine learning model is performed against testing datasets to ensure their overall effectiveness and the accuracy. Furthermore, a breakthrough comparison between pattern recognition algorithms (HMM, NB, RNN) is performed in order to show the performance outcomes of training and testing process.

The various IA (Activity-Intentions and SA (Spot-Activity) states are obtained through ACM(Ambient Cognition Model), where data preprocessing and features extractions are performed via statistical model approach. The CAiSH framework is trained over 30 SA states and 9 IA states. The Hidden Markov Model (HMM), Recurrent Neural Network (RNN) and NaiveBayes (NB) supervised machine learning algorithms are trained to predict most likelihood IA states of the inhabitant. Also, a posterior probability matrix would be generated by HMM, NB and ANN algorithms for analyzing the individual (higher)probability of each IA states. Thereafter, the results would be compared with testing data sets to measure errors to evaluate forecast accuracy. The datasets were collected in less controlled and more free-living conditions.

7.2 Complexity in the Activity Identification

The implementation complexity of machine learning algorithms for training and testing process depends on the type and complexity of the data sets to be used for the training process. However, the complexity in data sets also depends on the number of activities, the types of activities, and the challenges to collect them in the cognitive manners. In particular, it is extremely important to understand the nature of data sets and the complexity in activities that before applying the machine learning models on them. Therefore, careful consideration for the number of activities, the types of activities, and the data collection method are vital to the system training and evaluation process.

- **Number of Sensors Deployed to Observer Activities**: Recognizing five activities is easier than recognizing twenty or fifty activities. As the number of activities increases, the identification method has to learn how to discriminate among a larger set of activities, which is usually harder. Discrimination is also harder if activities are similar to one another. Algorithms that recognize activities from a small set of sensors are easier to use in real-world applications and have lower computational requirements (since fewer sensor signals need to be analyzed) than algorithms that use large sets of sensory data for models training. As a result, fewer sensors usually ensure the technology to be deployed in more affordable and economically in the IoT systems.
- **Complexity in the Types of Activities to Identify**: Activities that are static in nature are easier to recognize than activities that are periodic in nature. Furthermore,

activities that involve different intensity levels are also harder to recognize because of their motion similarity in the feature space. Likewise, activities involving highly unconstrained motions impacted by objects in the environment are more difficult to recognize than periodic activities.

- **Complexity in the Data Collections**: Activities for which training and testing data are collected in laboratory settings are usually easier to recognize that activities for which training and testing data is collected in the free-living conditions. Subjects will usually behave differently and in less constrained ways outside of formal laboratory settings. Therefore, this work utilizes data collected for 30 Spot Activities(SA) and 9 Intention-Activities (IA) states in a single inhabitant, one bedroom flat with free-living conditions for the training purpose.

- **Sensors Deployment to Ensure Unobtrusiveness in the Environment** : Sensors placement at appropriate locations in the apartment are more likely to be used for longer periods of time for data collections. However, sensors that might be integrated into existing clothing or devices already worn (watches, shoes) and carried (phones) are not appropriate for smart home experiments. As a result, data collection from as ECG or EMG monitors/sensors are inconvenient to the user and might be perceived as more difficult to use, uncomfortable and more burden to the user. Therefore, IoT sensors such as the PIR, Force-resistor, Photo-resistor, and the thermostats unobtrusively slipped into the environment and work invisibly without intervening inhabitant daily lifestyle.

7.3 CAiSH Components

The ACM (Ambient-Cognition Model) provides an elementary unit to the CAiSH (Cognitive Ambient Intelligent Smart Home) framework, through converting raw knowledge to contextual knowledge in the cognitive IoT environment. Subsequently, data preprocessing and appropriate labeling of activity states are the important factors for machine learning model to run intelligent algorithms experiments. Therefore ACM connects with the physical layer of IoT and appears as the first interface of the CAiSH framework. In contrast, the correlation between various associated Spot Activities(SA) and their occurrence in a cognitive manner have never been directly addressed for the ADL labeling before. The ACM applied threshold $< th_i >$ and weight $< W_i >$ schemes to avoid activities overlapping problem. while, Wen et al. (2015) used the global weight scheme, based on sensor events in different activities, considered the frequencies of individual sensor event rather than the temporal relationships among them. Whereas, ACM consider the temporal relationship among sensors during the weight assignment. However, Wen et al. (2015) applied a threshold for sensor events in order to avoid a negative effect on activity inference through false triggering. Whereas, ACM used thresholds to normalize raw analog values to the absolute digital ON(1)/OFF(0) states. In the ACM, IA (Intentional Activity) states are calculated based on individual sensor weight scheme and aggregated value of overall activity instance in the environment such that $< IA_1 = SA_1 * W_1 + SA_2 * W_2 + SA_3 * W_3 >$. While others, including Wen et al. (2015) have not applied a threshold on aggregated values, ACM successfully is able to differentiate between $< IA_k >$states such as ; $IA_1 > AgregatedWeight > IA_2$.

In addition, as the second elementary unit of CAiSH, the AEM (Ambient Expert Model) provides the forecasting capability of discrete Activity-Intension (IA) states in a time-series manner. The data mining process of HMM (Hidden Markov Model), NB(Naive Bayes) and RNN(Recurrent neural network) are applied to forecast the inhabitants most likelihood ADL patterns. The relevant association rule sets from the rule-based system are activated for proactive task execution in the smart home environment. However, existing adaptive models which combine the neural network like structure with fuzzy logic are limited in their ability to be adaptive to update the training parameters to forecast non-linear time series patterns. Whereas, the HMM is more effective use "Baum-welch" and "Viterbi" algorithms for parameters estimation and forecast patterns in time series manner with a higher probability that represents the system, than the data vectors that represent the minority scenario of the system (Hassan et al. 2013).

7.4 MatLab Experiments

In general, it is impossible to optimize all aspects (parameters) of an algorithm at once. This section presents a set of systematic experiments to determine the classifier, activity transitions probability, most likelihood pattern and performance evaluation results for the number of right and wrong prediction instances against real time series data sets. The experiments are performed in MatLab, which provides a high-performance scientific computing platform to perform experiments on various machine learning algorithms (packages) for statistical-predictive analysis, evaluations and provide advanced data visualization features. As per figure 7.1 , experiments are performed on collected 5499 time-series data sets from the CAiSH prototype deployment. In the Matlab for the training and evaluation process, datasets are divided into 1 : 1 equal ratio such that 2392 data sets for the training and 2390 data sets for the testing operations.

$$[TrainingSet, TestingSet] = divideblock(DataSets, .5, .5); \tag{7.1}$$

The supervised machine-learning experiments using HMM, NB and RNN are performed on the 30 SA (Spot Activity) and 9 IA (Intention Activity) contained in the CAiSH data sets. However, as per the table 7.1, some activities have more number of occurrence instances compared to other activities, are the part of inhabitant daily routines habits and do not have any direct impact on system training and testing process.

Table 7.1 Number of Instances and Occurrence Percentage of Activities

Intentional-State(IA)	ID	Number of instances	Percentage of time
working(bedroom)	3	611	10.59
study(livingroom)	6	376	6.25
study(bedroom)	2	372	6.45
working(kitchen)	8	378	6.55
sleeping(bedroom)	1	3343	57.93
relaxing(linvingroom)	4	233	4.04
Cooking(kitchen)	7	247	4.28
Working(kitchen)	8	211	3.66

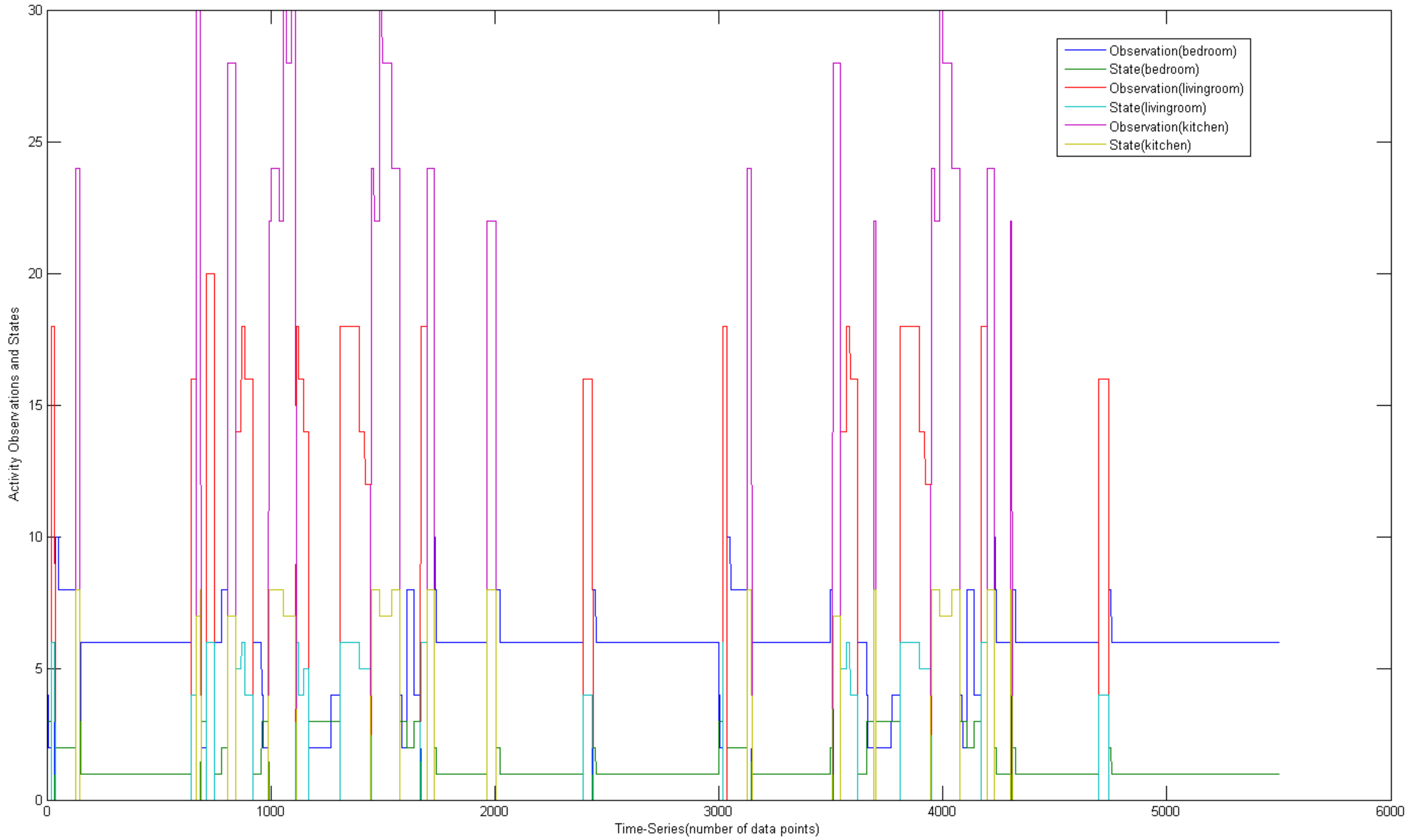


Fig. 7.1 The CAiSH Datasets of IA and SA States

The figure 7.1 represents the lineplots for collected CAiSH datasets of IA and SA states for model training and testing purpose. The each color shown the different IA and SA states. For stance, blue line plot shows the Observation(bedroom) and green lineplot shows the State(bedroom). The observation and state follow the SA and IA state respectively.

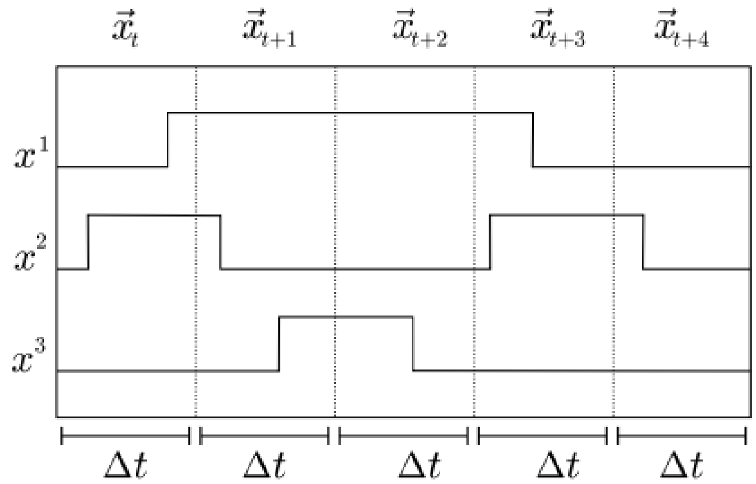


Fig. 7.2 Relation Between the Sensors Reading(x^i) and the Time Interval(Δ t).

The datasets are imported as a flat text file and converted into $(N \times M)$ matrix dimensions. The Spot-Activity (SA) has 2392 rows and 9 columns in $(2392{\times}9)$ dimensions, represent individual sensor values with fixed interval of time(t_i) at per 30 seconds. On other hand, Intentional-Activity (IA) collected and mapped as per the ACM architecture, represents the activity state with unique activity state id in 2392 rows and 1 column (2392×1) matrix dimension. The data set provides the research ground to identify the pattern and forecast inhabitant activity states using supervised machine learning methods of HMM, RNN and NB. The training output is evaluated for their accuracy and effectiveness for each method within the Matlab simulation tools and visualized for appropriate comparison result outputs.

$$ObsData = TargetObsData$$
$$StData = TargetStateData$$

7.4.1 HMM (Hidden Markov Model) for Time Series Forecasting

To start, $hmmestimate$ provides the one of two method for Hidden Markov model parameter estimation for emissions and states transition matrix . As per the given observation sequence of inhabitant activity(Obs_i) and situational intention activities (Sts_j), $hmmesimate$ method calculate the transition and emission probabilities matrix for parameters estimations with the function ;

$$[TRANS, EMIS] = hmmestimate(Obs_i, Sts_j) \tag{7.2}$$

Where $Obs_i = SA_k$ and $Sts_j = IA_l$

With the application of Baum-Welch algorithm, $hmmestimate$ calculates the maximum likelihood estimation of the transition (TRANS) and emission (EMIS) probabilities matrix , provided observation sequence (Obs_i) with known intentional states (Sts_j) of activities .

$$Obs_i = TrainingSet(1, :) \tag{7.3}$$

$$Sts_j = TrainingSet(2, :) \tag{7.4}$$

The TrainingSet consist data in $(n \times m)$ dimensions, having first row as the Observations (Obs_i) sequences and second row contains the States sequences (Sts_j). The two parameters are fed to the "hmmestimate" method in order to generated "transition" and "emission" matrix using the "baum-welch algorithm".

On the other hand, using expectation-maximization (EM) algorithms, without knowing the Intentional Activity states (Sts_a) data sets, the system can obtain $[Trans, Emis]$ transition and emission probability matrix with $hmmtrain$ function. To calculate the Hidden Markov Model parameters, estimated from the Observation sequence of (Obs_i) and initial guess probability of $[Tran_{guess}, Emis_{guess}]$ transition and emission matrix, the function applied as ;

$$[ESTTR, ESTEMIS] = hmmtrain(Obs_i, Tran_{(guess)}, Emis_{(guess)}) \tag{7.5}$$

The "hmmtrain" estimates the transition and emission probabilities for HMM using the Baum-Welch algorithm. The Observation sequences (Obs_i), are row vectors containing single sequences, a matrix with one row per sequence, or a cell array with each cell containing a sequence. While the $Tran_{(guess)}$ and the $Emis_{(guess)}$ represents the initial estimates or guess of transition and emission probability matrix. The $Tran_{(i,j)}$ represent the estimated probability of transition from state i to state j and $Emis_{(i,k)}$ represents the estimated probability that symbol k is emitted from state i. In particular, " hmmtrain (...,'Algorithm',algorithm) " specifies the training algorithm and can be either 'BaumWelch' or 'Viterbi' algorithm. However, the default algorithm is 'BaumWelch' used for the experiments. As per figure 7.3, shown the HMM training output results compared to the testing datasets.

HMM Most Probable State Path

Once the, parameter estimation achieved, the task of calculating most probable Hidden pattern of activity-state is accomplished by the $hmmviterbi$ function. The "Viterbi" algorithm is designed to find the most likelihood state sequences $O = (a_1, a_2, ..., a_T)$. It is sufficient to maximize for a fixed observation sequence :

$$O_{t+1} = argmaxP(O_{T+1}|O_1, O_2, ..., O_T, \lambda) \tag{7.6}$$

The function, $STATES = hmmviterbi(SeqObs(I_a), TRANS, EMIS)$
Given Sequence Observation $Obs_i = (o_1, o_2, ..., o_i)$, hmmviterbi function calculates the most likely path through the HMM specified by transition probability matrix(TRANS) and emission probability matrix (EMIS). In particular, the function $hmmviterbi$ begins with the model in state 1 at step 0, prior to the first emission such that $hmmviterbi$ computes

the most likely path based on the fact that the model begins in state 1.

HMM Posterior State Probabilities

The $hmmdecode$ function represents, the HMM posterior state probabilities of maximum likelihood activity-state pattern. The function :

$$PosteriorSTATES = hmmdecode(Obs_i, TRANS, EMIS) \tag{7.7}$$

It calculates the posterior state probabilities, PSTATES of the observation sequence (Obs_i), from a hidden Markov model. As per table 7.4, posterior state probabilities are the conditional probabilities of being at state *k* at step *i*, given the observed sequence of symbols. The model uses transition probability matrix(TRANS), and an emissions probability matrix(EMIS), where $Trans_{(i,j)}$ is the probability of transition from state i to state j and $Emis_{(k,seq)}$ is the probability that symbol seq is emitted from state k. The PSTATES is an array with the same length as observation-sequences and one row for each state in the model. The$(i, j)^{th}$ element of PSTATES gives the probability that the model is in state i at the j^{th} step, given the observation-sequence(Obs_i). The higher probability, ensure the most likelihood activity state of hidden pattern in the HMM. The function$hmmdecode$ begins with the model in state 1 at step 0, prior to the first emission and computes the probabilities in PSTATES based on the fact that the model begins in state 1.

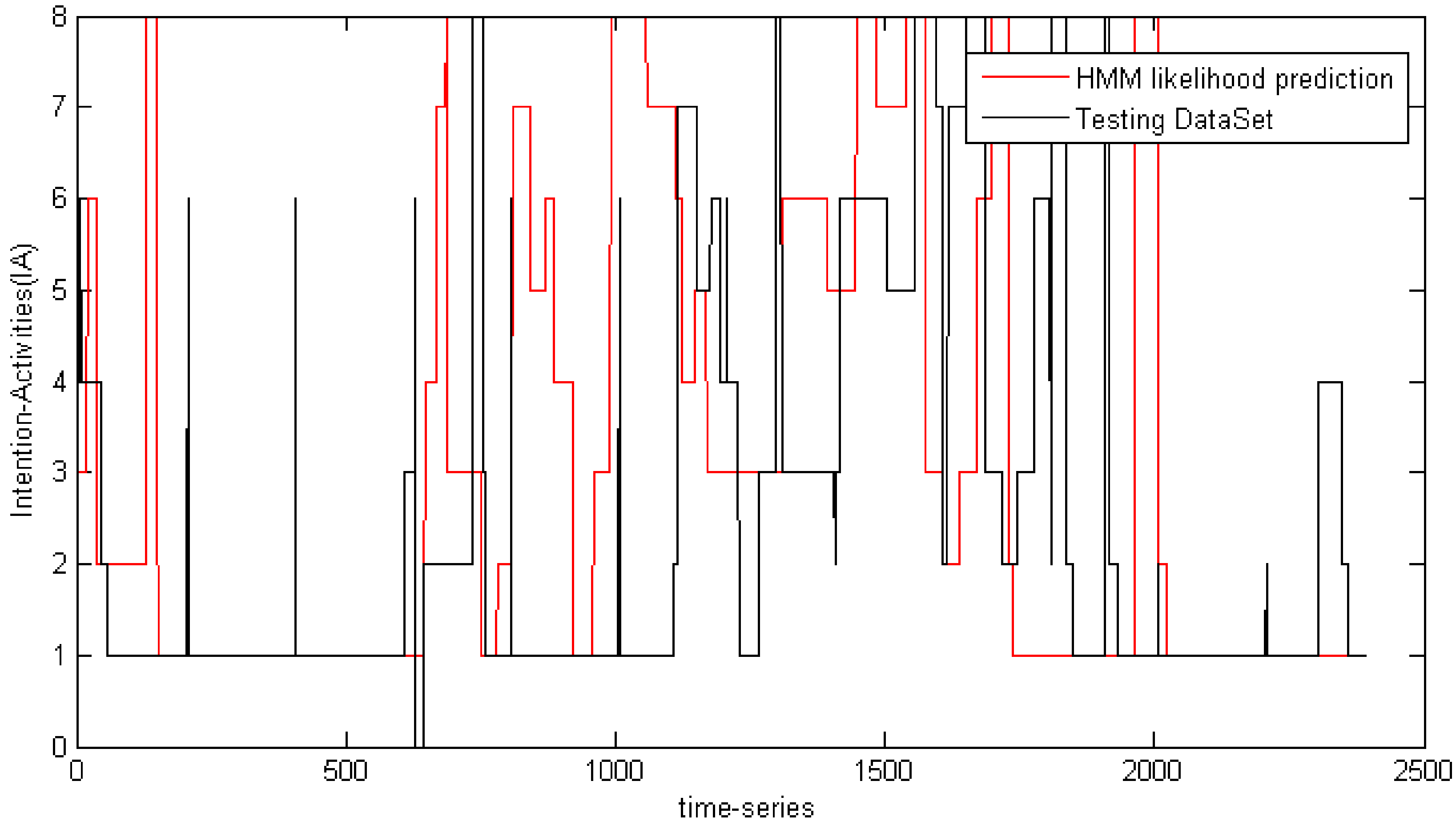


Fig. 7.3 HMM Likelihood Patterns in Comparison with the Testing Dataset

mmpost <2392x8 double>

1	2	3	4	5	6	7	8
0	0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	1.5385e-08	0	0
0	0	1.0000	0	0	6.9691e-08	0	0
0	0	1.0000	0	0	2.6137e-07	0	0
0	0	1.0000	0	0	9.3792e-07	0	0
0	0	1.0000	0	0	3.3259e-06	0	0
0	0	1.0000	0	0	1.1755e-05	0	0
0	0	1.0000	0	0	4.1505e-05	0	0
0	0	0.9999	0	0	1.4651e-04	0	0
0	0	0.9995	0	0	5.1715e-04	0	0
0	0	0.9982	0	0	0.0018	0	0
0	0	0.9936	0	0	0.0064	0	0
0	0	0.9773	0	0	0.0227	0	0
0	0	0.9197	0	0	0.0803	0	0
0	0	0.7167	0	0	0.2833	0	0
0	0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0	0

Fig. 7.4 HMM posterior Probability Results Output

The figure 7.4 describe the posterior probability results of various IA states from 1 to 8. The higher probability value of each IA state is opted out as maximum likelihood states.

7.4.2 Naive Bayes (NB) for Time Series Prediction

A Naive-Bayes object defines a Naive Bayes classifier. A Naive Bayes classifier assigns a new observation to the most probable class, assuming the features are conditionally independent given the class value. Naive Bayes classifier is based on applying Bayes' theorem with strong independence assumptions among the features. Naive Bayes classifier assumes the attribute conditional independence for a given category, for all attributes are independent of each other. Bayesian classifier classifies the test data into a class with the highest probability. As per figure 7.5 shown the NB result output compared to testing data sets.

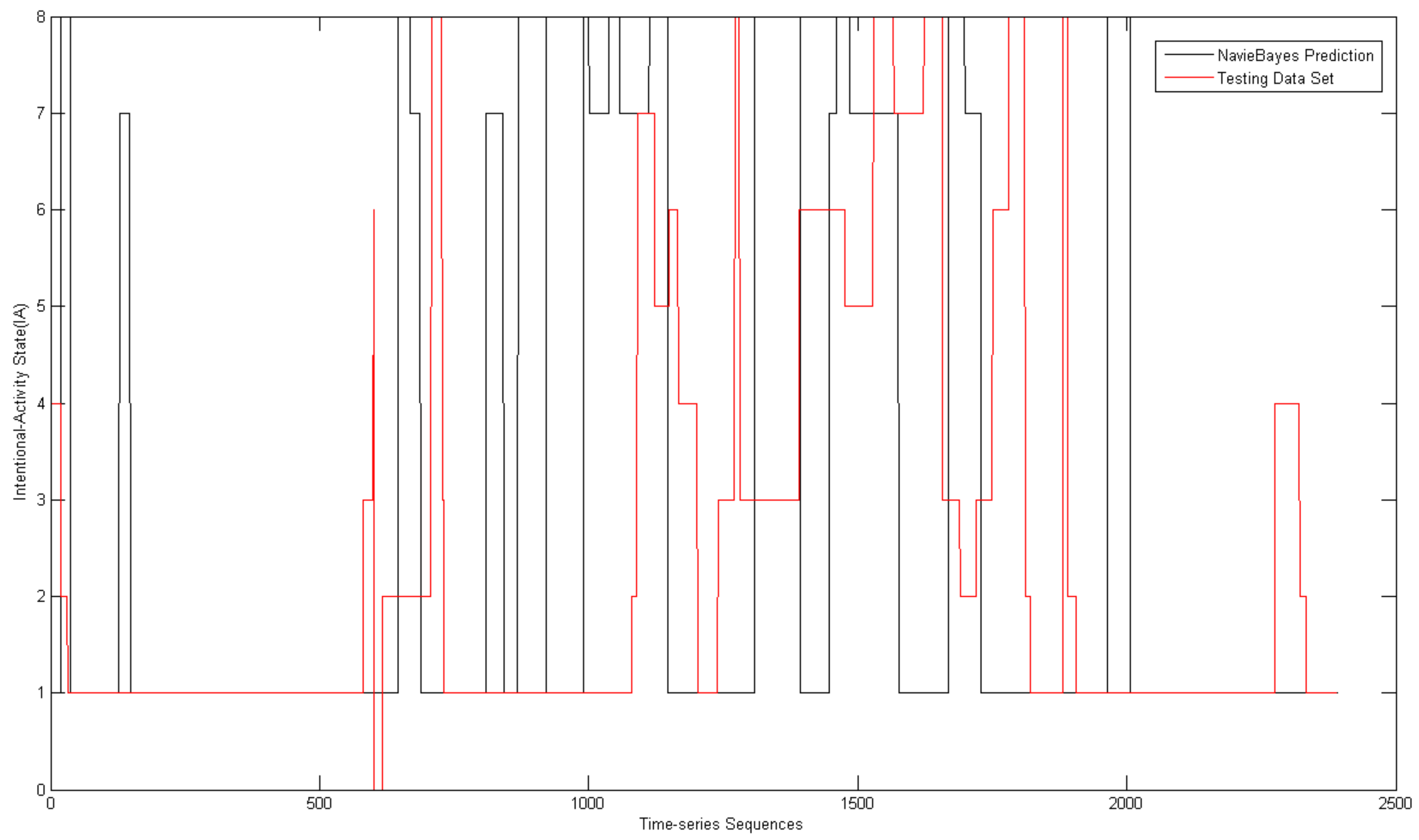


Fig. 7.5 Naive Bayes Prediction Results in Comparison with Testing Dataset

The naive bayes function, $nb = NaiveBayes.fit(Observation, State)$ builds a NaiveBayes classifier object nb, where training process is performed on $[N \times D]$ dimension matrix of the training data sets. The $NaiveBayes.fit$ function trains a Naive-Bayes classifier. The trained NaiveBayes classifiers store the training data, parameter values, data distribution, and prior probabilities related model information to predict labels or posterior probabilities for the new data. The $nb = NaiveBayes.fit(Observation, State)$ returns a multi-class naive bayes model (nb), trained by the predictors and use the default Gaussian distribution and a confusion matrix:

$$nb = NaiveBayes.fit(Observation, State); \tag{7.8}$$

$$nbprediction = nb.predict(Observation); \tag{7.9}$$

$$nbprediction = predict(nb, Observation) \tag{7.10}$$

The $nbprediction = nb.predict(Observation)$; classifies each row of data in test into one of the classes according to the NaiveBayes classifier nb, and returns the predicted class level .

$$[nbposterior, nbprediction] = posterior(nb, Observation) \tag{7.11}$$

$$nbconfmat = confusionmat(State, nbprediction) \tag{7.12}$$

In addition, $[nbposterior, nbprediction] = posterior(nb, Observation)]$ returns the posterior probability of the observations according to the NaiveBayes object (nb). As shown in figure **??** rows of test correspond to points, columns of test correspond to features. Therefore, post is a $[N \times n]$classes matrix containing the posterior probability of each observation for each classes. The post(i,j) represents the posterior probability of point i belonging to class j.

nbpost <2392x8 double>

	1	2	3	4	5	6	7	8
1	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
2	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
3	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
4	0.7432	0.0584	0.0827	0.0214	0.0012	0.0537	0.0058	0.0336
5	0.7432	0.0584	0.0827	0.0214	0.0012	0.0537	0.0058	0.0336
6	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
7	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
8	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
9	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
10	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
11	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
12	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
13	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
14	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
15	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
16	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
17	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
18	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
19	0.6852	0.0683	0.1110	0.0248	6.5581e-04	0.0678	0.0050	0.0372
20	0.0382	0.1007	0.1419	0.0892	0.1453	0.1245	0.1495	0.2107
21	0.0382	0.1007	0.1419	0.0892	0.1453	0.1245	0.1495	0.2107

Fig. 7.6 The Naive Bayes Posterior Probability Result Output Matrix

7.4.3 Recurrent-Neural Network(RNN) for time series predictions

The recurrent-neural network (RNN) is a class of artificial neural network, it provides the ability to work on time series datasets and allow the learning of time-based dependencies. Whereas, the feed-forward neural network is not suitable for a prediction on time series data sets. In each subsequent cycle of training, the outputs of the previous cycle are used as inputs in RNN. Each layer in an RNN has a recurrent link with a delay related to it. The RNN model is embodied in the form that the network will remember the previous information and apply it to the current output calculation. The nodes between the hidden layers are connected, and the hidden layer's input includes not only the output of the input layer but also the output of the hidden layer at the last time interval. Each recurrent neuron receives two sets of inputs, the actual input vector u and also the output vector y from the previous step, therefore weights for RNN learned by training in the relation

$[y(t), u(t) \rightarrow y(t+1)]$. In figure 7.8 shown the RNN training response with error and other validations. The future values of a time series y(t) are predicted only from past values of that series. This form of prediction is called ”nonlinear autoregressive”, or NAR and written as follows :

$$y(t+1) = W_2 g(W_1 y(t) + \delta u(t) + b)$$

(Agrawal and Muchahary 2018)

In contrast, the ”trainlm” is a network training function that updates weight and bias values according to ”Levenberg-Marquardt” optimization. The $net.trainFcn =' trainlm'$, ”trainlm” is often the fastest back-propagation algorithm in the MatLab toolbox, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms. As figure 7.7 shown the RNN result output in comparison with Testing Datasets.

In particular, the ”net.trainFcn = ’trainlm” function sets the network ”trainFcn” property and the ”$[net, tr] = train(net, ...)$” function trains the network with ”trainlm” parameter.

Algorithm 3 Recurrent Neural Network for time-series data

```
targetSeries = CAiSHdataset;
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays,hiddenLayerSize);
[inputs,inputStates,layerStates,targets] = preparets(net,,,targetSeries);
net.trainFcn = ’trainlm’;
net.performFcn = ’mse;
[net,tr] = train(net,inputs,targets,inputStates,layerStates);
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);
view(net);
```

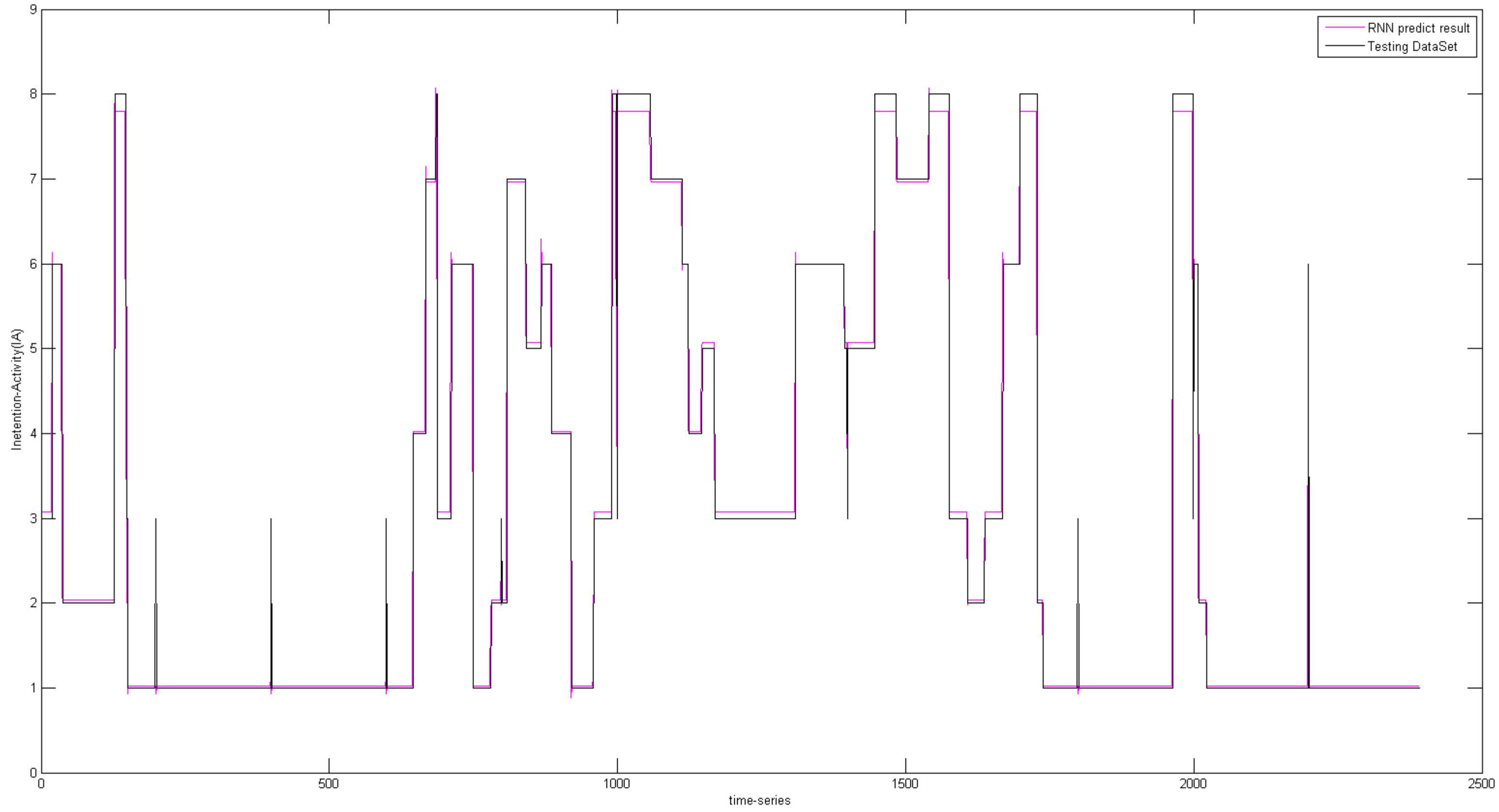


Fig. 7.7 The RNN Response Results using the Non-Linear Auto Regressive Model

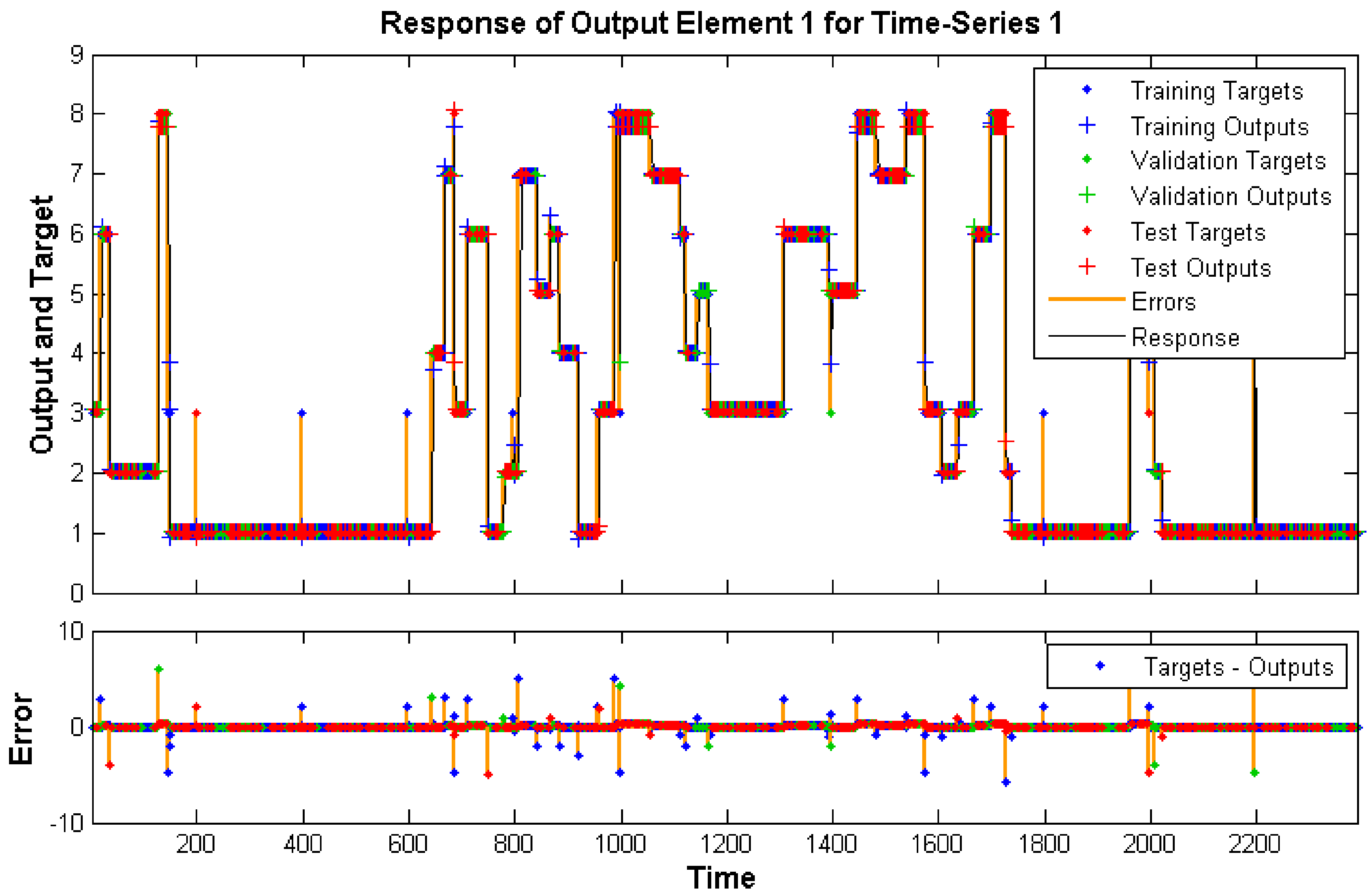


Fig. 7.8 The RNN State Response Result Output

7.5 Forecast Accuracy Measures

Furthermore, the time-series data sets have been used to extract the potential knowledge in various fields such as production industry, economics, finance and social trend observations etc. The prime focus on time-series predictive analysis includes trend analysis, similarity search, pattern mining and patterns forecasting, although forecasting time-series is the most important and popular area of current research in data mining. However, the forecasting would bring error inevitably based on three drawbacks or error in system training process. The first one is induced by indeterminacy of the problem, which can not be eliminated. The second one is caused by half backed messages/missing/inappropriate datasets due to lack of enough information collected. Finally, the last one induced by choosing unsuitable algorithms during training process. Rongling et al. (2009)

In the accuracy evaluation, If the predicted state is equal to the actual state, it is believed that the method correctly forecasts the user state otherwise, it infers that the method made a wrong prediction. Let $< X_i = x_1, x_2, x_3, ..., x_N >$ be a time series to be forecasted and $\Psi = \{\psi_i\}, i = (1, 2, ...N)$ is corresponding forecasted value of X. Therefore, $\kappa = \{\psi_i - X_i\}, i = (1, 2, ...N)$ is the error sequence in the time- series. A lot of useful information for evaluation of the relationship between x_i and ψ_i in the error sequence κ, it is commonly used to evaluate forecasting algorithms. There are many metrics to measures forecasting error such as the Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and the Mean Absolute Error (MAE) to show the accuracy results.

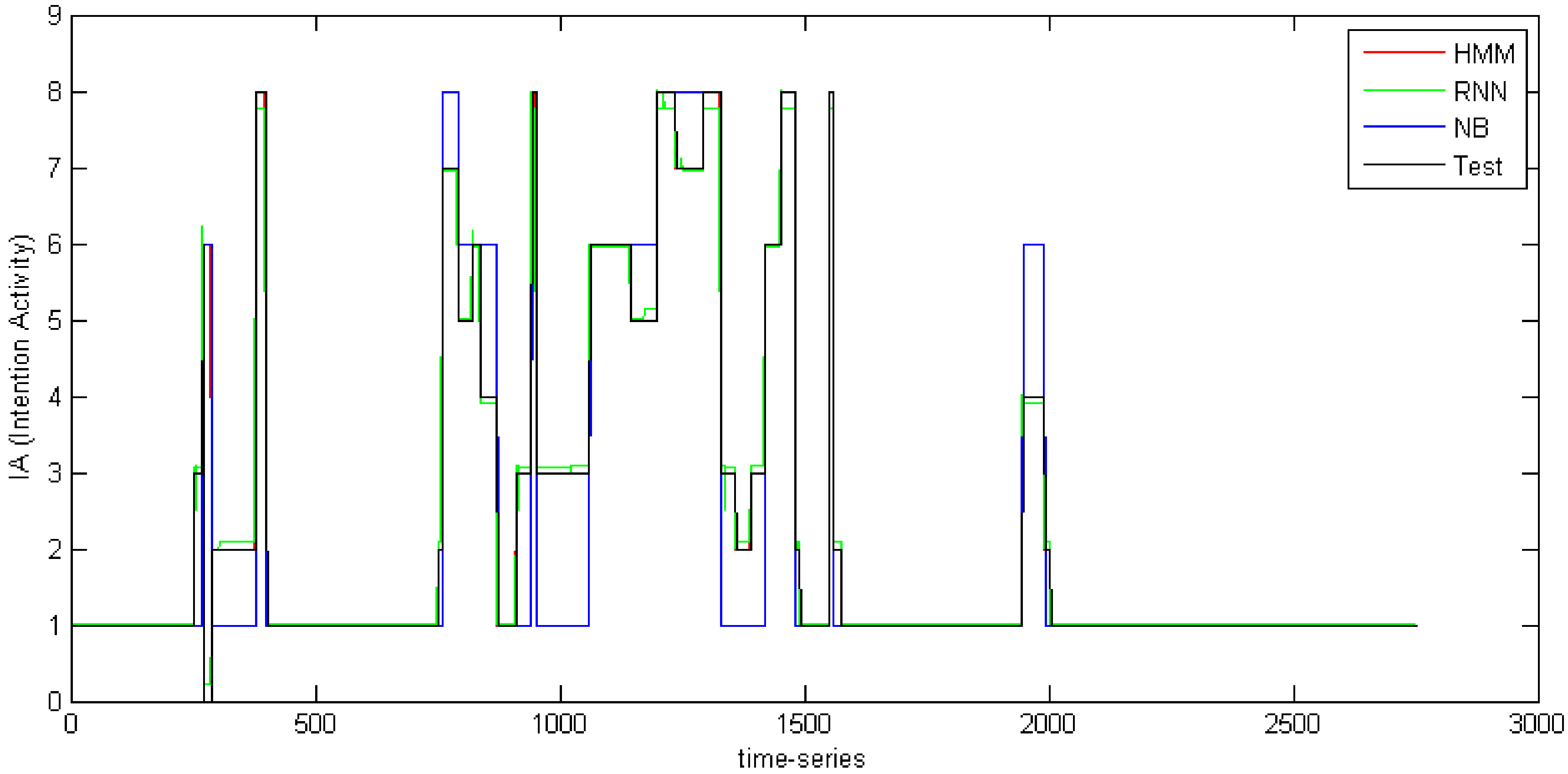


Fig. 7.9 The RNN, HMM and NB prediction Results in Comparison with the Testing Dataset

7.5.1 Mean Squared Error (MSE)

There are no acceptable limits for MSE except that the lower the MSE the higher the accuracy of prediction as there would be an excellent match between the actual and predicted data set. The CAiSH framework trained over the three supervised machine learning algorithms to forecast the time series Intention-Activity (IA) states. As divided the data sets into an equal ratio (1:1), it makes training and testing process easier and well organized for evaluation purpose. The proposed framework of CAiSH is designed and developed for single inhabitant activity recognition. The CAiSH framework is trained to predict the most likelihood state of hidden activity pattern with maximum posterior probability. As a result, MSE (mean square error) method applied to calculate the difference between Actual (Y_i) dataset and predicted $(\hat{Y}_i)$ dataset of similarity to ensure overall accuracy. The time-slice accuracy represents the percentage of correctly labeled time slices data points. Currently, the length of time slice t is set to 30-second intervals. The metric of the Mean Squared Error is defined as follows.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2. \tag{7.13}$$

Table 7.2 The result form MSE comparison

method	results error
HMMprediction	0.33
RNNprediction	0.15
NBprediction	0.77

The experiments performed on the Hidden Markov Model, Recurrent Neural network and Naive Bayes classifier with the same training data sets but the prediction output from each algorithm are different to each other. As per the MSE comparison table 7.2, HMMprediction is more viable with lower MSE value(15.60) compare to RNNprediction and NBprediction MSE values. Alongside, a recurrent neural network for time series data also performed well with lower MSE value(15.95) but HMMprediction results are more suitable for the evaluation. The table 7.2, present the three different results for 2900 predicted data points, evaluated against 2900 testing data sets.

7.5.2 Mean absolute percentage error (MAPE)

In addition, MAPE (Mean Absolute Percent Error) is a very commonly used metric for the forecast accuracy analysis. Since MAPE is a measure of error, high numbers are considered poor results and low numbers are acknowledged as the better results.The performance of the training model is compared quantitatively using the MAPE metric. The MAPE measures the size of the error in percentage terms as It calculate the average of the unsigned percentage error;

$$\mathrm{M} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|, \tag{7.14}$$

Where, A_t is the actual values and F_t is the forecast values, n is the number of observations (O_a) in the test data set. Agrawal and Muchahary (2018)

Table 7.3 The Result form MAPE Comparison

method	results error
HMMprediction	1.27
RNNprediction	2.83
NBprediction	11.16

The results of MAPE metric is the most common parametric measure for prediction accuracy analysis specifically in trend estimation tasks. It is commonly used when the amount by which numerical predictions are in error is evenly important. The MAPE is calculated using the absolute error in each period divided by the observed values that are evident during that period afterward average those fixed percentages. This approach is useful when the size or size of a predictor variable is significant for evaluating the prediction accuracy. MAPE indicates how much error in predicting compared with the real value. Khair et al. (2017) However, MAPE results are more viable for the RNN, with the low percentage of 2.032 compared to HMM and NB in the table 7.3. In some training purpose, RNN performs well in comparison to any other time series forecasting model but requires more computing resources during training process due to the complexity of the algorithm. While HMM and NB require less computing resources for training but also result in higher error rate compared to RNN.

$$Result.Mape = meanmat(abs(Test - Refernce)./Refernce) * 100;$$

7.5.3 Mean Absolute Error (MAE)

The simplest measure of forecast accuracy is called Mean Absolute Error (MAE). As the name suggests, MAE provides the mean of the absolute errors. The absolute error is the absolute value of the difference between the forecasted value and the actual value. MAE tells us how big of an error we can expect from the forecast on average. However, lower MAE ensures the lowest error in prediction.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |A_i - P_j|}{n} \tag{7.15}$$

Table 7.4 The Result form MAE Comparison

method	results
HMMprediction	0.01
RNNprediction	0.02
NBprediction	0.11

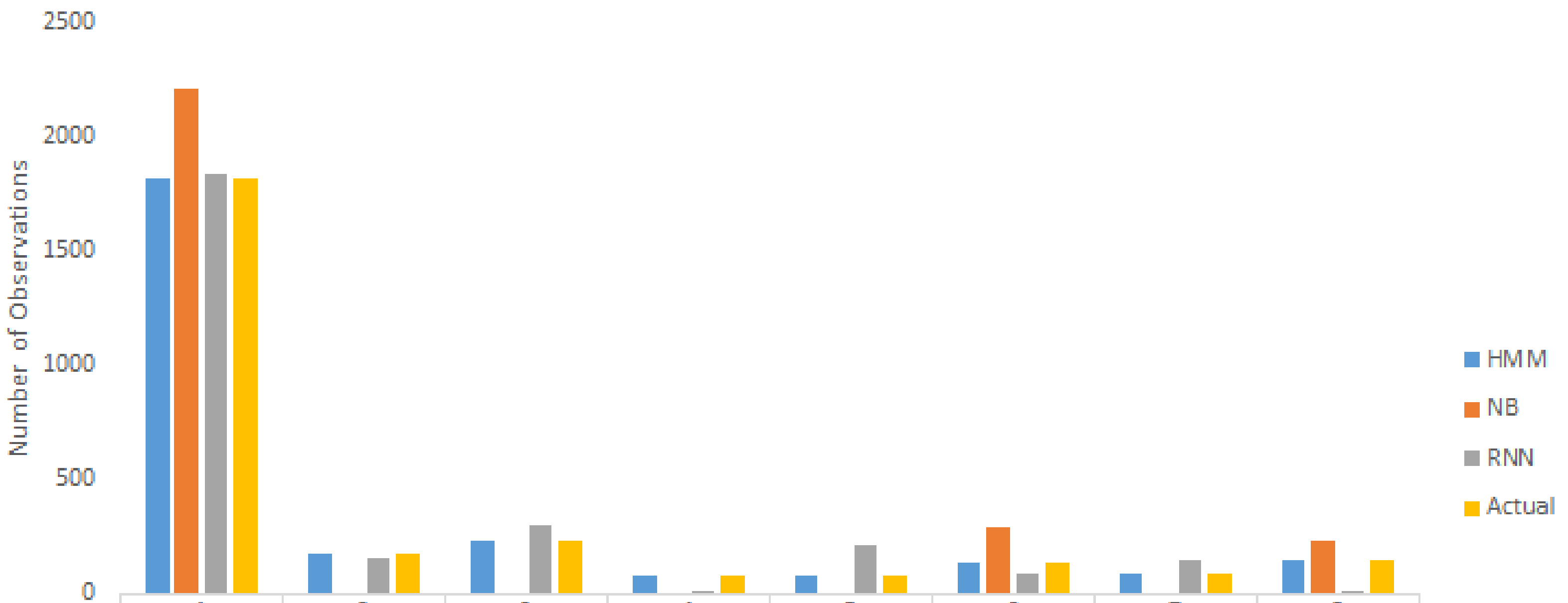
As per the MAE comparison table 7.4, HMMprediction results are more viable compared to RNN and NBprediction results. The HMMprediction results are lower at 3.0393 with a comparison to RNNprediciton results of 3.0840. However, the system trained over same data set and computing resources but evaluation results are better with HMM training model over RNN and NB training algorithms.

Overall, the training and evaluation processes of experiments are performed with three-fold cross-validation approach for HMM, RNN, and NB. The experiment focus is mainly to learn the behavior of HMM, NB and RNN training process in detail and compare their accuracy performances for different numbers of training iterations of the algorithms. As per the explanations, HMM has outperformed than NB, apart from some specific activities but RNN performed better compared to NB and has the similar output results like HMM. Although, training process of HMM required less computing resources as compare to RNN training process the training outcome was almost identical. The HMM gives an average accuracy as high as 96.7 percents compared to RNN which gives 95.5 percent. Furthermore, the table 7.5 compares the accuracies of HMM, RNN and NB for the individual activities and their corresponding values.

Table 7.5 Accuracy performance of HMM, NB and RNN with Comparison to Actual State

IA State	HMM	NB	RNN	Actual State
1	1814	2211	1835	1814
2	169	0	157	170
3	227	0	297	227
4	80	0	4	80
5	80	0	208	80
6	134	293	89	133
7	89	0	140	89
8	140	229	2	140

The table 7.5 describe the Actual IA states compared to the Machine learning models results. For instance, from 1 to 8 states denote the activities of sleeping in bedroom, working in bedroom, reading in bedroom, relaxing in livingroom, working in livingroom, reading in livingroom, cooking in kitchen and working in kitchen respectively. The machine learning model's of HMM, NB and RNN have been trained to produce results of forecasting inhabitant activity states. The first column represent various Intention Activity States and Second column represent the HMM forecasting results and third columns represent NB prediction outcomes and forth column represent the RNN predictions outcomes. The last column represent the testing data set as actual IA states for the comparison with various machine learning model outputs. In some cases, HMM predict better results compared to NB and RNN, while in some cases RNN performed better than others.



	1	2	3	4	5	6	7	8
HMM	1814	169	227	80	80	134	89	140
NB	2211	0	0	0	0	293	0	229
RNN	1835	157	297	4	208	89	140	2
Actual	1814	170	227	80	80	133	89	140

IA (Intention Activity) State

Fig. 7.10 Comparison of Accuracy Performance by HMM, RNN and NB against the Testing Dataset

7.6 Dynamic Training Pool (DATAWELL)

While many researchers have proposed a static IoT system for ADL patterns recognition to deliver personalized intelligent services for the smart home scenario, the CAiSH framework proposes, a recursive training and testing environment, where the system would be re-trained over a certain period with loopback. The changes in inhabitant's activity-patterns and lifestyles are non-linear in nature. A set of activities performed at specific durations(short/long) is dynamic in nature, causing a frequent change in the following activities sequences. The fundamental cause of such behavioral changes in inhabitant activities could be caused due to changes in environment or the physiological factors. In the CAiSH framework, IoT sensors collect data in the 24X7 environment in free movements of inhabitants for more natural, realistic and un-obstructive manners. The CAiSH works in a background process without affecting inhabitant daily habits and activity states, sensor readings generate a trajectory of real-time series activity observations for the research analysis.

In particular, the training and testing process of the CAiSH framework is iterative in nature. As per the system architectural design, a threshold $< th_{(HMM)} >$ can be defined to ensure the acceptable range of MAPE, MSE and MAE errors. The difference between actual and predicted activity sets provide the threshold $< th_{(hmm)} >$ value. However, due to the change in inhabitant life-style, the threshold value can be a shift(upward/downward) to retrain the CAiSH framework. At the moment, it has been set to the lowest MSE error rate (15.60). In addition, CAiSH introduced a new concept in the training data pool known as "DATAWELL". This is a segment of data that is updated dynamically throughout the training. The CAiSH discards old historical data, which is more than seven days old from the repository. This way the system always trained on new/fresh data sets. In the "DataWell" each data sample serves for the prediction targets and training samples.

In the essence to achieve realistic goals, if MSE value is less than defined threshold range, further training will not be required and rules will be invoked for task executions. Device controllers are mapped to the appropriate rule set to turn on and turn off the specific appliances to automate the process in a proactive manner. Whereas, if the MSE value is less than the threshold range, the system will be re-trained in order to achieve lowest error

Algorithm 4 Recursive Threshold for CAiSH Retraining

set : $th_{range} = 15.60$
$IF(MSE < th_{(range)})$
Retraining Not Required, Execute-Rules
$ELSE - IF(MSE > th_{(range)})$
Start Retraining

comparison rate to ensure higher accuracy to automate the smart home environment.

CHAPTER 8

Conclusion

This chapter presents a summary of the research findings and results, including the primary contributions with 3 publications and a discussion on how well the research aim and objectives stated in chapter 2 are met. In addition, some areas for the future work are identified and described.

8.1 Summary of Results

The existing IoT smart home systems are unable to understand the inhabitant's activity intentions and the situation in the surrounding environment in a proactive manner. In particular, the most IoT smart home technologies measures inhabitant's ADL (Activity Daily Living) routines without leveraging the contextual knowledge of inhabitant's intentions and a situation in the surrounding environment. The potential of ambient intelligence capabilities in IoT smart home systems to identify and understand Inhabitant's intention and the situation needs to be enhanced with artificial intelligent machine learning models. As a result, the development of ambient intelligent systems to infuse the contextual knowledge became vital to the current era of Cognitive IoT smart home research. The proposed framework of CAiSH is empowered with two intelligent and robust architecture of the ACM (Ambient Cognition Model) and the AEM (Ambient Expert Model), which runs un-obstructively in the background environment to capture micro-level information from IoT sensors and infuse into identifying IA (Intention Activity) states of inhabitant in a proactive manner for predictive analytics. The CAiSH framework capable of identifying the inhabitant's activity intentions at micro level activities and also provide the forecasting ability to predict inhabitant hidden ADL patterns in a time-series manner with maximum likelihood probability. The unified framework of CAiSH ensures a cognitive ambient intelligent smart home system to respond in a proactive manner to the inhabitant's intention and the situation in the smart home scenario. The research has successfully explored the trade-offs that needed to be made in order to achieve desirable goals.

8.2 Primary Contributions to the research aim

The thesis advances the state-of-art cognitive Internet-of-Things by improving various aspects of ambient intelligent smart home technologies, mechanisms, trade-offs and other technical constraints by integrating modern IoT technologies with machine learning models. The aim of developing CAiSH (Cognitive Ambient Intelligent Smart Home) framework has been successfully achieved with two robust architectures of ACM (Ambient Cognition Model) and AEM (Ambient Expert Model). The research findings enhance the assisted living experience and ensure the intelligence in IoT smart home systems to understand inhabitant's intention and the situation in a proactive manner. In summary, the main contributions of the research are:

8.2.1 First Publication

1. The proposed ACM (Ambient Cognition Model) provides a novel solution for the data preprocessing and activity identification problem. The ACM systematically transforms raw information form hetrogeneous IoT sensors into contextual knowledge to fulfill the primary task of activity identification over 30 SA (Spot Activity) and 9 IA (Intention Activity) states at the complex level. This approach allows the ACM to represent inhabitant's ADL (Activity Daily Living) routines in a higher contextual level and a more intuitive way to improve the ambient intelligence capability in the IoT smart home systems.

8.2.2 Second Publication

2. The architecture of AEM (Ambient Expert Model) focuses on the potential use of data science approach for the ADL (Activity Daily Living) routines predictions and forecasting inhabitant activity state intentions in the environment. This approach takes advantages of the ACM in order to use discrete ADL datasets for training-testing purpose and perform predictive analytic experiments with supervised machine learning models. It has been shown that predictive analytic capabilities to forecast time-series ADL patterns are different in the individual training process of HMM, NB and RNN. However, in terms of performance, resources, and efficiency, HMM results are more promising and effective for the IoT systems. Hence, the approach of AEM has been proved effective to forecast inhabitant's hidden ADL patterns in the Cognitive IoT environment.

8.2.3 Third Publication

3. The unified framework of CAiSH (Cognitive Ambient Intelligent Smart Home) is built upon the integration of ACM and AEM architectures adapting ability of inhabitant activities identification and forecast hidden ADL(Activity Daily Living) patterns in the cognitive IoT environment. The CAiSH framework performs as a personal digital assistant to the inhabitants by understanding their intentions and situations proactively in the environment. The rule-based system performs task execution through the appropriate rule activation to the identified and forecasted ADL patterns. Therefore, the CAiSH accomplish the task as an expert system to enhance assisted living experiences in the smart home scenario.

8.2.4 Prototype Development and Deployment

4. The development of IoT prototype for real-time data collection, enhance the learnings and research knowledge about the implementation challenges in IoT system using various micro-controllers and heterogeneous IoT sensors. The prototype provides ready-of-shelf and low-cost device for IoT smart home experiments with full access of source code, which make it even more flexible and effective for IoT developers to extract raw information

from the environment and build time-series datasets log of inhabitant's activity daily livings for further research investigations.

8.3 Revisiting the research Challenges and Gaps

This section briefly revisits the design goals stated at the beginning of this thesis and the evaluation measures used in most prior work to identify which ones were met and which ones were not met.

8.3.1 Complexity in Extracting Contextual Knowledge From IoT Sensors to Identify Inhabitant's Situations

Complexity in ACM Statistical Model for the Activity Identification The multivariate data sets of 30 spot activities states are explored in the thesis, including 9 intention activities with different intensity levels due to the number of instances in daily routine observations. The research also evaluates the recognition performance over prediction accuracy measures to evaluate the overall performance of the system.

Complexity in Data-Collection for the Activities Observations: The training data is collected from 3 micro-controllers and 9 IoT sensors, generating a total of 30 spot activities observations at a one bedroom residential apartment. In particular, the data collection at the residential flat is in the less constrained environment as the participant is allowed to perform the daily routine tasks in a free-living environment.

Threshold Determination: The ACM relatively more efficient using threshold techniques to identify activity-states compared to existing context-aware architectures. The existing research work applied a threshold on individual sensors states but not devised any techniques to determine overall activity intention states of the inhabitant. The ACM ensure the aggregated threshold to determine intention activity state of inhabitant and provide a discrete activity label for machine learning experiments.

Weight-Scheme for Sensors Precedence: The weight scheme application on sensors leverages the outcome of ACM with more advanced techniques in IoT smart home system. Individual sensor readings are vital to the environment to observe inhabitant activities in a continuous manner but every sensor has their different influence level compared to each other. The ACM embraced this challenge into the architecture and established the relationship between individual sensor ranking based on their precedence values to fill the current research gap.

Intrusiveness and Location Deployment of the Sensors in the Environment: The three micro-controllers at the bedroom area, living room area, and kitchen area are used to achieve the multivariate set of activities observations for data collections. The sensors at the locations can be easily relocated without hindering inhabitant daily activity routines and lifestyle.

Real-Time Data Interval: The data collection in a time-series manner provides the inhabitant's activity logs for machine learning experiments. In order to create discrete activity sets, the time interval has set up for a 30-second interval, which allow just-in-time data and real-time behavioral log for time series data sets.

GAP Fulfillment

The above-mentioned research challenges have been successfully embraced and addressed in the Ambient Cognition Model architecture. The ACM devised for the activity identification and data preprocessing. The problem of heterogeneous sensor data has been addressed by devising the unique thresholding and weight schemes in the ACM architecture. With the knowledge base about the IoT smart home, a threshold value and weight scheme are defined and have the flexibility to redefine or change further if required. Also, the deployment of CAiSH devices in smart home ensure the intrusiveness and do not hinder inhabitant to perform their ADL routines. As CAiSH devices observe the inhabitant activities and generate data in the 24X7 environment, so to control the large volume of data an interval for per60 seconds is applied. It ensures the data smoothness and captures all the vital information about inhabitant ADL routines.

8.3.2 Complexity in Predictive Models to Forecast Time Series ADL Patterns

Slicing Data Set into a Training and Testing Sets

The amount of data required for training depends on the complexity of the problem and on the complexity of a chosen algorithm. The factor of a number of classes, input features and model parameters play a vital role to fit in the training model. The data collection over 5500 SA (Spot Activity) and IA (Intention Activity) states provide the fundamental research ground for machine learning experiments. The data collection was large enough to yield statistically meaningful results and representativeness in training and testing sets to ensure similar characteristic. As a result, dataset split into equal ratio for the training and testing purpose.

Training Operations and Usability Factors Across Supervised Machine Learning Models and Results

The training process of individual algorithms depends on their parameter estimation techniques. The parameter estimation is relatively easier and automated in HMM and NB models while RNN requires explicit parameters (layers and neurons) which make the training process more complex and required more resources for computation. The experiments are performed for predictive regression analysis to determine the training output results.

Maximum Likelihood Activity States

The training results are provided in the individual IA (Intention Activity) states predicted maximum likelihood probabilities. The maximum likelihood ensures the higher state probability, the higher IA state probability chosen as compared to the lower probability. The training outputs of HMM, NB and RNN posterior probabilities are different in nature due to their internal algorithmic complexity and techniques.

Evaluation for Accuracy Measure in Training Models

The individual model has been tested for the training result outputs. The training results achieved a reasonable accuracy over 30 SA (Spot Activity) and 9 IA (Intention Activity) states prediction. The supervised machine learning algorithms (HMM, NB, RNN) required different parameters to perform training tasks. The AEM's predicted activity patterns are evaluated against testing dataset for accuracy measures. The forecast accuracy measure performed on MSE (Mean Squared Error), MAPE (Mean Absolute Percentage Error) and the MAE (Mean Absolute Error) metrics. The HMM, NB, and RNN have different error measure outcomes, however, the lesser is better error measure ensure the better results. In particular, the HMM proved to be more efficient with better results and require less computing resources during the training process.

Gap Fulfillment

The above-mentioned research challenges are successfully met into Ambient Expert Model and CAiSH architecture. The preprocessed datasets are used for the training and testing process. The data slicing in equal ratio(50:50) for training and testing sets ensure the effective supervised machine learning approach. In addition, the DataPool ensure the freshness and randomness in the datasets. The various supervised machine learning models(NB, RNN, HMM) are trained and tested for their effectiveness factor and among the best is selected for the deployment. In terms of computing resources and processing time, the Hidden Markov model turn out to be the best machine learning model and ensure the low error rate. The training result output is shown in a posterior probability matrix, which shows the individual probability of each Intentional Activity (IA) state. The IA with the highest probabilities is chosen for activity prediction. At last, the various performance metrics are applied to compare the individual ML model results for their effectiveness and accuracy measures.

8.4 Future Work

This section discusses the integration of wearable sensor, fuzzy logic learning and Blockchain technology, such that current research could be extended towards more future possibilities of robust, intelligent and secured cognitive IoT system. Hence, the focus would be on the key constraint factors to further enrich our understanding of IoT systems and contribute to the knowledge in the field of ambient intelligent systems.

Wearable Sensors

The advancement of nano-sensing technologies and miniaturization makes it possible to develop smart systems to monitor activities of human beings in unobstructed and continuous manner. The wearable sensors detect abnormal and unforeseen situations by monitoring physiological parameters along with other symptoms. Therefore, wearables sensor could provide the vital information about inhabitant wellbeing states in terms of monitoring several physical variables including motion, heart rates, blood pressure and body temperature etc. The information form wearables sensors could help to identify the inhabitant activity intention states and situation more accurately. In addition, this also could be indicated if inhabitant current states have any health hazard issue and a prompt action to avoid such situation.

Embrace Fuzziness

The fuzzy formulation of inhabitants ADL routines in IoT system would represent the enhanced version of ambient intelligent IoT system. The crisp information from sensors could be fuzzified to interpret inhabitant ADL patterns in the more realistic states. Therefore, the fuzzy logic could play an essential role in the development of human-like capabilities for ambient intelligent IoT system. The various SA (Spot Activity) and IA (Intention Activity) are fuzzy in nature, which would have a direct impact on identify hidden ADL patterns of the inhabitant. In addition, the rule sets for the proactive task execution could have updated human-like IF/ELSE conditions as fuzzy sets. Hence, the introduction of fuzzy logic would enhance the overall CAiSH framework effectiveness and ensure an improved experience for the assisted living environment.

Deploy Block-Chain Technologies

The future work, motivated by the increasing cyber crime instances and the recent explosion of interest in Blockchain technologies. The Blockchain technology allows having a distributed peer-to-peer network where non-trusting members can interact with each other without a trusted intermediary in a verifiable and secured manner. The future research would investigate certain issues that should be considered before the deployment of a Blockchain network in IoT systems right from transactional privacy to the expected value of the digitized assets traded on the network. The continued integration of Blockchain technology in the IoT systems would cause significant transformations across several industries including health informatics, smart cities, and pervasive computing projects. The Blockchain technology would bring a new business model to have reconsideration that how existing IoT systems and processes could be fine-tuned with higher security layer. (Christidis and Devetsikiotis 2016)

REFERENCES

Agrawal, R. K. and Muchahary, F. (2018), 'Long Term Load Forecasting with Hourly Predictions based on Long-Short-Term-Memory Networks'.

Banafa, A. (2017), 'IoT major challanges'.

Bourobou, S. T. M. and Yoo, Y. (2015), 'User activity recognition in smart homes using pattern clustering applied to temporal ANN algorithm', *Sensors (Switzerland)* **15**(5), 11953–11971.

Braun, A., Wichert, R., Eds, A. M., Conference, E. and Hutchison, D. (2017), *Ambient Intelligence*, Vol. 10217.

Burzagli, L. and Emiliani, P. L. (2017), 'Universal Access in HumanComputer Interaction. Design and Development Approaches and Methods', **10277**, 31–42.

Chen, H., Finin, T. and Joshi, A. (2003), 'An intelligent broker for context-aware systems', *Adjunct proceedings of Ubicomp* (January), 183–184.

Chen, W.-h. and Chen, Y. (2017), 'An Ensemble Approach to Activity Recognition Based on Binary Sensor Readings'.

Chen, Y., Zhou, J. and Guo, M. (2016), 'A context-aware search system for Internet of Things based on hierarchical context model', *Telecommunication Systems* **62**(1), 77–91.

Cho, S. B. (2016), 'Exploiting machine learning techniques for location recognition and prediction with smartphone logs', *Neurocomputing* **176**, 98–106.

Christidis, K. and Devetsikiotis, M. (2016), 'Blockchains and Smart Contracts for the Internet of Things', *IEEE Access* **4**, 2292–2303.

Cook, D. J., Augusto, J. C. and Jakkula, V. R. (2009), 'Ambient intelligence: Technologies, applications, and opportunities', *Pervasive and Mobile Computing* **5**(4), 277–298.

Fahad, L. G., Tahir, S. F. and Rajarajan, M. (2014), ‘Activity Recognition in Smart Homes Using Clustering Based Classification’, *2014 22nd International Conference on Pattern Recognition* pp. 1348–1353.

Forkan, A. R. M., Khalil, I., Tari, Z., Foufou, S. and Bouras, A. (2015), ‘A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living’, *Pattern Recognition* **48**(3), 628–641.

Friedewald, M., Da Costa, O., Punie, Y., Alahuhta, P. and Heinonen, S. (2005), ‘Perspectives of ambient intelligence in the home environment’, *Telematics and Informatics* **22**(3), 221–238.

futureelectronics2017 (n.d.).
URL: *https://www.futureelectronics.com/en/Microcontrollers/microcontrollers.aspx*

Gouin-Vallerand, C., Abdulrazak, B., Giroux, S. and Mokhtari, M. (2008), ‘Toward autonomic pervasive computing’, *Proceedings of the 10th International Conference on Information Integration and Web-based Applications and Services - iiWAS ’08* (c), 673.

Gouin-Vallerand, C., Abdulrazak, B., Giroux, S. and Mokhtari, M. (2010), ‘A software self-organizing middleware for smart spaces based on fuzzy logic’, *Proceedings - 2010 12th IEEE International Conference on High Performance Computing and Communications, HPCC 2010* pp. 138–145.

Graves, A. (n.d.), *No Title*.

Gu, T. (2004), ‘A middleware for building context-aware mobile services’, *Conference, 2004. VTC 2004-Spring. 2004* **5**, 2656–2660.

Gu, T., Wang, L., Wu, Z., Tao, X. and Lu, J. (2011), ‘A pattern mining approach to sensor-based human activity recognition’, *IEEE Transactions on Knowledge and Data Engineering* **23**(9), 1359–1372.

Guo, B., Zhang, D. and Wang, Z. (2011), ‘Living with internet of things: The emergence of embedded intelligence’, *Proceedings - 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, iThings/CPSCom 2011* pp. 297–304.

Hagras, H., Callaghan, V., Colley, M., Clarke, G., Pounds-Cornish, A. and Duman, H. (2004), ‘Creating an ambient-intelligence environment using embedded agents’, *IEEE Intelligent Systems* **19**(6), 12–20.

Hargreaves, T., Hauxwell-baldwin, R., Wilson, C., Hargreaves, T. and Hauxwell-baldwin, R. (2017), 'Benefits and risks of smart home technologies Bene fi ts and risks of smart home technologies', *Energy Policy* **103**(January), 72–83.

Hassan, M. R., Ramamohanarao, K., Kamruzzaman, J., Rahman, M. and Maruf Hossain, M. (2013), 'A HMM-based adaptive fuzzy inference system for stock market forecasting', *Neurocomputing* **104**(July 2016), 10–25.

Holmes, A., Duman, H. and Pounds-cornish, A. (2002), 'The iDorm : Gateway to Heterogeneous Networking Environments', *In International ITEA Workshop on Virtual Home Environments* **1**, 20–21.

Hong, J., Suh, E. H., Kim, J. and Kim, S. Y. (2009), 'Context-aware system for proactive personalized service based on context history', *Expert Systems with Applications* **36**(4), 7448–7457.

Huda, S., Yearwood, J. and Togneri, R. (2014), 'Hybrid metaheuristic approaches to the expectation maximization for estimation of the hidden markov model for signal modeling', *IEEE Transactions on Cybernetics* **44**(10), 1962–1977.

Hui, T. K., Sherratt, R. S. and Sánchez, D. D. (2017), 'Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies', *Future Generation Computer Systems* .

IBM (2017), 'No Title'.

International Energy Agency (2013), 'Energy Efficiency Market Report 2013', *ENERGY EFFICIENCY 2013 Market Trends and Medium-Term Prospects Market Report* .

Jurafsky, D. and Martin, J. (2017), 'Hidden Markov Models', *Speech and Language Processing* (Chapter 20), 21.

Këpuska, V. (2018), 'Next-Generation of Virtual Personal Assistants (Microsoft Cortana , Apple Siri , Amazon Alexa and Google Home)', (c), 99–103.

Khair, U., Fahmi, H., Hakim, S. A. and Rahim, R. (2017), 'Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error', *Journal of Physics: Conference Series* **930**(1).

Khalil, I. and Linz, A. (n.d.), *a Tlantis a Mbient and P Ervasive I Ntelligence S Eries E Ditor : I Smail K Halil*, Atlantis Press, Austria.

Lee, J., Lin, K.-j. and Science, C. (2010), ‘A Context Management Framework for Real-Time SOA’, pp. 559–564.

Lee, S.-h., Lee, S.-j. and Moon, K.-i. (2013), ‘An ANFIS Control System for Smart Home’, **5**(11), 464–470.

Liouane, Z., Lemlouma, T., Roose, P., Weis, F. and Messaoud, H. (2016), ‘A Markovian-based Approach For Daily Living Activities Recognition’, *The International Conference on Sensor Networks (SENSORNETS’16)* .

Liu, X. (n.d.), for Ambient Intelligent Smart Space.

Lotte, F. and Congedo, M. (2007), ‘A review of classification algorithms for EEG-based braincomputer interfaces’, *Journal of neural . . .* **7**(10), e44439.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P. and Sheth, A. P. (2017), ‘Machine learning for Internet of Things data analysis: A survey’, *Digital Communications and Networks* .

Manuscript, A. and Dysfunction, E. (2015), ‘NIH Public Access’, **25**(8), 713–724.

Mavropoulos, C. and Chung, P. T. (2014), ‘A rule-based expert system: Speakeasy - Smart drink dispenser’, *2014 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2014* pp. 1–6.

Memon, M., Wagner, S. R., Pedersen, C. F., Aysha Beevi, F. H. and Hansen, F. O. (2014), ‘Ambient Assisted Living healthcare frameworks, platforms, standards, and quality attributes’, *Sensors (Switzerland)* **14**(3), 4312–4341.

Moving average filter (2018).
URL: *https://www.gaussianwaves.com/2010/11/moving-average-filter-ma-filter-2/*

Nazerfard, E., Rashidi, P. and Cook, D. J. (2010), ‘Discovering temporal features and relations of activity patterns’, *Proceedings - IEEE International Conference on Data Mining, ICDM* pp. 1069–1075.

Negnevitsky, M. (2002), *Artificial Intelligence*.

Prieto, L. P., Rodríguez-Triana, M. J., Kusmin, M. and Laanpere, M. (2017), ‘Smart school multimodal dataset and challenges’, *CEUR Workshop Proceedings* **1828**, 53–59.

Ramos, C., Augusto, J. C. and Shapiro, D. (2008), ‘Ambient Intelligenceand#x02014;the Next Step for Artificial Intelligence’, *IEEE Intelligent Systems* **23**(2), 15–18.

Ricci, A. (2015), ‘Spotlight The Mirror World : Preparing for Mixed-Reality Living’, pp. 60–63.

Rish, I. (2001), ‘An empirical study of the naive Bayes classifier’, *Empirical methods in artificial intelligence workshop, IJCAI* **22230**(JANUARY 2001), 41–46.

Risteska Stojkoska, B. L. and Trivodaliev, K. V. K. (2017), ‘A review of Internet of Things for smart home: Challenges and solutions’, *Journal of Cleaner Production* **140**, 1454–1464.

Rongling, L., Yongle, L. and Qunfa, C. (2009), ‘Evaluation on forecasting algorithms of time series’, *Proceedings - International Conference on Management and Service Science, MASS 2009* (1).

Salovaara, A. and Oulasvirta, A. (2004), ‘Six modes of proactive resource management: a user-centric typology for proactive behaviors’, *In Proceedings of the 3rd Nordic Conference on HumanComputer Interaction NordiHCI* pp. 57–60.

Sasidharan, S., Somov, A., Biswas, A. R. and Giaffreda, R. (2014), ‘Cognitive management framework for Internet of Things: A prototype implementation’, *2014 IEEE World Forum on Internet of Things (WF-IoT)* pp. 538–543.

Satyanarayanan, M. (2001), ‘Pervasive computing: Vision and challenges’, *IEEE Personal Communications* **8**(4), 10–17.

Singh, R., Kainthola, A. and Singh, T. N. (2012), ‘Estimation of elastic constant of rocks using an ANFIS approach’, *Applied Soft Computing Journal* **12**(1), 40–45.

Stengler, J., Gaikward, G. and Hmida, H. B. (2015), ‘Towards the deployment of open platform AAL services in real life-advantages and lessons learned: uSmAAL: A case study for implementing intelligent AAL services in real life based on the open platform universAAL’, *ICT4AgeingWell 2015 - Proceedings of the 1st International Conference on Information and Communication Technologies for Ageing Well and e-Health* pp. 67–74.

Sun, F., Yu, X., Liu, S. and Lu, F. (2014), ‘Intelligent collaborative urban traffic management system based on SOA and cloud computing’, *26th Chinese Control and Decision Conference, CCDC 2014* (71171038), 1692–1695.

Taylor, R., Baron, D. and Schmidt, D. (2015), ‘The world in 2025 - Predictions for the next ten years’, *2015 10th International Microsystems, Packaging, Assembly and Circuits Technology Conference, IMPACT 2015 - Proceedings* pp. 192–195.

Taylor, R., Baron, D. and Schmidt, D. (2017), 'Main Menu The World in 2025 Predictions for the Next Ten Years Sessions Main Menu', pp. 358–361.

Technologies, A. and Change, S. (2014), *Ambient Assisted Living*.

TechTarget (n.d.).
URL: *http://whatis.techtarget.com/definition/time-series-forecasting*

Tennenhouse, D. (2000), 'Proactive computing', *Communications of the ACM* **43**(5), 43–50.

Trabelsi, D., Mohammed, S., Chamroukhi, F., Oukhellou, L. and Amirat, Y. (2013), 'An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression', *IEEE Transactions on Automation Science and Engineering* **10**(3), 829–835.

Tran, D. and Wagner, M. (1999), 'Fuzzy hidden Markov models for speech and speaker recognition', *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397)* pp. 426–430.

VanSyckel, S. and Becker, C. (2014), 'A Survey of Pervasive Computing', *UBICOMP,Computer Science* **3**(6), 1–7.

Vlacheas, P., Giaffreda, R., Stavroulaki, V., Kelaidonis, D., Foteinos, V., Poulios, G., Demestichas, P., Somov, A., Biswas, A. and Moessner, K. (2013), 'Enabling smart cities through a cognitive management framework for the internet of things', *IEEE Communications Magazine* **51**(6), 102–111.

Walek, B. and Janošek, M. (2014), 'Adaptive Fuzzy Control of Thermal Comfort in Smart Houses', pp. 675–678.

Wanglei and Shao, P. (2015), 'Intelligent control in smart home based on adaptive neuro fuzzy inference system', *Chinese Automation Congress (CAC)* pp. 1154–1158.

Want, R., Pering, T. and Tennenhouse, D. (2003), 'Comparing autonomic and proactive computing', *IBM Systems Journal* **42**(1), 129–135.

Weber, R. H. (2010), 'Internet of Things - New security and privacy challenges', *Computer Law and Security Review* **26**(1), 23–30.

Wen, J. and Wang, Z. (2016), 'Sensor-based adaptive activity recognition with dynamically available sensors', *Neurocomputing* **218**, 307–317.

Wen, J., Zhong, M. and Wang, Z. (2015), 'Activity recognition with weighted frequent patterns mining in smart environments', *Expert Systems with Applications* **42**(17-18), 6423–6432.

Wu, Q., Ding, G., Xu, Y., Feng, S., Du, Z., Wang, J. and Long, K. (2014), 'Cognitive Internet of Things: A New Paradigm beyond Connection', *IEEE Internet of Things Journal* **PP**(99), 1–1.

Wu, X., Wu, J., Cheng, B. and Chen, J. (2013), 'Neural network based situation detection and service provision in the environment of IoT', *IEEE Vehicular Technology Conference* .

Ye, J., Dobson, S. and McKeever, S. (2012), 'Situation identification techniques in pervasive computing: A review', *Pervasive and Mobile Computing* **8**(1), 36–66.

Zaheeruddin and Garima (2006), 'A neuro-fuzzy approach for prediction of human work efficiency in noisy environment', *Applied Soft Computing* **6**(3), 283–294.

Appendices

Appendix A

Arudino Code

A.1 Arudino1 placement in Kitchen area for Inhabitant activity observation

```
#include <UnoWiFiDevEd.h>
#include <Wire.h>
#include <stdio.h>
// we assigned weight as pressure = 5, motion = 3,  and Light = 2
int a = 0;
//weights
double tw = .5;
double  mw = .3;
//double nw = .1;
double lw = .2;
int TSW;
int MSW;
//int NSW;
int LSW;
//state
int state;
// this is for cash2 home(RED)
//temparatue
int tempin=A4;
int tempval = 0;
int tval= 0;
//pressure
//int fsrpin = A3;
//int fsrval;
//int pval;
//timer
int t=1;
//Noise variables
int noisepin = A0;
int noiseval = 0;
int nv = 0;
//PIR variables
int pirpin = 9;
int pirval = 0;
```

```
//ultrasonic variables
int echopin = 7;
int trigpin = 6;
int duration = 0;
int distance = 0;
int USS;
//light variables
int lightpin = A3;
int lightval = 0;
int LS;
// holding all sensor values
int allval[] = {0,0,0,0};
int sum = 0;
//buffer
char buffer[100];
void setup() {
// put your setup code here, to run once:
Wifi.begin();
//Serial.begin(9600);
//pinMode(trigpin, OUTPUT);
//pinMode(echopin, INPUT);
//pinMode(pirpin, INPUT);
pinMode(lightpin, INPUT);
//pinMode(noisepin,INPUT);
//pinMode(fsrpin, INPUT);
pinMode(tempin, INPUT);
pinMode(pirpin, INPUT);
}
// put your main code here, to run repeatedly:
void loop() {
//temprature
tempval = analogRead(tempin);
float mv = (tempval/1024.0)*5000;
float cel = mv/10;
if(tempval > 976)
{tval =1;
TSW = tval*tw*10+20;
}else
{tval =0;
TSW = 0;
}
//light
lightval = analogRead(lightpin);
if (lightval >400)
{LS = 1;
LSW = LS*lw*10+20;
}else
{LS = 0;
LSW = 0;
}
//ULTRA SONIC
digitalWrite(trigpin, LOW);
delay(2);
digitalWrite(trigpin, HIGH);
delay(10);
digitalWrite(trigpin, LOW);
```

```
duration = pulseIn(echopin,HIGH);
distance = duration*0.034/2;
if(distance < 100)
{USS = 1;}
else
{USS = 0;}
*/
//Motion
pirval = digitalRead(pirpin);
if(pirval == 1)
{ MSW = pirval*mw*10+20;
}
else if (pirval == 0)
{
  MSW = 0;
}
//Noise
noiseval = analogRead(noisepin);
if (noiseval > 320)
{nv=1;
NSW = nv*nw*10+20;
}
else
{nv=0 ;
NSW = 0;
}
*/
for(int i = 0; i < 3; i++)
{
//define a array to hold all values
allval[0]=LSW;
allval[1]=MSW;
allval[2]=TSW;
//allval[3]=NSW;
//sum total array value
sum = sum +  allval[i];
}
if (sum > 47)
{
  //cooking
 state =  7;
  }
else if (sum > 44)
{
  //working in kitchen
  state = 8;}
 int k = sprintf(buffer, "%d %d %d %d %d %d", t, TSW, MSW, LSW, sum, state);
for (int L = 0; L<k;L++)
Wifi.print(buffer[L]);
Wifi.println();
//Serial.print(buffer[L]);
delay(10000);
t++;
sum = 0;
}
```

A.2 Arudino2 placement in Living-room area for Inhabitant activity observation

```
#include <UnoWiFiDevEd.h>
#include <Wire.h>
#include <stdio.h>
// this is for CAiSH flat on board
//we assigned the weight as motion = 4, light = 4 and noise = 2 (mw:4,lw:4,nw:2)
double mw = .5;
double lw = .4;
double nw = .1;
int MSW;
int LSW;
int NSW;
//state
int state;
//timer
int t=1;
//Noise variables
int noisepin = A1;
int noiseval = 0;
int nv = 0;
//PIR variables
int pirpin = 3;
int pirval;
//light variables
int lightpin = A2;
int lightval = 0;
int LS;
// holding all sensor values
int allval[] = {0,0,0};
int sum=0;
char buffer[100];
void setup() {
// put your setup code here, to run once:
Wifi.begin();
//Serial.begin(9600);
pinMode(pirpin, INPUT);
pinMode(lightpin, INPUT);
pinMode(noisepin,INPUT);
}
// put your main code here, to run repeatedly:
void loop() {
//light
lightval = analogRead(lightpin);
if (lightval > 400)
{ LS = 1;
LSW = LS*lw*10+10;
}
else if (lightval < 400)
{LS = 0;
LSW = 0;
}
//Motion
```

```
pirval = digitalRead(pirpin);
if(pirval == 1)
{MSW = pirval*mw*10+10;
}
else if(pirval == 0)
{
MSW= 0 ;
}
//Noise
noiseval = analogRead(noisepin);
if (noiseval > 500)
{nv=1;
NSW = nv*nw*10+10;
}
else if(noiseval < 500)
{nv=0;
NSW = 0;}
for(int i = 0; i < 3; i++)
{
//define a array to hold all values
allval[0]=LSW;
allval[1]=MSW;
allval[2]=NSW;
//sum total array value
sum = sum + allval[i];
}
if(sum>0andandsum< 25)
{
  //reading
state = 6;
}
else if(sum > 24)
{
  //working
state = 5;
}
else if (sum <25)
{
  //sleeping
  state = 6;
}
int k = sprintf(buffer, "%d %d %d %d %d %d", t, LSW, NSW, MSW, sum, state);
for (int L = 0; L<k;L++)
Wifi.print(buffer[L]);
Wifi.println();
// Serial.print(buffer[L]);
// Serial.println();
delay(10000);
t++;
sum = 0;
}
```

A.3 Arduino 3 placement in Bed-room area for Activities observation

```
#include <UnoWiFiDevEd.h>
#include <Wire.h>
#include <stdio.h>
// we assigned weight as pressure = 5, motion = 2, Noise = 1 and Light = 2
//weights
double pw = .5;
double  mw = .2;
double nw = .1;
double lw = .2;
int PSW;
int MSW;
int NSW;
int LSW;
//  state
int state;
//pressure
int fsrpin = A3;
int fsrval;
int pval;
//timer
int t=1;
//Noise variables
int noisepin = A0;
int noiseval = 0;
int nv = 0;
//PIR variables
int pirpin = 4;
int pirval = 0;
//ultrasonic variables
int echopin = 8;
int trigpin = 9;
int duration = 0;
int distance = 0;
int USS;
//light variables
int lightpin = A1;
int lightval = 0;
int LS;
// holding all sensor values
int allval[4] = {0,0,0,0};
int sum = 0;
//fuzzy score
int fuzzy_score = 0;
//buffer
char buffer[100];
void setup() {
// put your setup code here, to run once:
Wifi.begin();
//Serial.begin(9600);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
pinMode(pirpin, INPUT);
```

```
pinMode(lightpin, INPUT);
pinMode(noisepin,INPUT);
pinMode(fsrpin, INPUT);
}
// put your main code here, to run repeatedly:
void loop() {
//pressure
fsrval = analogRead(fsrpin);
if(fsrval > 966)
{ pval = 1;
PSW = pval*pw*10;
}
else if(fsrval <967)
{ pval = 0;
PSW = pval*pw*10;
}
//light
lightval = analogRead(lightpin);
if (lightval >400)
{LS = 1;
LSW = LS*lw*10;
}
else if(lightval <400 )
{LS = 0;
LSW = LS*lw*10;
}
//ULTRA SONIC
digitalWrite(trigpin, LOW);
delay(2);
digitalWrite(trigpin, HIGH);
delay(10);
digitalWrite(trigpin, LOW);
duration = pulseIn(echopin,HIGH);
distance = duration*0.034/2;
if(distance < 100)
{ USS = 1;}
else
{USS = 0;}
//Motion
pirval = digitalRead(pirpin);
if(pirval == 1)
{ MSW = pirval*mw*10;
}
else if (pirval == 0)
{
  MSW = pirval*mw*10;
}
//Noise
noiseval = analogRead(noisepin);
if (noiseval > 320)
{nv=1;
NSW = nv*nw*10;
}
else if (noiseval <320)
{nv=0 ;
NSW = nv*nw*10;
```

```
}
for(int i = 0; i < 4 ; i++)
{
//define a array to hold all values
allval[0]=PSW;
allval[1]=MSW;
allval[2]=NSW;
//allval[2]=USS;
allval[3]=LSW;
//sum total array value
//sum +=   allval[i];
 sum = sum + allval[i];
//i++;
}
if(sum >= 6)
{
  // sleeping
  state = 1;
}
else if(sum == 5)
{
  //reading
  state = 2;
}
else if(sum <5)
{
  //working
  state = 3;
}
 int k = sprintf(buffer, "%d %d %d %d %d %d %d", t, PSW, MSW, NSW, LSW, sum, state);
for (int L = 0; L<k;L++)
Wifi.print(buffer[L]);
Wifi.println();
//Serial.print(buffer[L]);
//Serial.println();

delay(10000);
t++;
sum = 0;
}
```

A.4 Hidden Markov Model

```
\begin{table}[h]
\caption{sample data set for HMM} %title of the table
\centering % centering table
\begin{tabular}{|c | c| c| } % creating eight columns
\hline\hline %inserting double−line
Seq/Obs andStates andHMM likelihood Patterns \\
 \hline
17and      2and2\\
16         and3         and3\\
20         and3         and3\\
6          andand       1\\
5          andand       2\\
14         and4         and4\\
4          and2         and2\\
7          and1         and1\\
15         andand       3\\
12         andand       4\\
23         andand       5\\
28         andand       6\\
25         andand       5\\
\hline
\end{tabular}
\label{tab:hresult}
\end{table}
\begin{lstlisting}
%% Import data from text file.
% Script for importing data from the following text file:
% Initialize variables.
 clc
 clear all
 close all
 seqtxt = 'C:\Users\40008665\Desktop\seq.txt';
  [DataSet, delimiterOut]=importdata(seqtxt);
% script.
%% Create output variable
%% Clear temporary variables
%Covert table into array matrix for HMM algorithms application%
N = count(DataSet);
 Observations = DataSet(:,1);
 States = DataSet(:,2);
[transMat, emisMat]=hmmestimate(observation, states)
[transMat2, emisMat2] = hmmtrain(Observation, transMat, emisMat)
 LikelihoodState = hmmviterbi(observation, transMat2, emisMat2)
sum(LikelihoodState== States)/N
 PosteriorStateProbability = hmmdecode(Observations, transMat2, emisMat2)
 figure
 plot(LikelihoodState, r)
 hold on
 plot(Observations, b)
 hold off
 axis ()
```

A.5 Recurrent Neural network

```
clc
clear all
close all
% Solve an Autoregression Problem with External Input with a NARX Neural Network
% Script generated by NTSTOOL
% Created Thu Jun 28 16:03:51 BST 2018
%
% This script assumes these variables are defined:
%
%   testspot - input time series.
%   testdata - feedback time series.

inputSeries = tonndata(testspot,false,false);
targetSeries = tonndata(testdata,false,false);

% Create a Nonlinear Autoregressive Network with External Input
inputDelays = 1:2;
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original time series data unchanged, while
% easily customizing it for networks with differing numbers of delays, with
% open loop or closed loop feedback modes.
[inputs,inputStates,layerStates,targets] = preparets(net,inputSeries,{},targetSeries);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,inputs,targets,inputStates,layerStates);

% Test the Network
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotregression(targets,outputs)
%figure, plotresponse(targets,outputs)
%figure, ploterrcorr(errors)
%figure, plotinerrcorr(inputs,errors)
```

```
% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the outout layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,inputSeries,{},targetSeries);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(netc,tc,yc)

% Early Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is given y(t+1).
% For some applications such as decision making, it would help to have predicted
% y(t+1) once y(t) is available, but before the actual y(t+1) occurs.
% The network can be made to return its output a timestep early by removing one delay
% so that its minimal tap delay is now 0 instead of 1. The new network returns the
% same outputs as the original network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,inputSeries,{},targetSeries);
ys = nets(xs,xis,ais);
earlyPredictPerformance = perform(nets,ts,ys)
```

A.6 NaiveBayes

```
clc
clear all
close all
%% Import the data
[~, ~, raw] = xlsread('C:\Users\40008665\Desktop\NBonly.xlsx','Main');
raw = raw(2:5500,1:15);
%% Replace non-numeric cells with 0.0
R = cellfun(@(x) ~isnumeric(x) || isnan(x),raw); % Find non-numeric cells
raw(R) = {0.0}; % Replace non-numeric cells
%% Create output variable
untitled = cell2mat(raw);
%% Clear temporary variables
clearvars raw R;
SpotActivity = untitled(:,[1:3]);
IntentionActivity = untitled(:,4);
%Since NaiveBayes by default treats all features as part of a normal distribution, it cannot work with a co
% data is closer to a multinomial model mn:
Model = NaiveBayes.fit(SpotActivity,IntentionActivity, 'Distribution', 'mn');
Result = Model.predict(SpotActivity);
Confmat = confusionmat(IntentionActivity, Result)
figure
plot(result, r)
```

A.7 Accuracy Measure

```
clc
clear all
close all
% Compute mean square  error
AccuracyErrorMSE = mse(net,t,y,ew)
% Compute mean absolute error
AccuracyErrorMAE= mae(E,Y,X,FP)
% Compute mean absolute percent error
AccuracyErrorMAPE = mape(actual, pred)
figure
plot(AccuracyErrorMSE, r)
hold on
plot(AccuracyErrorMAE, g)
hold on
plot(AccuracyErrorMAPE, b)
hold off

% rmse tutorial.

% The actual values that we want to predict.
Actual = [1 2 3 4];

% The values we actually predicted.
Predicted = [1 3 1 4];

% One way is to use the Root Mean Square function and pass in the "error" part.
```

```
rmse = rms(Predicted-Actual)

% That's it! You're done.
% But for those of you who are the curious type,
% here's how to calculate the root-mean-square-error by hand.

% First calculate the "error".
err = Actual - Predicted;

% Then "square" the "error".
squareError = err.^2;

% Then take the "mean" of the "square-error".
meanSquareError = mean(squareError);

% Then take the "root" of the "mean-square-error" to get
% the root-mean-square-error!

rootMeanSquareError = sqrt(meanSquareError)

% That's it! You have calculated the RMSE by hand.

% So, this is true.
rootMeanSquareError == rmse
```